# Stable Leader Election in Population Protocols Requires Linear Time

**David Doty**[*]
University of California, Davis
doty@ucdavis.edu

**David Soloveichik**[†]
University of Texas at Austin
david.soloveichik@utexas.edu

### Abstract

A population protocol *stably elects a leader* if, for all $n$, starting from an initial configuration with $n$ agents each in an identical state, with probability 1 it reaches a configuration $\mathbf{y}$ that is *correct* (exactly one agent is in a special leader state $\ell$) and *stable* (every configuration reachable from $\mathbf{y}$ also has a single agent in state $\ell$). We show that any population protocol that stably elects a leader requires $\Omega(n)$ expected "parallel time" — $\Omega(n^2)$ expected total pairwise interactions — to reach such a stable configuration. Our result also informs the understanding of the time complexity of chemical self-organization by showing an essential difficulty in generating exact quantities of molecular species quickly.

## 1   Introduction

**Background.**   Population protocols (PPs) were introduced by Angluin, Aspnes, Diamadi, Fischer, and Peralta [2] as a model of distributed computing in which the agents have very little computational power and no control over their schedule of interaction with other agents. They also can be thought of as a special case of Petri nets/vector addition systems [14, 15], which were introduced in the 1960s as a model of concurrent processing. In addition to being an appropriate model for electronic computing scenarios such as mobile sensor networks, they are a useful abstraction of "fast-mixing" physical systems such as animal populations [17], chemical reaction networks, and gene regulatory networks [7].

A PP is defined by a finite set $\Lambda$ of *states* that each agent may have, together with a *transition function* $\delta : \Lambda \times \Lambda \to \Lambda \times \Lambda$.[1]   Given states $r_1, r_2, p_1, p_2 \in \Lambda$, if $\delta(r_1, r_2) = (p_1, p_2)$ (denoted $r_1, r_2 \to p_1, p_2$) and a pair of agents in respective states $r_1$ and $r_2$ interact, then their states become $p_1$ and $p_2$.[2]   A *configuration* of a PP is a vector $\mathbf{c} \in \mathbb{N}^\Lambda$ describing, for each state $s \in \Lambda$, the *count* $\mathbf{c}(s)$ of how many agents are in state $s$. Executing a transition $r_1, r_2 \to p_1, p_2$ alters the configuration by decrementing the counts of $r_1$ and $r_2$ by 1 each and incrementing $p_1$ and $p_2$ by 1 each.[3]

[1]Some work on PPs allows "non-deterministic" transitions, in which the transition function maps to subsets of $\Lambda \times \Lambda$. Our results are independent of whether the PP is deterministic or nondeterministic in this manner.

[2]In the most generic model, there is no restriction on which agents are permitted to interact. If one prefers to think of the agents as existing on nodes of a graph, then it is the complete graph $K_n$ for a population of $n$ agents.

[3]Possibly some of $r_1, r_2, p_1, p_2$ are equal to each other, so the count of a state could change by 0, 1, or 2.

Associated with a PP is a set of *valid initial configurations* that we expect the PP to be able to handle.[4] Agents interact in a pairwise manner and change state based on the transition function. The next pair of agents to interact is chosen uniformly at random among the $n$ agents. (An interaction may be a "null transition" $r_1, r_2 \rightarrow r_1, r_2$.) We count the expected number of *interactions* until some event occurs, and then define the "parallel time" until this event as the expected number of interactions divided by the number of agents $n$. This measure of time is based on the natural parallel model where each agent participates in a constant number of interactions in one unit of time, hence $\Theta(n)$ total interactions are expected per unit time [4]. In this paper all references to "time" refer to parallel time.

In order to define error-free computation in PPs, we rely on to the notion of *stable* computation [5]. The PP must get to a configuration that is correct[5] and "stable" in the sense that no subsequent sequence of transitions can take the PP to an incorrect configuration. Error-free computation must be correct in an "adversarial" schedule of transitions: we require that from every configuration reachable by *any* sequence of transitions from the initial configuration, it is possible to reach to a correct stable configuration. Since the configuration space is finite, this is equivalent to requiring, under the randomized model, that a correct stable configuration is reached with probability 1.[6]

A PP works "with a leader" if there is a special "leader" state $\ell$, and every valid initial configuration $\mathbf{i}$ satisfies $\mathbf{i}(\ell) = 1$. This is in contrast to a uniform initial configuration ($\mathbf{i}(x) = n$ for some state $x$ and $\mathbf{i}(y) = 0$ for all states $y \neq x$) or an initial configuration only encoding the input ($\mathbf{i}(x_i) = n_i$ for $i \in \{1, \ldots, k\}$ to represent any input $(n_1, n_2, \ldots, n_k) \in \mathbb{N}^k$). It is known that the predicates $\phi : \mathbb{N}^k \rightarrow \{0, 1\}$ stably computable by PPs are exactly the semilinear predicates, whether an initial leader is allowed or not [5]. Although the initial leader does not alter the class of computable predicates, it may allow faster computation. Specifically, the fastest known PPs to stably compute semilinear predicates without a leader take as long as $\Theta(n)$ to converge.[7] In contrast, with a leader, it is known that any semilinear predicate can be stably computed with expected convergence time $O(\log^5 n)$ [4]. Thus, in certain circumstances, the presence of a initial leader seems to give PPs more computational power (e.g., to converge quickly). Angluin, Aspnes, and Eisenstat [4] asked whether polylogarithmic time stable computation of semilinear predicates is possible without a leader; absent a positive answer, the presence of a leader appears to add power to the model.

**Statement of main result.** Motivated in part by the apparent speedup possible with an initial leader, we ask how quickly a leader may be elected from a configuration lacking one. We pose the problem as follows: design a PP $\mathcal{P}$ with two special states $x$ (the initial state) and $\ell$ (the leader state, which may or may not be identical to $x$) such that, for every $n \in \mathbb{N}$, from the initial configuration $\mathbf{i}_n$ defined as $\mathbf{i}_n(x) = n$ and $\mathbf{i}_n(y) = 0$ for all other states $y$, has the following property. For every configuration $\mathbf{c}$ reachable from $\mathbf{i}_n$, there is a configuration $\mathbf{y}$ reachable from $\mathbf{c}$ that *has a stable leader*, meaning that in all configurations $\mathbf{y}'$ reachable from $\mathbf{y}$ (including $\mathbf{y}$ itself), $\mathbf{y}'(\ell) = 1$.[8]

---

[4]The set of valid initial configurations for a "self-stabilizing" PP is $\mathbb{N}^\Lambda$, where leader election is provably impossible [6]. We don't require the PP to work if started in any possible configuration, but rather allow potentially "helpful" initial configurations as long as they don't already have small count states (see "$\alpha$-dense" below).

[5]What "correct" means depends on the task. For computing a predicate, for example, $\Lambda$ is partitioned into "yes" and "no" voters, and a "correct" configuration is one in which every state present has the correct vote.

[6]It is also equivalent to requiring that every *fair* sequence of transitions reaches a correct stable configuration, where "fair" means that every configuration infinitely often reachable is infinitely often reached [5].

[7]See "Open questions" for the distinction between time to *converge* and time to *stabilize*. In this paper, the time lower bound we prove is on stabilization.

[8]Note that this problem abstracts away the idea that the leader might be *useful* for something (such as computing predicates quickly). In particular, if a certain PP requires an initial leader, and the correctness of the PP depends

There is a simple $O(n)$ expected time PP for stable leader election, with (assuming $x \equiv \ell$) the single transition $\ell, \ell \to \ell, f$. Our main theorem shows that *every* PP that stably elects a leader requires time $\Omega(n)$ to reach a state with a stable leader; thus the previous PP is asymptotically optimal. Section 3.3 discusses why some straightforward approaches to proving a time lower bound for leader election fail.

**Multiple leader states, multiple leaders, and other initial configurations.** A more general notion of leader election is to identify a subset $\Psi \subset \Lambda$ of states that are all considered leader states, and to require the PP to eventually reach a configuration $\mathbf{y}$ in which $\sum_{\ell \in \Psi} \mathbf{y}(\ell) = 1$, and this sum is 1 in every configuration reachable from $\mathbf{y}$. This corresponds more appropriately to how leader states actually coordinate computation in PPs: a leader agent must remember some state information in between transitions (hence it changes state while remaining the unique leader). Our techniques actually show this stronger result as well (as explained in Section 3.2). Further, our result implies that a PP cannot elect any fixed quantity of leaders (e.g. exactly 256) or variable quantity of leaders under a fixed bound (e.g. at most 256) in sublinear expected time.

In the simplest formulation of the task of leader election, we always start with $n$ agents in state $x$ (as described above). Can we capture more generally leader election from a configuration "without a pre-existing leader"? Intuitively, we want to exclude initial configurations with states present in small but non-zero count. We can exclude such initial configurations, but allow otherwise deliberately prepared starting conditions, using the notion of $\alpha$-dense configurations: any state present in the initial configuration has count $\geq \alpha n$. Our general negative result (Theorem 3.7) implies that even starting with best-case initial configurations, as long as, for some constant $\alpha > 0$, they are all $\alpha$-dense, sublinear time leader election is impossible. An open question relates to weakening the notion of $\alpha$-dense (see below).

**Chemical reaction networks.** The main result and proof are stated in the language of PPs; however, the result holds for more general systems that have PPs as a special case. The discrete, stochastic chemical reaction network (CRN) model has been extensively used in the natural sciences to model chemical kinetics in a well-mixed solution [13]. The CRN model is also used prescriptively for specifying the behavior of synthetic chemical systems [9, 16]. A CRN is a finite set of *species* (corresponding to PP states) such as $X, Y, Z$, and *reactions* (corresponding to PP transitions) such as $X + Y \overset{k_1}{\to} Z$ or $Y \overset{k_2}{\to} 2X + Z$. CRNs can be thought of as a generalization of PPs in which spontaneous transitions are possible (unimolecular reactions), the transition may cause the number of agents to change (if the reaction has a different number of products than reactants), and each transition has an associated constant $k$ that affects its probability of being selected.

As an essential form of self-organization, biological cells seem able to precisely control the count of certain molecules (centriole number [10] is a well studied example). How chemical systems transform relatively uncontrolled initial conditions to precisely controlled amounts of desired species is still not well understood. Our negative result applied to CRNs[9] implies that generating with probability 1 an exact count of a certain species, whether 1 or 256, is *necessarily* slower ($\Omega(n)$ time) than, for example, destroying all molecules of the species (through the reaction $X \to \varnothing$), which takes $O(\log(n))$ time.

---

on the count of the leader never exceeding 1, prior to the conclusion of the leader election, the presence of multiple leaders may result in unintended transitions. However, our main result is a impossibility theorem, showing that even if the objective is simplified to stable leader election, without requiring the leader to be useful for any subsequent task, this *still* requires $\Omega(n)$ time.

[9]Our result holds for any CRN that obeys Theorem 4.3, the precise constraints of which are specified in [12] (those constraints automatically apply to all PPs).

**Open questions.** An important open question concerns the contrast between *convergence* and *stabilization*. We say a PP electing a leader *converges* when it stops changing the count of the leader (if it is correct, this count should be 1), and we say it *stabilizes* when it first enters a configuration from which the count of the leader *cannot* change. In many PPs these two events coincide, but it is possible to converge strictly before stabilizing. Section 3.3 shows some examples of PPs that converge before stabilizing. Our proof shows only that stabilization must take expected $\Omega(n)$ time. We leave as an open question whether there is a PP that stably elects a leader and converges in expected $o(n)$ time. Recall that there are PPs that work with a leader to stably compute semilinear predicates with convergence time $O(\log^5 n)$ [4]. Thus if stable leader election can converge in expected sublinear time, by coupling the two PPs it might be possible to achieve stable computation of arbitrary semilinear predicates with sublinear convergence time.

It is similarly open to determine the optimal stabilization time for computing semilinear predicates. The stably computing PPs converging in $O(\log^5 n)$ time [4] provably require expected time $\Omega(n)$ to stabilize, and it is unknown whether faster stabilization is possible even with an initial leader.

The open question of Angluin, Aspnes, and Eisenstat [4] asks whether their efficient high-probability simulation of a space-bounded Turing machine by a PP could remove the assumption of an initial leader. That simulation has some small probability $\epsilon > 0$ of failure, so if one could elect a leader with a small probability $\epsilon' > 0$ of error and subsequently use it to drive the simulation, by the union bound the total probability of error would be at most $\epsilon + \epsilon'$ (i.e., still close to 0). However, it remains an open question whether the necessary PP exists. Alistairh and Gelashvili [1] showed that relaxing the requirement of $O(1)$ states to $O(\log^3 n)$ states allows for a leader to be elected with high probability in expected time $O(\log^3 n)$.[10]

Our general negative result applies to $\alpha$-dense initial configurations. However, is sublinear time stable leader election possible from other kinds of initial configurations that satisfy our intuition of not having preexisting leaders? It is known, for example, that for each $0 < \varepsilon < 1$, an initial configuration with $\Theta(n)$ agents in one state and $\Theta(n^\varepsilon)$ in another state can elect a leader in expected time $O(\log^2 n)$ with high probability [4], although this protocol has a positive probability of failure. In Section 3.3 we give an example PP that stably elects a leader in $O(n^{1/2} \log n)$ time starting from an initial configuration with $\Theta(n)$ agents in one state and $\Theta(n^{1/4})$ in another state. In general we want to better characterize the initial configurations for which sublinear time leader election is possible.

## 2 Preliminaries

If $\Lambda$ is a finite set (in this paper, of *states*), we write $\mathbb{N}^\Lambda$ to denote the set of functions $\mathbf{c} : \Lambda \to \mathbb{N}$. Equivalently, we view an element $\mathbf{c} \in \mathbb{N}^\Lambda$ as a vector of $|\Lambda|$ nonnegative integers, with each coordinate "labeled" by an element of $\Lambda$. Given $s \in \Lambda$ and $\mathbf{c} \in \mathbb{N}^\Lambda$, we refer to $\mathbf{c}(s)$ as the *count of $s$ in $\mathbf{c}$*. Let $\|\mathbf{c}\| = \|\mathbf{c}\|_1 = \sum_{s \in \Lambda} \mathbf{c}(s)$ denote the total number of agents. We write $\mathbf{c} \leq \mathbf{c}'$ to denote that $\mathbf{c}(s) \leq \mathbf{c}'(s)$ for all $s \in \Lambda$. Since we view vectors $\mathbf{c} \in \mathbb{N}^\Lambda$ equivalently as multisets of elements from $\Lambda$, if $\mathbf{c} \leq \mathbf{c}'$ we say $\mathbf{c}$ is a *subset* of $\mathbf{c}'$. It is sometimes convenient to use multiset notation to denote vectors, e.g., $\{x, x, y\}$ and $\{2x, y\}$ both denote the vector $\mathbf{c}$ defined by $\mathbf{c}(x) = 2$, $\mathbf{c}(y) = 1$, and $\mathbf{c}(z) = 0$ for all $z \notin \{x, y\}$. Given $\mathbf{c}, \mathbf{c}' \in \mathbb{N}^\Lambda$, we define the vector component-wise operations of addition $\mathbf{c} + \mathbf{c}'$, subtraction $\mathbf{c} - \mathbf{c}'$, and scalar multiplication $m\mathbf{c}$ for $m \in \mathbb{N}$. For a set $\Delta \subset \Lambda$, we view a vector $\mathbf{c} \in \mathbb{N}^\Delta$ equivalently as a vector $\mathbf{c} \in \mathbb{N}^\Lambda$ by assuming $\mathbf{c}(s) = 0$ for all $s \in \Lambda \setminus \Delta$.

---

[10]Indeed, our proof technique fails if the number of states is not constant with respect to $n$.

## 2.1 Population Protocols

A *population protocol (PP)* is a pair $\mathcal{P} = (\Lambda, \delta)$,[11] where $\Lambda$ is a finite set of *states*, and $\delta : \Lambda \times \Lambda \to \Lambda \times \Lambda$ is the (symmetric) *transition function*. A *configuration* of a PP is a vector $\mathbf{c} \in \mathbb{N}^\Lambda$, with the interpretation that $\mathbf{c}(s)$ agents are in state $s$. By convention, the value $n \in \mathbb{Z}^+$ represents the total number of agents $\|\mathbf{c}\|$. A *transition* is a 4-tuple $\alpha = (r_1, r_2, p_1, p_2) \in \Lambda^4$, written $\alpha : r_1, r_2 \to p_1, p_2$, such that $\delta(r_1, r_2) = (p_1, p_2)$. This paper typically defines a PP by a list of transitions, with $\delta$ implicit (there is a null transition $\delta(r_1, r_2) = (r_1, r_2)$ if a different transition is not specified). If an agent in state $r_1$ interacts with an agent in state $r_2$, then they change states to $p_1$ and $p_2$.

More formally, given a configuration $\mathbf{c}$ and transition $\alpha : r_1, r_2 \to p_1, p_2$, we say that $\alpha$ is *applicable* to $\mathbf{c}$ if $\mathbf{c} \geq \{r_1, r_2\}$, i.e., $\mathbf{c}$ contains 2 agents, one in state $r_1$ and one in state $r_2$. If $\alpha$ is applicable to $\mathbf{c}$, then write $\alpha(\mathbf{c})$ to denote the configuration $\mathbf{c} - \{r_1, r_2\} + \{p_1, p_2\}$ (i.e., the configuration that results from applying $\alpha$ to $\mathbf{c}$); otherwise $\alpha(\mathbf{c})$ is undefined. A finite or infinite sequence of transitions $(\alpha_i)$ is a *transition sequence*. Given an initial configuration $\mathbf{c}_0$ and a transition sequence $(\alpha_i)$, the induced *execution sequence* (or *path*) is a finite or infinite sequence of configurations $(\mathbf{c}_0, \mathbf{c}_1, \ldots)$ such that, for all $\mathbf{c}_i$ $(i \geq 1)$, $\mathbf{c}_i = \alpha_{i-1}(\mathbf{c}_{i-1})$.[12] If a finite execution sequence, with associated transition sequence $q$, starts with $\mathbf{c}$ and ends with $\mathbf{c}'$, we write $\mathbf{c} \Longrightarrow_q \mathbf{c}'$. We write $\mathbf{c} \Longrightarrow \mathbf{c}'$ if such a transition sequence exists (i.e., it is possible for the system to reach from $\mathbf{c}$ to $\mathbf{c}'$) and we say that $\mathbf{c}'$ is *reachable* from $\mathbf{c}$. If it is understood from context what is the initial configuration $\mathbf{i}$, then say $\mathbf{c}$ is simply *reachable* if $\mathbf{i} \Longrightarrow \mathbf{c}$. Note that this notation omits mention of $\mathcal{P}$; we always deal with a single PP at a time, so it is clear from context which PP is defining the transitions. If a transition $\alpha : r_1, r_2 \to p_1, p_2$ has the property that for $i \in \{1, 2\}$, $r_i \notin \{p_1, p_2\}$, or if ($r_1 = r_2$ and ($r_i \neq p_1$ or $r_i \neq p_2$)), then we say that $\alpha$ *consumes* $r_i$. In other words, applying $\alpha$ reduces the count of $r_i$. We similarly say that $\alpha$ *produces* $p_i$ if it increases the count of $p_i$.

We will find ourselves frequently dealing with infinite sequences of configurations.[13] The following lemma, used frequently in reasoning about population protocols, shows that we can always take a nondecreasing subsequence.

**Lemma 2.1** (Dickson's Lemma [11]). *Any infinite sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots \in \mathbb{N}^k$ has an infinite nondecreasing subsequence $\mathbf{x}_{i_0} \leq \mathbf{x}_{i_1} \leq \ldots$, where $i_0 < i_1 < \ldots \in \mathbb{N}$.*

## 2.2 Time Complexity

In any configuration the next interaction is chosen by selecting a pair of agents uniformly at random and applying transition function $\delta$. To measure time we count the expected total number of interactions (including null), and divide by the number of agents $n$. (In the population protocols literature, this is often called "parallel time"; i.e. $n$ interactions among a population of $n$ agents corresponds to one unit of time). Let $\mathbf{c} \in \mathbb{N}^\Lambda$ and $C \subseteq \mathbb{N}^\Lambda$. Denote the probability that the PP reaches from $\mathbf{c}$ to some configuration $\mathbf{c}' \in C$ by $\Pr[\mathbf{c} \Longrightarrow C]$. If $\Pr[\mathbf{c} \Longrightarrow C] = 1$,[14] define the *expected*

---

[11]We give a slightly different formalism than that of [5] for population protocols. The main difference is that since we are not deciding a predicate, there is no notion of inputs being mapped to states or states being mapped to outputs. Another difference is that we assume (for the sake of brevity in some explanations, not because the difference is essential to the proof) the transition function is symmetric (so there is no notion of a "sender" and "receiver" agent as in [5]; the unordered pair of states completely determines the next pair of states).

[12]When the initial configuration to which a transition sequence is applied is clear from context, we may overload terminology and refer to a transition sequence and an execution sequence interchangeably.

[13]In general these will not be *execution* sequences. Typically none of the configurations are reachable from any others because they are configurations with increasing numbers of agents.

[14]Since PP's have a finite reachable configuration space, this is equivalent to requiring that for all $\mathbf{x}$ reachable from $\mathbf{c}$, there is a $\mathbf{c}' \in C$ reachable from $\mathbf{x}$.

*time to reach from* $\mathbf{c}$ *to* $C$, denoted $\mathsf{T}[\mathbf{c} \Longrightarrow C]$, to be the expected number of interactions to reach from $\mathbf{c}$ to some $\mathbf{c}' \in C$, divided by the number of agents $n$.

# 3 Main Results

## 3.1 Impossibility of Sublinear Time Stable Leader Election

We consider the following *stable leader election* problem. Suppose that each PP $\mathcal{P} = (\Lambda, \delta)$ we consider has a specially designated state $\ell \in \Lambda$, which we call the *leader state*. Informally, the goal of stable leader election is to be guaranteed to reach a configuration with count 1 of $\ell$ (a leader has been "elected"), from which no transition sequence can change the count of $\ell$ (the leader is "stable"). We also assume there is a special initial state $x$ (it could be that $x \equiv \ell$ but it is not required), such that the only valid initial configurations $\mathbf{i}$ are of the form $\mathbf{i}(x) > 0$ and $\mathbf{i}(y) = 0$ for all states $y \in \Lambda \setminus \{x\}$. We write $\mathbf{i}_n$ to denote such an initial configuration with $\mathbf{i}_n(x) = n$.

**Definition 3.1.** A configuration $\mathbf{y}$ is *stable* if, for all $\mathbf{y}'$ such that $\mathbf{y} \Longrightarrow \mathbf{y}'$, $\mathbf{y}'(\ell) = \mathbf{y}(\ell)$ (in other words, after reaching $\mathbf{y}$, the count of $\ell$ cannot change); $\mathbf{y}$ is said to have a *stable leader* if it is stable and $\mathbf{y}(\ell) = 1$.

The following definition captures our notion of stable leader election. It requires the PP to be "guaranteed" eventually to reach a configuration with a stable leader.

**Definition 3.2.** We say a PP elects a leader *stably* if, for all $n \in \mathbb{Z}^+$, for all $\mathbf{c}$ such that $\mathbf{i}_n \Longrightarrow \mathbf{c}$, there exists $\mathbf{y}$ with a stable leader such that $\mathbf{c} \Longrightarrow \mathbf{y}$.

In other words, every reachable configuration can reach to a configuration with a stable leader. It is well-known [5] that the above definition is equivalent to requiring that the PP reaches a configuration with a stable leader with probability 1.

**Definition 3.3.** Let $t : \mathbb{Z}^+ \to \mathbb{R}^+$, and let $Y$ be the set of all configurations with a stable leader. We say a PP elects a leader stably *in time* $t(n)$ if, for all $n \in \mathbb{Z}^+$, $\mathsf{T}[\mathbf{i}_n \Longrightarrow Y] \leq t(n)$.

Our main theorem says that stable leader election requires at least linear time to stabilize:

**Theorem 3.4.** *If a PP stably elects a leader in time* $t(n)$*, then* $t(n) = \Omega(n)$*.*

Thus a PP that elects a leader in sublinear time cannot do so stably, i.e., it must have a positive probability of failure.

The high-level strategy to prove Theorem 3.4 is as follows. With high probability the PP initially goes from configuration $\mathbf{i}_n$ to configuration $\mathbf{x}_n$, such that in the sequence $(\mathbf{x}_n)$ for increasing population size $n$, every state count grows without bound as $n \to \infty$ (indeed $\Omega(n)$); this follows from Theorem 4.3. We then show that any such configuration must have an "$O(1)$-bottleneck transition" before reaching a configuration with a stable leader (informally this means that every transition sequence from $\mathbf{x}_n$ to a configuration $\mathbf{y}$ with a stable leader must have a transition in which both input states have count $O(1)$, depending on the PP but not on $n$). Since it takes expected time $\Omega(n)$ to execute a transition when both states have constant count, from any such configuration it requires linear time to stably elect a leader. Since one of these configurations is reached from the initial configuration with high probability, those configurations' contribution to the overall expected time dominates, showing that the expected time to stably elect a leader is linear.

## 3.2 More General Impossibility Result in Terms of Inapplicable Transitions and Dense Configurations

Rather than proving Theorem 3.4 using the notion of leader stability directly, we prove a more general result concerning the notion of a set of inapplicable transitions. The two generalizations are as follows. (1) A configuration $\mathbf{y}$ is stable by Definition 3.1 if no transition altering the count of $\ell$ is applicable in any configuration reachable from $\mathbf{y}$; Definition 3.5 generalizes this to an arbitrary subset $Q$ of transitions. (2) The valid initial configurations of Section 3.1 are those with $\mathbf{i}_n(x) = n$ and $\mathbf{i}_n(y) = 0$ for all $y \in \Lambda \setminus \{x\}$; Theorem 3.7 generalizes this to any set $I$ of configurations that are all "$\alpha$-dense" (defined below) for a fixed $\alpha > 0$ independent of $n$, with a weak sort of "closure under addition" property: namely, that for infinitely many $\mathbf{i}, \mathbf{i}' \in I$, we have $\mathbf{i} + \mathbf{i}' \in I$.

**Definition 3.5.** Let $Q$ be a set of transitions. A configuration $\mathbf{y} \in \mathbb{N}^\Lambda$ is said to be $Q$-stable if no transition in $Q$ is applicable in any configuration reachable from $\mathbf{y}$.

If we let $Q$ be the set of transitions that alter the count of the leader state $\ell$, then a $Q$-stable configuration $\mathbf{y}$ with $\mathbf{y}(\ell) = 1$ exactly corresponds to the property of having a stable leader.

Let $I \subseteq \mathbb{N}^\Lambda$ and $Q$ be a set of transitions. Let $Y$ be the set of $Q$-stable configurations reachable from some configuration in $I$. We say that a PP $\mathcal{P} = (\Lambda, \delta)$ *Q-stabilizes from $I$* if, for any $\mathbf{i} \in I$, $\Pr[\mathbf{i} \Longrightarrow Y] = 1.$[15] If $I$ and $Q$ are understood from context, we say that $\mathcal{P}$ *stabilizes*. For a time bound $t(n)$, we say that $\mathcal{P}$ stabilizes *in expected time $t(n)$* if, for all $\mathbf{i} \in I$ such that $\|\mathbf{i}\| = n$, $\mathsf{T}[\mathbf{i} \Longrightarrow Y] \leq t(n)$.

To prove our time lower bound, we show that a "slow" transition necessarily occurs, which means that the counts of the two states in the transition are "small" when it occurs. We will pick a particular nondecreasing infinite sequence $C$ of configurations and define "small" relative to it: the "small count" states are those whose counts are bounded in $C$ (denoted $\mathsf{bdd}(C)$ below).

**Definition 3.6.** For an (infinite) set/sequence of configurations $C$, let $\mathsf{bdd}(C)$ be the set of states $\{ s \in \Lambda \mid (\exists b \in \mathbb{N})(\forall \mathbf{c} \in C) \; \mathbf{c}(s) < b \}$. Let $\mathsf{unbdd}(C) = \Lambda \setminus \mathsf{bdd}(C)$.

**Remark.** *Note that if $C = (\mathbf{c}_m)$ is a nondecreasing sequence, then for all $k \in \mathbb{N}$, there is $\mathbf{c}_m$ such that for all $s \in \mathsf{unbdd}(\mathbf{c}_m)$, $\mathbf{c}_m(s) \geq k$. (Note that if $C$ is not nondecreasing, the conclusion can fail; e.g., $\mathbf{c}_m(s_1) = m, \mathbf{c}_m(s_2) = 0$ for $m$ even and $\mathbf{c}_m(s_1) = 0, \mathbf{c}_m(s_2) = m$ for $m$ odd.)*

Let $0 < \alpha \leq 1$. We say that a configuration $\mathbf{c}$ is *$\alpha$-dense* if for all $s \in \Lambda$, $\mathbf{c}(s) > 0$ implies that $\mathbf{c}(s) \geq \alpha\|\mathbf{c}\|$, i.e., all states present in $\mathbf{c}$ occupy at least an $\alpha$ fraction of the total count of agents.

Theorem 3.4 is implied by the next theorem, which the rest of the paper is devoted to proving.

**Theorem 3.7.** *Let $\mathcal{P} = (\Lambda, \delta)$, let $Q$ be any subset of transitions of $\mathcal{P}$, let $\alpha > 0$, and let $I \subseteq \mathbb{N}^\Lambda$ be a set of $\alpha$-dense initial configurations such that, for infinitely many $\mathbf{i}, \mathbf{i}' \in I$, $\mathbf{i} + \mathbf{i}' \in I$. Let $Y$ be the set of $Q$-stable configurations reachable from $I$, and let $\Delta = \mathsf{bdd}(Y)$. Suppose $\mathcal{P}$ $Q$-stabilizes from $I$ in expected time $o(n)$. Then there are infinitely many $\mathbf{y} \in Y$ such that $\forall s \in \Delta$, $\mathbf{y}(s) = 0$.*

In other words, if some states have "small" count in all reachable stable configurations, then there is a reachable stable configuration in which those states have count 0. A PP $\mathcal{P}$ that stably elects a leader is a PP in which $Q$ is the set of transitions that alter the count of $\ell$, $I = \{ \mathbf{i}_n \mid n \in \mathbb{N} \}$ (note all $\mathbf{i}_n$ are 1-dense), $Y$ is the set of configurations reachable from $I$ with a stable leader, and $\mathcal{P}$ $Q$-stabilizes from $I$. Hence by Theorem 3.7, if $\mathcal{P}$ stabilizes in expected time $o(n)$, there is a stable reachable $\mathbf{y}$ where $\mathbf{y}(\ell) = 0$, a contradiction. Thus Theorem 3.4 follows from Theorem 3.7.

---

[15]Recall that the condition $\Pr[\mathbf{i} \Longrightarrow Y] = 1$ is equivalent to $[(\forall \mathbf{c} \in \mathbb{N}^\Lambda) \; \mathbf{i} \Longrightarrow \mathbf{c}$ implies $(\exists \mathbf{y} \in Y) \; \mathbf{c} \Longrightarrow \mathbf{y}]$.

We can also use Theorem 3.7 to prove that stable leader election requires linear time under the more relaxed requirement that there is a set $\Psi \subset \Lambda$ of "leader states," and the goal of the PP is to reach a configuration $\mathbf{y}$ in which $\sum_{\ell \in \Psi} \mathbf{y}(\ell) = 1$ and stays 1 in any configuration reachable from $\mathbf{y}$. Choosing $Q$ as the set of transitions that alter that sum, Theorem 3.7 implies this form of stable leader election also requires $\Omega(n)$ expected time.

Throughout the rest of this paper, fix $\mathcal{P} = (\Lambda, \delta)$, $\alpha$, $I$, and $Q$ as in the statement of Theorem 3.7.

## 3.3 Why Simple Proofs Fail

It is tempting to believe that the main theorem follows by a simple argument based on reasoning about the last transition to change the count of the leader. Indeed, if we start with more than one leader, and no transition rule can produce a new leader, then we can easily prove the impossibility of sublinear time leader election as follows. To quickly reduce from two leaders to one, the other agent's state must be numerous in the population, so the same transition could occur again. This would leave us with no leaders and no possibility to make a new leader. However, if transitions *can* produce new leaders, then the argument cannot reason only about the last transition involving the leader. We illustrate this using two examples, which the authors have found helpful in ruling out plausible-sounding but ultimately insufficient ideas for proving a negative result.

We describe two PPs that stably elect a leader in sublinear time starting from initial configurations that are not $\alpha$-dense (for $\alpha > 0$ independent of $n$). (Since the initial configurations are not $\alpha$-dense these PPs do not contradict the statement of our main theorem.) In both examples, with high probability exactly one transition involving the leader occurs. In the first example the transition produces precisely one leader in a configuration that previously had none, whereas in the second example, it consumes precisely one leader in a configuration that previously had two. (Clearly, these are the only two possible forms of the final transition involving the leader.) The examples imply that any proof of the main result cannot be based solely on reasoning about the final transition, but must additionally establish that configurations such as the initial configurations of these PPs cannot be reached with high probability in sublinear time.

Consider the following PP, with initial configuration $\mathbf{i}$ given by $\mathbf{i}(r) = n^{1/4}$, $\mathbf{i}(x) = n - n^{1/4}$, and transitions:

$$r, r \;\rightarrow\; \ell, k \tag{1}$$
$$r, k \;\rightarrow\; k, k \tag{2}$$
$$x, k \;\rightarrow\; k, k \tag{3}$$
$$\ell, \ell \;\rightarrow\; \ell, k \tag{4}$$

It takes expected time $\Theta(n^{1/2})$ for transition (1) to occur for the first time, producing a single leader. Transition (4) ensures that if transition (1) occurs more than once, the PP will eventually stabilize to a single leader. However, with high probability transitions (2) and (3) consume all $r$ and $x$ *before* (1) executes a second time. After exactly one instance of transition (1) occurs, let a *speed fault* denote the event that transition (1) occurs again (this is the same speed fault concept studied in ref. [8]). For convenience, for state $s \in \Lambda$, let $s$ also denote the count of that state in the configuration considered. The probability of a speed fault in any particular configuration is $\frac{r(r-1)}{r(r-1)+2k(n-k-1)} < \frac{n^{1/2}}{k(n-k-1)}$.[16] By the union bound, the probability that a speed fault occurs in between $k = 1$ and $k = n - 1$ (at which point transition (1) is disabled and the PP stabilizes) is at

---

[16]$r(r-1)$ is the number of ways of choosing two agents in state $r$, and $2k(n-k-1)$ is the number of ways of choosing an agent in state $k$ and another agent in state either $r$ or $x$, when $\ell = 1$, and therefore $r + x = n - k - 1$.

most

$$n^{1/2} \sum_{k=1}^{n-2} \frac{1}{k(n-k-1)} = n^{1/2} O\left(\frac{\log n}{n}\right) = O\left(\frac{\log n}{n^{1/2}}\right).$$

Whether or not a speed fault occurs, to produce $\ell$, transition (1) must occur for the first time, taking expected time $O(n^{1/2})$. After this, if no speed fault occurs, the expected time to stabilize is at most $O(\log n)$ (the expected time for transitions (2) and (3) to consume all $r$ and $x$).[17] If a speed fault occurs, then transition (4) must execute enough times to reduce $\ell$ to 1, which requires expected time $O(n)$ [4]. Thus, the total expected time to stabilize to a single leader is as follows, where we let T be the random variable denoting the time to stabilization *after* transition (1) has occurred for the first time:

$$O(n^{1/2}) + \Pr[\text{speed fault}] \cdot \mathsf{E}[\text{T}|\text{speed fault}] + \Pr[\text{no speed fault}] \cdot \mathsf{E}[\text{T}|\text{no speed fault}]$$
$$\leq O(n^{1/2}) + O\left(\frac{\log n}{n^{1/2}}\right) \cdot O(n) + 1 \cdot O(\log n) = O(n^{1/2} \log n),$$

i.e., sublinear time.

The above PP uses a non-dense initial configuration since $\mathbf{i}(r) = o(n)$. Thus, although it does not directly contradict the existence of a linear time lower bound from dense configurations, it points out that any proof based on reasoning about the last transition to alter the count of $\ell$ must disallow the possibility that a leader is elected from some *intermediate* configuration in the manner described above. With high probability all states obtain count $\Omega(n)$ in a constant amount of time;[18] however, it is possible to subsequently reduce some states to sublinear count after superconstant time. *A priori*, it is conceivable that after, say, $O(\log n)$ time, the PP reaches a non-dense configuration, with $\ell = 0$ and $r \approx n^{1/4}$ similar to $\mathbf{i}$ above, which would then elect a leader in sublinear time by producing a single $\ell$ with high probability.

Even if the final change of $\ell$ takes it from 2 to 1, it is *a priori* conceivable that the final transition $r, \ell \to p_1, p_2$ to consume $\ell$ has count $o(n)$ of $r$, so that, although a second execution of the transition is *possible*, the second execution requires sufficiently long expected time that the system, in the meantime, likely consumes all remaining copies of $r$, along with a mechanism to ensure that a leader is elected even if the second leader is also consumed by an $r$. The following PP achieves this, with initial configuration $\mathbf{i}$ given by $\mathbf{i}(\ell) = 2$, $\mathbf{i}(r) = n^{1/2}$, $\mathbf{i}(x) = n - \mathbf{i}(r) - \mathbf{i}(\ell)$, with transitions

$$r, \ell \ \to \ r, \ell' \tag{5}$$
$$\ell', x \ \to \ \ell', k \tag{6}$$
$$k, x \ \to \ k, k \tag{7}$$
$$k, r \ \to \ k, k \tag{8}$$
$$\ell', \ell' \ \to \ \ell, k \tag{9}$$

An analysis similar to the previous PP shows that the expected time to stabilize to $\ell = 1$ is $O(n^{1/2} \log n)$. Informally, transition (5) consumes one copy of $\ell$ after expected time $O(n^{1/2})$. Transition (6) subsequently produces $k$ in expected time $O(1)$, and transitions (7) and (8) remove all $r$ and $x$ in expected time $O(\log n)$. With high probability this happens before transition (5) can execute a second time, but if not, then $\ell' = 2$, so transition (9) guarantees that a single leader is stably elected (as above, if this is needed, it requires expected time $\Omega(n)$, but it is needed with such low probability that the overall expected time remains sublinear).

---

[17]Since conditioning on no speed fault can only *reduce* the expected time for transitions (2) and (3) to complete, we can use $O(\log n)$ as an upper bound for that conditional expectation.

[18]This is the main theorem of [12], of which Theorem 4.3 is a corollary.

In summary, the two PPs above demonstrate that the proof cannot be based solely on reasoning about the final transition to alter $\ell$, no matter whether that transition increases $\ell$ from 0 to 1 or decreases it from 2 to 1.

# 4 Technical Tools

## 4.1 Bottleneck Transitions Require Linear Time

This section proves a straightforward observation used in the proof of our main theorem. It states that, if to get from a configuration $\mathbf{x} \in \mathbb{N}^\Lambda$ to some configuration in a set $Y \subseteq \mathbb{N}^\Lambda$, it is necessary to execute a transition $r_1, r_2 \to p_1, p_2$ in which the counts of $r_1$ and $r_2$ are both at most some number $b$, then the expected time to reach from $\mathbf{x}$ to some configuration in $Y$ is $\Omega(n/b^2)$.

Let $b \in \mathbb{N}$. We say that transition $\alpha : r_1, r_2 \to p_1, p_2$ is a $b$-*bottleneck* for configuration $\mathbf{c}$ if $\mathbf{c}(r_1) \le b$ and $\mathbf{c}(r_2) \le b$.

**Observation 4.1.** *Let $b \in \mathbb{N}$, $\mathbf{x} \in \mathbb{N}^\Lambda$, and $Y \subseteq \mathbb{N}^\Lambda$ such that $\mathsf{Pr}[\mathbf{x} \Longrightarrow Y] = 1$. If every transition sequence taking $\mathbf{x}$ to a configuration $\mathbf{y} \in Y$ has a $b$-bottleneck, then $\mathsf{T}[\mathbf{x} \Longrightarrow Y] \ge \frac{n-1}{2(b \cdot |\Lambda|)^2}$.*

*Proof.* Because every transition sequence taking $\mathbf{x}$ to a configuration $\mathbf{y} \in Y$ has a $b$-bottleneck, it suffices to show that the expected time of the first $b$-bottleneck transition to occur from configuration $\mathbf{x}$ is at least $\frac{n-1}{2(b \cdot |\Lambda|)^2}$, since the definition of $b$-bottleneck implies that no configuration in $Y$ can appear before that event. In any configuration $\mathbf{c}$ reachable from $\mathbf{x}$, for any pair of states $r_1, r_2 \in \Lambda$ such that $r_1, r_2 \to p_1, p_2$ is a $b$-bottleneck transition in $\mathbf{c}$, the definition implies that $\mathbf{c}(r_1), \mathbf{c}(r_2) \le b$. Thus the probability that the next pair of agents selected to interact are in states $r_1$ and $r_2$, respectively is at most $\frac{2b^2}{n(n-1)}$.[19] There are at most $|\Lambda|^2$ such transitions,[20] so the union bound implies that the probability that any of them is the next transition is at most $|\Lambda|^2 \frac{2b^2}{n(n-1)}$. Thus we can bound the number of interactions until the first $b$-bottleneck transition occurs by a geometric random variable with success probability at most $\frac{2(b \cdot |\Lambda|)^2}{n(n-1)}$, whence the expected number of interactions until the first $b$-bottleneck is at least $\frac{n(n-1)}{2(b \cdot |\Lambda|)^2}$. Since the parallel time is defined as the number of interactions divided by $n$, this gives the stated expected time bound. $\qquad\square$

**Corollary 4.2.** *Let $\gamma > 0$, $b \in \mathbb{N}$, $\mathbf{c} \in \mathbb{N}^\Lambda$, and $X, Y \subseteq \mathbb{N}^\Lambda$ such that $\mathsf{Pr}[\mathbf{c} \Longrightarrow X] \ge \gamma$, $\mathsf{Pr}[\mathbf{c} \Longrightarrow Y] = 1$, and every transition sequence from every $\mathbf{x} \in X$ to some $\mathbf{y} \in Y$ has a $b$-bottleneck. Then $\mathsf{T}[\mathbf{c} \Longrightarrow Y] \ge \gamma \frac{n-1}{2(b \cdot |\Lambda|)^2}$.*

## 4.2 Sublinear Time from Dense Configurations Implies Bottleneck Free Path from Configurations with Every State "Populous"

The following theorem, along with Corollary 4.2, fully captures the probability theory necessary to prove our main theorem.[21] Given it and Corollary 4.2, Theorem 3.7 is provable (through

---

[19]If $r_1 \ne r_2$ and $\mathbf{c}(r_1) = \mathbf{c}(r_2) = b$, then the probability to pick the first agent in one of the states $r_1$ or $r_2$ is $\frac{2b}{n}$, and the probability to pick the second agent in the other state is $\frac{b}{n-1}$, so the total probability of both is $\frac{2b^2}{n(n-1)}$. The case for $r_1 = r_2$ gives $\frac{b}{n}$ for the first times $\frac{b-1}{n-1}$ for the second, resulting in lower total probability $\frac{b^2-b}{n(n-1)}$.

[20]With a nondeterministic transition function, the total number of transitions would replace the quantity $|\Lambda|^2$ in the conclusion, but it would remain a constant independent of the size of the initial configuration.

[21]Theorem 4.3 was proven for a more general model called Chemical Reaction Networks (CRNs) that obey a certain technical condition [12]; as observed in that paper, the class of CRNs obeying that condition includes all PPs, so the theorem holds unconditionally for PPs. The theorem proved in [12] is more general than Theorem 4.3, but we have stated a corollary of it here. A similar statement is implicit in the proof sketch of Lemma 5 of a technical report on a variant model called "urn automata" that has PPs as a special case [3].

Lemma 4.4) using only combinatorial arguments about reachability between configurations.

For ease of notation, we assume throughout this paper that all states in $\Lambda$ are *producible*, meaning they have positive count in some reachable configuration. Otherwise the following theorem applies only to states that are actually producible. Recall that for $\alpha > 0$, a configuration $\mathbf{c}$ is $\alpha$-*dense* if for all $s \in \Lambda$, $\mathbf{c}(s) > 0$ implies that $\mathbf{c}(s) \geq \alpha \|\mathbf{c}\|$. Say that $\mathbf{c} \in \mathbb{N}^\Lambda$ is *full* if $(\forall s \in \Lambda)\, \mathbf{c}(s) > 0$, i.e., every state is present. The following theorem states that with high probability, a PP will reach from an $\alpha$-dense configuration to a configuration in which all states are present (full) in "high" count ($\beta$-dense, for some $0 < \beta < \alpha$).

**Theorem 4.3** (adapted from [12]). *Let $\mathcal{P} = (\Lambda, \delta)$ be a PP and $\alpha > 0$. Then there are constants $\epsilon, \beta > 0$ such that, letting $X = \left\{\, \mathbf{x} \in \mathbb{N}^\Lambda \mid \mathbf{x} \text{ is full and } \beta\text{-dense} \,\right\}$, for all $\alpha$-dense configurations $\mathbf{i}$, $\Pr[\mathbf{i} \Longrightarrow X] \geq 1 - 2^{-\epsilon \|\mathbf{i}\|}$.*

In [12], the theorem is stated for "sufficiently large" $\|\mathbf{i}\|$, but of course one can always choose $\epsilon$ to be small enough to make it true for all $\mathbf{i}$.

The following lemma reduces the problem of proving Theorem 3.7 to a combinatorial statement involving only reachability among configurations (and the lack of bottleneck transitions between them). In Section 5 we will prove Theorem 3.7 by showing that the existence of the configurations $\mathbf{x}_m$ and $\mathbf{y}_m$ and the transition sequence $p_m$ in the following lemma implies that we can reach a $Q$-stable configuration $\mathbf{v} \in \mathbb{N}^\Gamma$, where $\Gamma = \mathsf{unbdd}(Y)$ and $Y$ is the set of $Q$-stable configurations reachable from $I$.

**Lemma 4.4.** *Let $\alpha > 0$. Let $\mathcal{P} = (\Lambda, \delta)$ be a PP such that, for some set of transitions $Q$ and infinite set of $\alpha$-dense initial configurations $I$, $\mathcal{P}$ reaches a set of $Q$-stable configurations $Y$ in expected time $o(n)$. Then for all $m \in \mathbb{N}$, there is a configuration $\mathbf{x}_m$ reachable from some $\mathbf{i} \in I$ and transition sequence $p_m$ such that (1) $\mathbf{x}_m(s) \geq m$ for all $s \in \Lambda$, (2) $\mathbf{x}_m \Longrightarrow_{p_m} \mathbf{y}_m$, where $\mathbf{y}_m \in Y$, and (3) $p_m$ has no $m$-bottleneck transition.*

*Proof.* Intuitively, the lemma follows from the fact that states $\mathbf{x}_m$ are reached with high probability by Theorem 4.3, and if no paths such as $p_m$ existed, then all paths from $\mathbf{x}_m$ to a stable configuration would have a bottleneck and require linear time. Since $\mathbf{x}_m$ is reached with high probability, this would imply the entire expected time is linear.

For any configuration $\mathbf{x}_m$ reachable from some configuration in $I$, there is a transition sequence $p_m$ satisfying condition (2) by the fact that $\mathcal{P}$ $Q$-stabilizes from $I$. It remains to show we can find $\mathbf{x}_m$ and $p_m$ satisfying conditions (1) and (3).

By Theorem 4.3 there exist $\epsilon, \beta$ (which depend only on $\mathcal{P}$ and $\alpha$) such that, starting in any configuration $\mathbf{i}$, with probability at least $1 - 2^{\epsilon n}$, $\mathcal{P}$ reaches a configuration $\mathbf{x}$ where all states $s \in \Lambda$ have count at least $\beta n$, where $n = \|\mathbf{i}\|$. Choose $n$ to be large enough that $1 - 2^{\epsilon n} \geq \frac{1}{2}$ and there is $\mathbf{i}_n \in I$ such that $\|\mathbf{i}_n\| = n$. Then by Theorem 4.3, $\Pr[\mathbf{i}_n \Longrightarrow X_n] \geq \frac{1}{2}$ where $X_n = \left\{\, \mathbf{x} \mid \mathbf{i}_n \Longrightarrow \mathbf{x} \text{ and } (\forall s \in \Lambda)\, \mathbf{x}(s) \geq \beta n \,\right\}$.

Suppose that every transition sequence from every configuration $\mathbf{x} \in X_n$ to some $\mathbf{y} \in Y$ has a $b(n)$-bottleneck, for some function $b : \mathbb{N} \to \mathbb{N}$. Corollary 4.2 tells us that, then $\mathsf{T}[\mathbf{i}_n \Longrightarrow Y] \geq \frac{1}{2} \frac{n-1}{(b(n) \cdot \Lambda)^2}$. For this expected time to be $o(n)$, it must be that $b(n) = \omega(1)$. Let $\mathbf{x}_m$ be a configuration in $X_n$ that can reach $Y$ by a path without a $b(n)$-bottleneck transition, where the indexing $m$ satisfies $m \leq \min\{\lfloor \beta n \rfloor, b(n)\}$. Since $\min\{\lfloor \beta n \rfloor, b(n)\} = \omega(1)$, we have an appropriate configuration for all indexes $m \in \mathbb{N}$. Since $m \leq \lfloor \beta n \rfloor$, condition (1) is satisfied. Finally, since $m \leq b(n)$, condition (3) is satisfied. $\square$

## 4.3 Transition Ordering Lemma

The following lemma was first proven (in the more general model of Chemical Reaction Networks) in [8]. We provide a proof for the sake of self-containment. Intuitively, the lemma states that a "fast" transition sequence (meaning one without a bottleneck transition) that decreases certain states from large counts to small counts must contain transitions of a certain restricted form. In particular the form is as follows: if $\Delta$ is the set of states whose counts decrease from large to small, then we can write the states in $\Delta$ in some order $d_1, d_2, \ldots, d_k$, such that for each $1 \leq i \leq k$, there is a transition $\alpha_i$ that consumes $d_i$, and every other state involved in $\alpha_i$ is either not in $\Delta$, or comes later in the ordering. These transitions will later be used to do controlled "surgery" on fast transition sequences, because they give a way to alter the count of $d_i$, by inserting or removing the transitions $\alpha_i$, knowing that this will not affect the counts of $d_1, \ldots, d_{i-1}$.

**Lemma 4.5** (Adapted from [8]). *Let $b_1, b_2 \in \mathbb{N}$ such that $b_2 > |\Lambda| \cdot b_1$. Let $\mathbf{x}, \mathbf{y} \in \mathbb{N}^\Lambda$ such that $\mathbf{x} \Longrightarrow \mathbf{y}$ via transition sequence $q$ that does not contain a $b_2$-bottleneck. Define*

$$\Delta = \{ \, d \in \Lambda \mid \mathbf{x}(d) \geq b_2 \text{ and } \mathbf{y}(d) \leq b_1 \, \}.$$

*Then there is an order on $\Delta$, so that we may write $\Delta = \{d_1, d_2, \ldots, d_k\}$, such that, for all $i \in \{1, \ldots, k\}$, there is a transition $\alpha_i$ of the form $d_i, s_i \to o_i, o'_i$, such that $s_i, o_i, o'_i \notin \{d_1, \ldots, d_i\}$, and $\alpha_i$ occurs at least $(b_2 - |\Lambda| \cdot b_1)/|\Lambda|^2$ times in $q$.*

*Proof.* We define the ordering based on increasing sets $\varnothing = \Delta_0 \subset \Delta_1 \subset \Delta_2 \subset \ldots \Delta_{k-1} \subset \Delta_k = \Delta$, where for each $1 \leq i \leq k$, $\Delta_i = \Delta_{i-1} \cup \{d_i\}$.

We define the ordering inductively "in reverse," by first defining $d_k$, then $d_{k-1}$, etc. For all $1 \leq i \leq k$, define $\Phi_i : \mathbb{N}^\Lambda \to \mathbb{N}$ for all configurations $\mathbf{c}$ by $\Phi_i(\mathbf{c}) = \sum_{d \in \Delta_i} \mathbf{c}(d)$. $\Phi_k$ is well-defined since $\Delta_k = \Delta$, and $\Phi_i$ is well-defined once we have defined $d_{i+1}, \ldots, d_k$, because $\Delta_i = \Delta \setminus \{d_{i+1}, \ldots, d_k\}$.

Because $\mathbf{y}(d) \leq b_1$ for all $d \in \Delta$, it follows that $\Phi_i(\mathbf{y}) \leq i \cdot b_1 \leq |\Lambda| \cdot b_1$. Recall that $\mathbf{x}(d) \geq b_2$ for all $d \in \Delta$. Let $r$ be the suffix of $q$ after the last configuration $\mathbf{c}'$ along $q$ such that $\Phi_i(\mathbf{c}') \geq b_2$. Then in all configurations $\mathbf{c}$ in $r$, $\mathbf{c}(d) < b_2$ for all $d \in \Delta_i$. Because $\Phi_i(\mathbf{c}') \geq b_2$ but $\Phi_i(\mathbf{y}) \leq |\Lambda| \cdot b_1$, $r$ contains a subsequence $u$ of transitions, each of which strictly decreases $\Phi_i$, and the total decrease in $\Phi_i$ over all of $u$ is at least $(b_2 - |\Lambda| \cdot b_1)$ between configurations $\mathbf{c}'$ and $\mathbf{y}$.

Let $\alpha : r_1, r_2 \to p_1, p_2$ be a transition in $u$. Since $\alpha$ strictly decreases $\Phi_i$, $r_1 \in \Delta_i$ or $r_2 \in \Delta_i$; assume without loss of generality that $r_1 \in \Delta_i$. Further, since $u$ does not contain a $b_2$-bottleneck, and all configurations $\mathbf{c}$ along $u$ have $\mathbf{c}(d) < b_2$ for all $d \in \Delta_i$, for $\alpha$ not to be a $b_2$-bottleneck, we must have $r_2 \notin \Delta_i$. Since exactly one state in $\Delta_i$ decreases its count, $p_1 \notin \Delta_i$ and $p_2 \notin \Delta_i$, or else $\alpha$ would not decrease $\Phi_i$. Let $d_i = r_1, s_i = r_2, o_i = p_1$, and $o'_i = p_2$.

Then $\alpha$ decreases $\Phi_i$ by exactly 1. Since there are at least $b_2 - |\Lambda| \cdot b_1$ instances of such transitions in $u$, and there are at most $|\Lambda|^2$ total types of transitions, by the pigeonhole principle at least one transition type must repeat in $u$ at least $(b_2 - |\Lambda| \cdot b_1)/|\Lambda|^2$ times. $\square$

# 5 Proof of Theorem 3.7

By Lemma 4.4, there are sequences $(\mathbf{x}_m)$ and $(\mathbf{y}_m)$ of configurations, and a sequence $(p_m)$ of transition sequences, such that, for all $m$, (1) $\mathbf{x}_m(s) \geq m$ for all $s \in \Lambda$, and for some $\mathbf{i} \in I$, $\mathbf{i} \Longrightarrow \mathbf{x}_m$, (2) $\mathbf{y}_m$ is $Q$-stable, and (3) $\mathbf{x}_m \Longrightarrow_{p_m} \mathbf{y}_m$ and $p_m$ does not contain an $m$-bottleneck.

By Dickson's Lemma there is an infinite subsequence of $(\mathbf{x}_m)$ for which both $(\mathbf{x}_m)$ and $(\mathbf{y}_m)$ are nondecreasing. Without loss of generality, we take $(\mathbf{x}_m)$ and $(\mathbf{y}_m)$ to be these subsequences. Let $\Delta = \mathsf{bdd}(\mathbf{y}_m)$ and $\Gamma = \mathsf{unbdd}(\mathbf{y}_m)$.

To prove Theorem 3.7 we need to show that there are configurations in $Y$ (the set of $Q$-stable configurations reachable from $I$) that contain states only in $\Gamma$. Note that stability is closed downward: subsets of a $Q$-stable configuration are $Q$-stable. For any fixed $\mathbf{v}^\Gamma \in \mathbb{N}^\Gamma$, $\mathbf{v}^\Gamma \leq \mathbf{y}_m$ for sufficiently large $m$, by the definition of $\Gamma$ (the states that grow unboundedly in $\mathbf{y}_m$ as $m \to \infty$). Thus *any* configuration $\mathbf{v}^\Gamma \in \mathbb{N}^\Gamma$ is automatically $Q$-stable. This is why Claims 5.1, 5.2, and 5.3 of this proof center around reaching configurations that have count 0 of every state in $\Delta$.

Recall the path $\mathbf{x}_m \Longrightarrow_{p_m} \mathbf{y}_m$ from Lemma 4.4. Intuitively, Claim 5.1 below says that because this path is $m$-bottleneck free, Lemma 4.5 applies, and its transitions can appended to the path to consume all states in $\Delta$ from $\mathbf{y}_m$, resulting in a configuration $\mathbf{z}_m^\Gamma$ that contains only states in $\Gamma$. The "cost" of this manipulation is that, to ensure the appended transitions are applicable, we add extra agents in specific states corresponding to $\mathbf{e} \in \mathbb{N}^\Lambda$. Claim 5.1 is not sufficient to prove Theorem 3.7 because of this additional $\mathbf{e}$; the subsequent Claims 5.2 and 5.3 will give us the machinery to handle it.

**Claim 5.1.** *There is $\mathbf{e} \in \mathbb{N}^\Lambda$ such that for all large enough $m$, there is $\mathbf{z}_m^\Gamma \in \mathbb{N}^\Gamma$, such that $\mathbf{x}_m + \mathbf{e} \Longrightarrow \mathbf{z}_m^\Gamma$.*

**Example.** We illustrate Claim 5.1 through an example. Define a PP by the transitions

$$b, a \;\to\; f, c \tag{10}$$
$$b, c \;\to\; f, a \tag{11}$$
$$a, c \;\to\; f, f \tag{12}$$
$$f, c \;\to\; f, b \tag{13}$$
$$f, b \;\to\; f, f \tag{14}$$

For convenience, for state $s \in \Lambda$, let $s$ also denote the count of that state in the configuration considered. Let configuration $\mathbf{x}_m$ be where $f = 100$, $a = 100$, $b = 100$, $c = 100$. Suppose a transition sequence $p_m$ without an $m$-bottleneck ($m = 100$) takes the PP from $\mathbf{x}_m$ to $\mathbf{y}_m$, in which $a = 3$, $b = 2$, $c = 1$, and $f = 394$. Then in the language of Lemma 4.5, $\Delta = \{a, b, c\}$; these states go from "large" count in $\mathbf{x}_m$ to "small" count in $\mathbf{y}_m$.

Our strategy is to add interactions to $p_m$ in order to reach a configuration $\mathbf{z}_m^\Gamma$ with $a = b = c = 0$. There are two issues we must deal with. First, to get rid of $a$ we may try to add 3 instances of (10) at the end of $p_m$. However, there is only enough $b$ for 2 instances. To eliminate such dependency, in Claim 5.1, whenever we add a transition $b, a \to f, c$, we add an extra agent in state $b$ to $\mathbf{e}$. (In general if we consume $r_2$ by adding transition $r_1, r_2 \to p_1, p_2$, we add an extra agent in state $r_1$ to $\mathbf{e}$.) Second, we need to prevent circularity in consuming and producing states. Imagine trying to add more executions of (10) to get $a$ to 0 and more of (11) to get $c$ to 0; this will fail because these transitions conserve the quantity $a + c$. To drive each of these states to 0, we must find some ordering on them so that each can be driven to 0 using a transition that does not affect the count of any state previously driven to 0.

Lemma 4.5 gives us a way to eliminate such dependency systematically. In the example above, we can find the ordering $d_1 \equiv a$, $d_2 \equiv c$, and $d_3 \equiv b$, with respective transitions (10) to drive $a$ to 0 (3 executions), (13) to drive $c$ to 0 (4 executions: 1 to consume the 1 copy of $c$ in $\mathbf{y}_m$, and 3 more to consume the extra 3 copies that were produced by the 3 extra executions of (10)), and (14) to drive $b$ to 0 (6 executions: 2 to consume 2 copies of $b$ in $\mathbf{y}_m$, and 4 more to consume the extra 4 copies that were produced by the 4 extra executions of (13)).

*Proof.* Intuitively, the proof works as follows. Recall that $\mathbf{x}_m \Longrightarrow_{p_m} \mathbf{y}_m$ and $p_m$ does not contain an $m$-bottleneck. The goal is to get from configuration $\mathbf{y}_m$ (which may be positive on some elements of $\Delta$) to $\mathbf{z}_m^\Gamma$ (which is 0 on all elements of $\Delta$). We will show that we can append to the end of $p_m$ transitions $\alpha_i : d_i, s_i \to o_i, o_i'$, for $i \in \{1, \ldots, k\}$ — in that order — such that for all $i$, $d_i \in \Delta$ and $s_i, o_i, o_i' \notin \{d_1, \ldots, d_i\}$. We use Lemma 4.5 to find the necessary transitions. We add enough $\alpha_i$ transitions to consume all copies of $d_i$. However, this will also consume copies of $s_i$, so we add more copies of $s_i$ to $\mathbf{e}$ to account for this. Once we have added enough $\alpha_i$ transitions to make the count of $d_i$ equal to 0, by the fact that for all $j$, $s_j, o_j, o_j' \notin \{d_1, \ldots, d_j\}$, subsequently added transitions $\alpha_j$ for $j > i$ will not produce $d_i$ (so its count will stay 0), nor will it require consuming $d_i$ (so the transitions will be applicable). However, prior to reaching the point where we add $\alpha_i$ transitions, if $d_i = o_j$ or $o_j'$ for $j < i$, then the excess copies of $d_i$ generated by the extra $\alpha_j$ transitions mean that we may need to add more than $\mathbf{y}_m(d_i)$ copies of $\alpha_i$ to consume all the copies of $d_i$. The resulting configuration will be $\mathbf{z}_m^\Gamma \in \mathbb{N}^\Gamma$.

More formally, we  choose large enough $m$ such that the counts of species in $\Delta$ are no longer changing with $m$ in $(\mathbf{y}_m)$. Recall that $\mathbf{x}_m \Longrightarrow_{p_m} \mathbf{y}_m$ and $p_m$ does not contain an $m$-bottleneck. We'll apply Lemma 4.5 on this path with $b_2 = m$ and let $b_1$ be the largest count of any species in $\Delta$ anywhere in the sequence $(\mathbf{y}_m)$. (If necessary, increase $m$ further to ensure $b_2 > |\Lambda| \cdot b_1$.) Note that with these parameters, $\Delta = \mathsf{bdd}(\mathbf{y}_m)$ exactly matches the set of states "$\Delta$" defined in the statement of Lemma 4.5. Then this lemma tells us that there is an ordering on $\Delta$, so that we can write $\Delta = \{d_1, \ldots, d_k\}$, such that for each $1 \leq i \leq k$, there is a transition $\alpha_i : d_i, s_i \to o_i, o_i'$ such that $d_i \in \Delta$ and $s_i, o_i, o_i' \notin \{d_1, \ldots, d_i\}$.

We determine $\mathbf{e}$ and show how to consume all states in $\Delta$ using the above transitions in the following iterative procedure. Start with $\mathbf{e}_0 = \mathbf{0}$ and $\mathbf{o}_0 = \mathbf{0}$. Intuitively, $\mathbf{e}_i$ represents the total number of transitions, among $\alpha_1, \ldots, \alpha_{i-1}$, that we add that have $d_i$ as an input state. Therefore, that many extra copies of $d_i$ must be present in $\mathbf{e}$ to allow us to execute all of those transitions starting at the configuration $\mathbf{y}_m + \mathbf{e}$. $\mathbf{o}_i$ represents the total number of transitions, among $\alpha_1, \ldots, \alpha_{i-1}$, that we add that have $d_i$ as an *output* state (counting twice each time a transition has $d_i$ as both output states). Therefore, when we apply reaction $\alpha_i$ in order to consume all of $d_i$, we need to consume not just $\mathbf{y}_m(d_i)$, but $\mathbf{y}_m(d_i) + \mathbf{o}(d_i)$, since the previous added transitions that produce $d_i$ increased its count.

For all $i \in \{1, \ldots, k\}$, let

- $c_i = \mathbf{y}_m(d_i) + \mathbf{o}_{i-1}(d_i)$

- $\mathbf{e}_i = \mathbf{e}_{i-1} + \{c_i s_i\}$

- $\mathbf{o}_i = \mathbf{o}_{i-1} + \{c_i o_i, c_i o_i'\}$ (note that if $o_i = o_i'$, this is the same as writing $\{2c_i o_i\}$).

Given transition sequences $p$ and $q$, let $p \oplus q$ denote $q$ appended to the end of $p$, and given a single transition $\alpha$ and $j \in \mathbb{N}$, let $j \cdot \alpha$ denote the transition sequence consisting of $j$ copies of $\alpha$.

Let $\mathbf{e} = \mathbf{e}_k$. For all $i \in \{0, \ldots, k\}$, define $p_{m,i}$ inductively to be $p_{m,i-1} \oplus (c_i \cdot \alpha_i)$, where the base case is $p_{m,0} = p_m$, and define $\mathbf{z}_{m,i}$ inductively to be such that $\mathbf{x}_m + \mathbf{e}_i \Longrightarrow_{p_{m,i}} \mathbf{z}_{m,i}$, where the base case is $\mathbf{z}_{m,0} = \mathbf{y}_m$. Then $\mathbf{z}_m^\Gamma$ in the statement of the claim will be $\mathbf{z}_{m,k}$.

We prove by induction on $i$ that

1. for all $j \in \{1, \ldots, i\}$, $\mathbf{z}_{m,i}(d_j) = 0$,

2. $p_{m,i}$ is a valid transition sequence for $\mathbf{x}_m + \mathbf{e}_i$ (i.e., it never has a transition in a configuration in which the input states are not present), and

3. for all $j \in \{i+1, \ldots, k\}$, $\mathbf{z}_{m,i}(d_j) = \mathbf{y}_m(d_j) + \mathbf{o}_i(d_j)$ (in particular, $\mathbf{z}_{m,i}(d_{i+1}) = c_{i+1}$).

14

The base case is vacuous for 1 and immediate for 2 and 3 from the fact that $p_m$ is a valid transition sequence to apply to $\mathbf{x}_m$ and the definition of $c_1 = \mathbf{y}_m(d_i) + \mathbf{o}_0(d_i) = \mathbf{y}_m(d_i)$.

Assume the inductive case for $i-1$. Then by induction hypothesis 1, $\mathbf{z}_{m,i-1}(d_j) = 0$ for all $j \in \{1, \ldots, i-1\}$. Since $c_i \cdot \alpha_i$ does not have $d_j$ as an input or output state, and because $\mathbf{e}_i - \mathbf{e}_{i-1} = \{c_i s_i\}$ is 0 on $d_j$ (since $s_i \notin \{d_1, \ldots, d_{i-1}\}$), it follows that $\mathbf{z}_{m,i}(d_j) = 0$ for all $j \in \{1, \ldots, i-1\}$. By induction hypothesis 3, $\mathbf{z}_{m,i-1}(d_i) = c_i$, and since $c_i \alpha_i$ consumes exactly $c_i$ copies of $d_i$, this shows that $\mathbf{z}_{m,i}(d_i) = 0$, establishing the inductive case for 1.

Induction hypothesis 2 implies that $p_{m,i-1}$ is a valid transition sequence for $\mathbf{x}_m + \mathbf{e}_i$, since it is valid for $\mathbf{x}_m + \mathbf{e}_{i-1}$ by the hypothesis and $\mathbf{x}_m + \mathbf{e}_i \geq \mathbf{x}_m + \mathbf{e}_{i-1}$. We must show that all of $p_{m,i}$ is valid for $\mathbf{x}_m + \mathbf{e}_i$ by showing the final $c_i \cdot \alpha_i$ transitions on $p_{m,i}$, which apply to configuration $\mathbf{z}_{m,i-1}(d_i)$, are valid. By induction hypothesis 3, there is sufficient count of $d_i$ to apply the transitions. By the fact that $\mathbf{e}_i - \mathbf{e}_{i-1} = \{c_i s_i\}$, there is sufficient count of the other input state $s_i$ to apply the transitions. This establishes the inductive case for 2.

Let $j \in \{i, \ldots, k\}$. Induction hypothesis 3 gives that $\mathbf{z}_{m,i-1}(d_j) = \mathbf{y}_m(d_j) + \mathbf{o}_{i-1}(d_j)$. $\mathbf{o}_i(d_j) - \mathbf{o}_{i-1}(d_j)$ is 0 if $d_j$ is not an output state of $\alpha_i$, is $c_i$ if $d_j$ is exactly one output state, and is $2c_i$ if $d_j$ is both output states. Thus after applying $c_i$ copies of $\alpha_i$ to $\mathbf{z}_{m,i-1}$ to result in configuration $\mathbf{z}_{m,i}$, we have increased the count of $d_j$ by exactly $\mathbf{o}_i(d_j) - \mathbf{o}_{i-1}(d_j)$, resulting in $\mathbf{z}_{m,i}(d_j) = \mathbf{y}_m(d_j) + \mathbf{o}_{i-1}(d_j) + (\mathbf{o}_i(d_j) - \mathbf{o}_{i-1}(d_j)) = \mathbf{y}_m(d_j) + \mathbf{o}_i(d_j)$, proving the inductive case for 3.

Since inductive case 1 for the final value $k$ shows that, for all $j \in \{1, \ldots, k\}, \mathbf{z}_{m,k}(d_j) = \mathbf{z}_m^\Gamma(d_j) = 0$, this proves that $\mathbf{z}_m^\Gamma \in \mathbb{N}^\Gamma$. $\qquad\square$

Intuitively, Claim 5.2 below works toward generating the vector of states $\mathbf{e}$ that we needed for Claim 5.1. The vector $\mathbf{e}$ can be split into the $\Delta$ component and the $\Gamma$ component; we will handle the $\Gamma$ component later. The "cost" for Claim 5.2 is that the path must be taken "in the context" of additional agents in states captured by $\mathbf{p}$. Importantly, the net effect of the path preserves $\mathbf{p}$, which will give us a way to "interleave" Claims 5.1 and 5.2 as shown in Claim 5.3.

**Claim 5.2.** *For all $\mathbf{e}^\Delta \in \mathbb{N}^\Delta$, there is $\mathbf{p} \in \mathbb{N}^\Lambda$, such that for all large enough $m$, there is $\mathbf{w}_m^\Gamma \in \mathbb{N}^\Gamma$, such that $\mathbf{p} + \mathbf{x}_m \Longrightarrow \mathbf{p} + \mathbf{w}_m^\Gamma + \mathbf{e}^\Delta$, and $\mathsf{unbdd}(\mathbf{w}_m^\Gamma) = \Gamma$.*

**Example.** Recall the example above illustrating Claim 5.1. Claim 5.2 is more difficult than Claim 5.1 for two reasons. First, we need to be able to obtain any counts of states $a$, $b$, $c$ (ie $\mathbf{e}^\Delta$) and not only $a = b = c = 0$. Second, we no longer have the freedom to add extra states as $\mathbf{e}$ and consume them. Note that $\mathbf{p}$ cannot fulfill the same role as $\mathbf{e}$ because $\mathbf{p}$ must be recovered at the end.

For instance suppose $\mathbf{e}^\Delta$ is $a = 7$, $b = 2$, $c = 1$. Recall that $\mathbf{y}_m$ has $a = 3$, $b = 2$, $c = 1$. How can we generate additional 4 copies of $a$? Note that all transitions preserve or decrease the sum $a + b + c$. Thus we cannot solely add interactions to $p_m$ to get to our desired $\mathbf{e}^\Delta$. The key is that we can increase $a$ by removing existing interactions from $p_m$ that consumed it. Indeed, Lemma 4.5 helps us by giving a lower bound on the number of instances of transitions (10),(13),(14) that must have occurred in $p_m$. (Note that in Claim 5.1, we didn't need to use the fact that these transitions occurred in $p_m$. Now, we need to ensure that there are enough instances for us to remove.) In our case, we can remove 4 instances of interaction (10), which also decreases $c$ by 4. To compensate for this, we can remove 4 instances of interaction (13), which also decreases $b$ by 4. Finally, we remove 4 instances of interaction (14). The net result is that we reach the configuration $a = 7$, $b = 2$, $c = 1$, $f = 130$.

Note that unlike in Claim 5.1, we have more potential for circularity now because we cannot add the other input to a transition as $\mathbf{e}$. For example, we can't use transition (12) to affect $c$ because it affects $a$ (which we have previously driven to the desired count). Luckily, the ordering

given by Lemma 4.5 avoids any circularity because the other input and both of the outputs come later in the ordering.

Importantly, as we remove interactions from $p_m$, we could potentially drive the count of some state temporarily negative. Performing these interactions in the context of more agents ($\mathbf{p}$) ensures that the path can be taken.

*Proof.* Recall that $\mathbf{x}_m \Longrightarrow_{p_m} \mathbf{y}_m$ and $p_m$ does not contain an $m$-bottleneck. Intuitively, we will try to modify $p_m$ so that in the end we get exactly $\mathbf{e}^\Delta$ of $\Delta$. As in the proof of Claim 5.1, we will use the fact that $p_m$ does not contain an $m$-bottleneck and Lemma 4.5 to find transitions affecting $\Delta$ in a non-circular manner. However, unlike in Claim 5.1, we cannot simply consume an additional $\mathbf{e} \in \mathbb{N}^\Lambda$ to ensure that the count of the "other input state $s_i$" does not become negative. Rather, to increase the amounts of $s_i$ we will remove certain transition instances originally in $p_m$. It turns out that even with removing transitions, our modification to $p_m$ may still *temporarily* take certain species negative if we start from $\mathbf{x}_m$. However, executing the path in the context of $\mathbf{p}$ provides "buffer room" to ensure that no counts ever go below zero.

More formally, as in the proof of Claim 5.1 (see Appendix) apply Lemma 4.5 with $b_2 = m$ and let $b_1$ be the largest count of any species in $\Delta$ anywhere in the sequence ($\mathbf{y}_m$). The lower bound on $b_2 = m$ is determined below ("bound on the amount of fixing"). Lemma 4.5 tells us that there is an ordering on $\Delta$, so that we can write $\Delta = \{d_1, \ldots, d_k\}$, such that for each $1 \le i \le k$, there is a transition $\alpha_i : d_i, s_i \to o_i, o_i'$ such that $d_i \in \Delta$ and $s_i, o_i, o_i' \notin \{d_1, \ldots, d_i\}$, and $\alpha_i$ occurs at least $(b_2 - |\Lambda| \cdot b_1)/|\Lambda|^2$ times in $p_m$. Note that the final condition was not necessary to prove Claim 5.1 since its proof only added transitions to $p_m$. However, since this proof removes transitions as well, we require the condition to ensure that there are sufficiently many existing instances to be removed.

We iteratively fix the counts of states in $\Delta$ one by one, in the ordering given, i.e. we first adjust $p_m$ to fix $d_1$, then we fix $d_2$ (while showing that the fixing of $d_2$ cannot affect the count of $d_1$ in any configuration, so it remains fixed), etc. We start with $\mathbf{e}_0^\Delta(s) = \mathbf{y}_m(s)$ for $s \in \Delta$ and $\mathbf{e}_0^\Delta(s) = 0$ for $s \in \Gamma$, and $\mathbf{w}_{m,0}^\Gamma(s) = \mathbf{y}_m(s)$ for $s \in \Gamma$ and $\mathbf{w}_{m,0}^\Gamma(s) = 0$ for $s \in \Delta$. Having fixed $d_1, \ldots, d_{i-1}$, and obtaining new $\mathbf{e}_{i-1}^\Delta$, $\mathbf{w}_{m,i-1}^\Gamma$ (which could now be negative) such that $\mathbf{e}_{i-1}^\Delta$ agrees with the desired $\mathbf{e}^\Delta$ over $d_1, \ldots, d_{i-1}$, we process $d_i$ as follows. If $\delta_i = \mathbf{e}^\Delta(d_i) - \mathbf{e}_{i-1}^\Delta(d_i) < 0$: add $\delta_i$ instances of transition $\alpha_i$ at the end of the transition sequence. If $\delta_i > 0$: remove $\delta_i$ instances of $\alpha_i$ where they occur in the transition sequence; property (3) ensures that $q$ contains enough instances of $\alpha_i$ (see below). Let $\mathbf{e}_i^\Delta$ be the counts of the species in $\Delta$ at the end of this path. By property (2) and (3), adding or removing instances of $\alpha_i$ affects only the counts of states in $\Gamma$ and $d_{i+1}, \ldots, d_k$. Since we fix these counts in the prescribed order, when we are done, the counts of each $d_i$ is equal to its count in $\mathbf{e}^\Delta$ (ie $\mathbf{e}^\Delta = \mathbf{e}_k^\Delta$), while counts of elements of $\Gamma$ have been altered (letting $\mathbf{w}_m^\Gamma = \mathbf{w}_{m,k}^\Gamma$). We now claim that for large enough $m$, $\mathbf{w}_{m,k}^\Gamma$ is nonnegative, and that $\mathbf{p}$ can be independent of $m$. Finally, we derive a bound on the number of transition instances that we may need to remove, which determines another bound on $m$ (ie $b_2$) to ensure that there are enough instances by property (4) above.

Note that the amount of fixing we need to do only depends on the desired $\mathbf{e}^\Delta$ as well as on the counts of $\Delta$ states in $\mathbf{y}_m$. Because $\mathbf{y}_m$ are nondecreasing, and $\Delta = \mathsf{bdd}(\mathbf{y}_m)$, for large enough $m$, the counts of $\Delta$ states in $\mathbf{y}_m$ stop changing, and the amount of fixing depends only on the desired $\mathbf{e}^\Delta$. This implies that the $\mathbf{p}$ we need to add to ensure that no counts go negative can be independent of $m$. Further, for large enough $m$, the difference between $\mathbf{w}_m^\Gamma$ and $\mathbf{y}_m$ is independent of $m$, and thus $\mathsf{unbdd}(\mathbf{w}_m^\Gamma) = \mathsf{unbdd}(\mathbf{y}_m) = \Gamma$. This also implies that for large enough $m$, $\mathbf{w}_{m,k}^\Gamma$ is nonnegative.

*Bound on the amount of fixing:* Let $c_b = \max_{m \in \mathbb{N}, d \in \Delta} |\mathbf{y}_m(d) - \mathbf{e}^\Delta(d)|$ be the maximum amount

16

that any state in $\Delta$ deviates from its desired count. We add or remove at most $|\delta_1| \leq c_b$ instances of $\alpha_1$, which affects the count of states in $\Gamma \cup \{d_2, \ldots, d_k\}$ by at most $2c_b$ (it could be 2 per transition if the transition is $\alpha_1 : d_1, s \rightarrow s', s'$ for some state $s' \in \Lambda$). Thus, $|\delta_2| \leq c_b + 2|\delta_1|$ (the original $c_b$ error plus the additional error from altering the number of $\alpha_1$ transitions). In general, $|\delta_i| \leq c_b + 2(|\delta_1| + \cdots + |\delta_{i-1}|) \leq 3^{i-1}c_b$. Thus if we let $m = b_2 \geq k \cdot b_1 + 3^{k-1}c_b|\Lambda|^2$, we will have enough transition instances by property (4) to remove $((b_2 - |\Lambda| \cdot b_1)/|\Lambda|^2 = 3^{i-1}c_b)$. $\qquad\square$

**Claim 5.3.** *For infinitely many* $\mathbf{i} \in I$, *there is* $\mathbf{v}^\Gamma \in \mathbb{N}^\Gamma$ *such that* $\mathbf{i} \Longrightarrow \mathbf{v}^\Gamma$.

*Proof.* Intuitively, Claim 5.3 follows by expressing $\mathbf{i} = \mathbf{i}_1 + \mathbf{i}_2$ where $\mathbf{i}_1 \Longrightarrow \mathbf{x}_{m_1}$ and $\mathbf{i}_2 \Longrightarrow \mathbf{x}_{m_2}$, so $\mathbf{i} \Longrightarrow \mathbf{x}_{m_1} + \mathbf{x}_{m_2}$. We then apply Claim 5.2 to $\mathbf{x}_{m_2}$ (with $\mathbf{x}_{m_1}$ playing the role of $\mathbf{p}$) to get to a configuration with the correct $\mathbf{e}$ for Claim 5.1, and then apply Claim 5.1 to remove all states in $\Delta$.

By the hypothesis of Theorem 3.7, there are infinitely many pairs $\mathbf{i}_1, \mathbf{i}_2 \in I$ such that $\mathbf{i} = \mathbf{i}_1 + \mathbf{i}_2 \in I$. Let $m_1, m_2$ be such that $\mathbf{i}_1 \Longrightarrow \mathbf{x}_{m_1}$ and $\mathbf{i}_2 \Longrightarrow \mathbf{x}_{m_2}$, hence $\mathbf{i} \Longrightarrow \mathbf{x}_{m_1} + \mathbf{x}_{m_2}$; choose $m_1$ and $m_2$ large enough to satisfy the conditions stated below as they are needed. By Claim 5.1 (with $\mathbf{x}_{m_1}$), there is $\mathbf{e} \in \mathbb{N}^\Lambda$, and $\mathbf{z}^\Gamma \in \mathbb{N}^\Gamma$ such that $\mathbf{x}_{m_1} + \mathbf{e} \Longrightarrow_{p_{m_1}} \mathbf{z}^\Gamma$. Write $\mathbf{e}$ as $\mathbf{e} = \mathbf{e}^\Delta + \mathbf{e}^\Gamma$ where $\mathbf{e}^\Delta \in \mathbb{N}^\Delta$ and $\mathbf{e}^\Gamma \in \mathbb{N}^\Gamma$. Then apply Claim 5.2 (with $\mathbf{x}_{m_2}$) on $\mathbf{e}^\Delta$ (making sure $m_2$ is large enough to satisfy the claim on $\mathbf{e}^\Delta$). Thus, there is $\mathbf{p} \in \mathbb{N}^\Lambda$ and $\mathbf{w}^\Gamma \in \mathbb{N}^\Gamma$ such that $\mathbf{p} + \mathbf{x}_{m_2} \Longrightarrow_{p_{m_1}} \mathbf{p} + \mathbf{w}^\Gamma + \mathbf{e}^\Delta$. If $m_1$ is large enough that $\mathbf{x}_{m_1} \geq \mathbf{p}$, then $\mathbf{x}_{m_1} + \mathbf{x}_{m_2} \Longrightarrow_{p_{m_2}} \mathbf{x}_{m_1} + \mathbf{w}^\Gamma + \mathbf{e}^\Delta$. If further $m_2$ is large enough that $\mathbf{w}^\Gamma \geq \mathbf{e}^\Gamma$ (recall $\mathsf{unbdd}(\mathbf{w}^\Gamma) = \Gamma$), then we can rewrite the right hand side as $\mathbf{x}_{m_1} + (\mathbf{w}^\Gamma - \mathbf{e}^\Gamma) + \mathbf{e}$. Then, by Claim 5.1, $\mathbf{x}_{m_1} + (\mathbf{w}^\Gamma - \mathbf{e}^\Gamma) + \mathbf{e} \Longrightarrow_{p_{m_1}} (\mathbf{w}^\Gamma - \mathbf{e}^\Gamma) + \mathbf{z}^\Gamma$. Since the right hand side is only over $\Gamma$, we let $\mathbf{v}^\Gamma = (\mathbf{w}^\Gamma - \mathbf{e}^\Gamma) + \mathbf{z}^\Gamma$. $\qquad\square$

Finally, Theorem 3.7 is proven because $\mathbf{v}^\Gamma$ is $Q$-stable and it contains zero count of states in $\Delta$. To see that $\mathbf{v}^\Gamma$ is $Q$-stable recall that $\mathbf{v}^\Gamma \leq \mathbf{y}_{m'}$ for sufficiently large $m'$ since $\Gamma = \mathsf{unbdd}(y_m)$ and $\mathbf{v}^\Gamma$ contains only states in $\Gamma$. Since stability is closed downward, and $\mathbf{y}_{m'}$ is $Q$-stable, we have that $\mathbf{v}^\Gamma$ is $Q$-stable as well.

# References

[1] Dan Alistarh and Rati Gelashvili. Polylogarithmic-time leader election in population protocols. In *ICALP 2015: Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, Kyoto, Japan*, 2015.

[2] Dana Angluin, James Aspnes, Zoë Diamadi, Michael Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18:235–253, 2006. Preliminary version appeared in PODC 2004.

[3] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Urn automata. Technical Report YALEU/DCS/TR-1280, Yale University, November 2003.

[4] Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199, September 2008. Preliminary version appeared in DISC 2006.

[5] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.

[6] Dana Angluin, James Aspnes, Michael J Fischer, and Hong Jiang. Self-stabilizing population protocols. In *Principles of Distributed Systems*, pages 103–117. Springer, 2006.

[7] James M Bower and Hamid Bolouri. *Computational modeling of genetic and biochemical networks*. MIT press, 2004.

[8] Ho-Lin Chen, Rachel Cummings, David Doty, and David Soloveichik. Speed faults in computation by chemical reaction networks. In *DISC 2014: Proceedings of the 28th International Symposium on Distributed Computing, Austin, TX, USA*, pages 16–30, 2014.

[9] Yuan-Jyue Chen, Neil Dalchau, Niranjan Srinivas, Andrew Phillips, Luca Cardelli, David Soloveichik, and Georg Seelig. Programmable chemical controllers made from DNA. *Nature Nanotechnology*, 8(10):755–762, 2013.

[10] Inês Cunha-Ferreira, Inês Bento, and Mónica Bettencourt-Dias. From zero to many: control of centriole number in development and disease. *Traffic*, 10(5):482–498, 2009.

[11] Leonard E. Dickson. Finiteness of the odd perfect and primitive abundant numbers with $n$ distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, October 1913.

[12] David Doty. Timing in chemical reaction networks. In *SODA 2014: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 772–784, January 2014.

[13] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[14] Richard M Karp and Raymond E Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.

[15] Carl A Petri. Communication with automata. Technical report, DTIC Document, 1966.

[16] David Soloveichik, Georg Seelig, and Erik Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393, 2010. Preliminary version appeared in DNA 2008.

[17] Vito Volterra. Variazioni e fluttuazioni del numero dindividui in specie animali conviventi. *Mem. Acad. Lincei Roma*, 2:31–113, 1926.