Output-Oblivious Stochastic Chemical Reaction Networks

Ben Chugg[1]
The University of British Columbia, Canada
ben.ih.chugg@gmail.com

Hooman Hashemi
The University of British Columbia, Canada
hhoomn390@gmail.com

Anne Condon[2]
The University of British Columbia, Canada
condon@cs.ubc.ca
https://orcid.org/0000-0003-1458-1259

Abstract

We classify the functions $f : \mathbb{N}^2 \to \mathbb{N}$ which are stably computable by *output-oblivious* Stochastic Chemical Reaction Networks (CRNs), i.e., systems of reactions in which output species are never reactants. While it is known that precisely the semilinear functions are stably computable by CRNs, such CRNs sometimes rely on initially producing too many output species, and then consuming the excess in order to reach a correct stable state. These CRNs may be difficult to integrate into larger systems: if the output of a CRN $\mathcal{C}$ becomes the input to a downstream CRN $\mathcal{C}'$, then $\mathcal{C}'$ could inadvertently consume too many outputs before $\mathcal{C}$ stabilizes. If, on the other hand, $\mathcal{C}$ is output-oblivious then $\mathcal{C}'$ may consume $\mathcal{C}$'s output as soon as it is available. In this work we prove that a semilinear function $f : \mathbb{N}^2 \to \mathbb{N}$ is stably computable by an output-oblivious CRN with a leader if and only if it is both increasing and either *grid-affine* (intuitively, its domains are congruence classes), or the minimum of a finite set of *fissure functions* (intuitively, functions behaving like the min function).

2012 ACM Subject Classification Theory of computation → Computability, Theory of computation → Formal languages and automata theory

Keywords and phrases Chemical Reaction Networks, Stable Function Computation, Output-Oblivious, Output-Monotonic

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2018.21

Related Version A full version of the paper is available at `https://arxiv.org/abs/1812.04401`.

1 Introduction

Stochastic Chemical Reaction Networks (CRNs) – systems of reactions involving chemical species – have traditionally been used to reason about extant physical systems, but are currently also of strong interest as a distributed computing model for describing molecular programs [8, 16]. They are closely related to Population Protocols [1, 3, 4, 9], another

[1] Supported by an NSERC Undergraduate Student Research Award.
[2] Supported by an NSERC Discovery Grant.

distributed computing model; these models have found applications in areas as diverse as signal processing [13], graphical models [14], neural networks [12], and modeling cellular processes [5, 6]. CRNs can simulate Universal Turing Machines [16, 2]. However, these simulations have drawbacks: the number of reactions or molecules may scale with the space usage and the computation is only correct with an arbitrarily small probability of error. If we require *stable computation* – that the CRN always eventually produces the correct answer – then Angluin et al. [4] showed that precisely the class of semilinear predicates can be stably computed. Chen et al. [8] extended this result to show that precisely the semilinear functions can be stably computed.

Recent advances in physical implementations of CRNs and, more generally, chemical computation using strand displacement systems (e.g., [15, 17, 18, 19]) are a step towards the use of CRNs in biological environments and nanotechnology. As these systems become more complex, it may be necessary to integrate multiple, interacting CRNs in one system. However, current CRN constructions may perform poorly in such scenarios. As a concrete example, consider a CRN $\mathcal{C}$ given by the reactions $X \to 2Y$, $Y + L \to \emptyset$, where the system begins with n copies of input species X, and one copy of L (called the *leader*). This CRN eventually produces $2n - 1$ copies of output species Y, and so (stably) computes the function $n \mapsto 2n - 1$. If another CRN $\mathcal{C}'$ uses the output of $\mathcal{C}$ as its input, and if the first reaction occurs n times before the second occurs at all, then $\mathcal{C}'$ may consume all $2n$ copies of Y and may thus itself produce an erroneous output. Current CRN constructions circumvent this issue by using *diff-representation*, where the count y of output species Y of a CRN is represented indirectly as the difference $y = y^P - y^C$ between the counts of two species Y^P and Y^C [8], rather than as the count of one output species Y. While these constructions enable the counts of both Y^C and Y^P to be non-decreasing throughout the computation, it is not immediately clear how a second CRN might use these two species reliably as input.

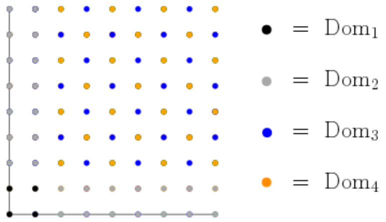
More generally, if multiple function-computing CRNs comprise a larger system it can be desirable that no CRN ever produces a number of outputs that exceeds its function value. We might even demand more: that an output species of a CRN is never used as a reactant species, i.e., is never consumed. This ensures that any secondary CRN relying on the first's output can consume the output indiscriminately.

It is thus natural to ask: What functions can be stably computed in an *output-oblivious manner*, in which outputs are never reactants, without using diff-representation?

This question is the focus of this paper. Doty and Hajiaghayi [11] already observed that output-oblivious functions must not only be semilinear but also increasing, that is, $f(\mathbf{n}_1) \leq f(\mathbf{n}_2)$ whenever $\mathbf{n}_1 \leq \mathbf{n}_2$, but did not provide further insights. Chalk et al. [7] asked the same question but for a different model, namely mass-action CRNs. That model tracks real-valued species concentrations, unlike the stochastic model in which configurations are vectors of species counts. In contrast with the mass-action mode, leader molecules can play a very important role in the stochastic model, and we focus on the case where leaders are present. Mass-action CRN models cannot have leaders since there are no species counts. Functions that are stably computable by output-oblivious mass-action CRNs must be super-additive [7], that is $f(n) + f(n') \leq f(n + n')$. Semilinear functions that are super-additive are a proper subset of the class of output-oblivious functions (characterized in this paper) that can be stably computed by stochastic CRNs with leaders.

1.1 Our Results

In this work we characterize the class of output-oblivious semilinear functions, i.e., those functions that can be stably computed by an output-oblivious stochastic CRN. We assume that one copy of a leader species is present initially in addition to the input. We focus on



$$f(\mathbf{n}) = \begin{cases} \varphi_1(\mathbf{n}) = 2, & \mathbf{n} \in \mathrm{Dom}_1, \\ \varphi_2(\mathbf{n}) = n_1 + n_2, & \mathbf{n} \in \mathrm{Dom}_2, \\ \varphi_3(\mathbf{n}) = n_1 + 2n_2 + 5, & \mathbf{n} \in \mathrm{Dom}_3, \\ \varphi_4(\mathbf{n}) = n_1 + 2n_2 + 4, & \mathbf{n} \in \mathrm{Dom}_4\,. \end{cases}$$

Figure 1 Here we represent a grid-affine function $f : \mathbb{N}^2 \to \mathbb{N}$ by its decomposition on different domains, all of which are grids. The domains of f are illustrated on the left. Each black point is a zero-dimensional grid, while the grey points represent four one-dimensional grids, namely the lines $\{(\alpha, 0) + (2, i) : \alpha \in \mathbb{N}\}$ and $\{(0, \alpha) + (i, 2) : \alpha \in \mathbb{N}\}$ for $i = 0, 1$. The blue points represent points (n_1, n_2) such that $n_1 + n_2$ is even, and cover the union of two grids: $\{(2\alpha_1, 2\alpha_2) : \alpha_i \in \mathbb{N}\} \cup \{(2\alpha_1, 2\alpha_2) + (1, 1) : \alpha_i \in \mathbb{N}\}$. Similarly, the gold points represent two grids.

functions with two inputs and one output, since this case already is quite complex. Our results generalize trivially when there are more outputs since each output can be handled independently, and we believe that our techniques also generalize to multiple inputs.

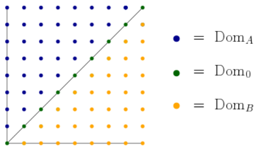
Our results also hold for Population Protocols, since stable function-computing CRNs can be translated into Population Protocols and vice versa. Section 2 introduces the relevant background in order to formally describe our results, but we describe them informally here.

Perhaps the simplest type of output-oblivious function with domain $\mathbb{N}^2$ is an affine function, such as $f(n_1, n_2) = 2n_1 + 3n_2 + 1$ which could be computed by a CRN with reactions $L \to Y$, $X_1 \to 2Y$ and $X_2 \to 3Y$ where L is a single leader. Here and hereafter, X_i will typically correspond to the input species representing n_i.

In Section 3 we show that an increasing function that can be specified as partial affine functions whose domains are different "grids" of $\mathbb{N}^2$ is also output-oblivious; for example, the function $f(n_1, n_2) = 2n_1 + 3n_2 + 1$ when $n_1 + n_2 = 0 \mod 2$, and $f(n_1, n_2) = 2n_1 + 3n_2$ when $n_1 + n_2 = 1 \mod 2$. More generally, a function that can be specified in terms of output-oblivious partial functions f_i, $1 \leq i \leq k$, defined on different grids of $\mathbb{N}^2$, is output-oblivious. The grids may be 0-dimensional, in which case they are points; 1-dimensional in which case they are lines, or 2-dimensional. We call such functions *grid-affine* functions. See Figure 1 for a slightly more complicated example of a grid-affine function, and a representation of its domains. We show how the CRNs for partial functions f_i on the different grids can be "stitched" together to obtain an output-oblivious CRN for f.

It is also straightforward to obtain an output-oblivious CRN for a function f that is the min of a finite set of output-oblivious functions. In the simplest case, for example, $\min(n_1, n_2)$ can be computed as $X_1 + X_2 \to Y$. In our main positive result we describe a more general type of "min-like" function, which we call a *fissure function*, and we show how to construct output-oblivious CRNs for such functions. We give a very simple example of a fissure function and a corresponding output-oblivious CRN in Figure 2.

However, constructing CRNs for other fissure functions appears to be significantly trickier than that shown in Figure 2. Consider the function $f(n_1, n_2) = 2n_1 + 3n_2 + 2$ if $n_1 > n_2$, $f(n_1, n_2) = 3n_1 + 2n_2 + 2$ if $n_1 < n_2$ and $f(n_1, n_2) = 5n_1$ on the "fissure line" $n_1 = n_2$. The simple line-tracking mechanism of the CRN of Figure 2 can't be used here because the affine functions for the "wedge" domains "$n_1 > n_2$" and "$n_1 < n_2$" depend both on n_1 and n_2. Also the function cannot be written as the sum of an increasing grid-affine function and an increasing simple fissure function of the type in Figure 2, where the "above" function $\varphi_A()$ depends only on n_1 and the "below" function $\varphi_B()$ depends only on n_2. Our main positive result is a construction that can handle such fissure functions, as well as functions



$$f(\mathbf{n}) = \begin{cases} \varphi_A(\mathbf{n}) = n_1 + 1, & \mathbf{n} \in \mathrm{Dom}_A, \\ \varphi_0(\mathbf{n}) = n_1, & \mathbf{n} \in \mathrm{Dom}_0, \\ \varphi_B(\mathbf{n}) = n_2 + 1, & \mathbf{n} \in \mathrm{Dom}_B\,. \end{cases}$$

$$\begin{aligned} [0] + X_1 &\to [A] + Y \\ [0] + X_2 &\to [B] + Y \\ [A] + X_2 &\to [0] \\ [B] + X_1 &\to [0] \end{aligned}$$

Figure 2 A simple fissure function $f : \mathbb{N}^2 \to \mathbb{N}$. On the left the three domains of f are illustrated. There is one "fissure line" called Dom_0, and two "wedge" domains called Dom_A and Dom_B ("A" is above and "B" is below the fissure line). The function value on each of these domains is specified in the center. The function f agrees with the function $\min\{n_1 + 1, n_2 + 1\}$ except that it dips down by 1 on the fissure line Dom_0. On the right is a CRN which stably computes f. In the CRN, the input $\mathbf{n} = (n_1, n_2)$ is represented as counts of species X_1 and X_2 and the leader is initially $[0]$. The three possible states $[0], [A]$ and $[B]$ of the leader track whether the input lies on the fissure line Dom_0, which is the line where $\varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = 0$, or whether the input lies above or below the fissure line, i.e., in domains Dom_A or Dom_B respectively. In this simple example, the CRN need not track how far above (or how far below) the fissure line an input might be, since the function φ_A does not depend on n_2 (and the function φ_B does not depend on n_1).

with multiple parallel fissure lines.

In Section 4 we present results on the negative side. A non-trivial example of a function that is not output-oblivious is the maximum function. Intuitively, a CRN that attempts to compute the max would have to keep track of the relative difference of its two inputs in order to know when the count of one input overtakes the count of the other, and it's not possible to keep track of that difference with a finite number of states. Developing this intuition further, we show that an increasing semilinear function $f : \mathbb{N}^2 \to \mathbb{N}$ is output-oblivious if and only if f is grid-affine or is the min of finitely many fissure functions.

Putting both positive and negative results together, we state our main result here (see Section 2 for precise definitions of grid-affine and fissure functions).

▶ **Theorem 1.** *A semilinear function $f : \mathbb{N}^2 \to \mathbb{N}$ is output-oblivious if and only if f is increasing and is either grid-affine or the minimum of finitely many fissure functions.*

Since only semilinear functions are stably computable by CRNs, Theorem 1 provides a complete characterization of functions $f : \mathbb{N}^2 \to \mathbb{N}$ which are output-oblivious. Moreover, in Section 4, we will prove that if f is output-*monotonic*, then it is either grid-affine or the minimum of fissure functions, a stronger statement than in Theorem 1. A function is output-monotonic if it is stably computable by a CRN whose output count never decreases but unlike an output-oblivious CRN an output may act as a catalyst of a reaction, being both a reactant and product. For example, the CRN $X \to Y$, $L + Y \to 2Y$ which computes the function $n \mapsto n + 1$ for $n \geq 1$ and $0 \mapsto 0$ is output-monotonic, but not output-oblivious. Thus, we also obtain a characterization for output-monotonic functions.

To obtain our results, we provide new characterizations of semilinear sets and functions. We show that all semilinear sets can be written as finite unions of sets which are the intersection of grids and hyperplanes. Such sets are points, lines or wedges (pie-shaped slices) on 2D grids. Using this and the representation of semilinear functions as piecewise affine functions discovered by Chen et al. [8], we give a new representation of semilinear functions as “periodic semiaffine functions”, essentially piecewise affine functions whose domains are points, lines or wedges.

The rest of the paper is structured as follows. Section 2 provides the relevant technical background on CRNs, stable computation and semilinear functions. It also contains our new results on the structure of semilinear sets and functions, and rigorous definitions of grid-affine and fissure functions. In the remaining two sections we prove Theorem 1, with Section 3 providing explicit constructions of CRNs and Section 4 proving that any function which is stably computable by an output-oblivious CRN obeys certain properties. Some proofs are omitted and will appear in the full version of the paper.

2 Preliminaries

We begin by introducing Chemical Reaction Networks, and what it means for a CRN to stably compute a function. We then formally define grid-affine and fissure functions and, along the way, state new results concerning semilinear sets and functions.

2.1 Chemical Reaction Networks (CRNs)

CRNs specify possible behaviours of systems of interacting *species*. Let $\mathcal{Z} = \{Z_1, \ldots, Z_m\}$ be a finite set of species. At any given instant, the system is described by a configuration $\mathbf{c} \in \mathbb{N}^{\mathcal{Z}}$, where $c(Z_i)$ is the current count of the species $Z_i \in \mathcal{Z}$ in the system. The system's configuration changes by way of *reactions*, each of which is described as a pair $(\mathbf{s}, \mathbf{t}) = ((s_1, \ldots, s_m), (t_1, \ldots, t_m)) \in \mathbb{N}^{\mathcal{Z}} \times \mathbb{N}^{\mathcal{Z}}$ such that for at least one $1 \le j \le m$, $s_j \ne t_j$. Reaction $(\mathbf{s}, \mathbf{t})$ can be written as

$$\sum_{k:s_k>0} s_k Z_k \to \sum_{k:t_k>0} t_k Z_k.$$

The species Z_k with $s_k > 0$ are the *reactants*, which are *consumed*, while those with $t_k > 0$ are the *products* (if both $s_k > 0$ and $t_k > 0$ then species Z_k is a catalyst). A CRN is thus formally described as a pair $\mathcal{C} = (\mathcal{Z}, \mathcal{R})$, where $\mathcal{Z}$ is a set of species, and $\mathcal{R}$ a set of reactions. Reaction $r = (\mathbf{s}, \mathbf{t})$ is *applicable* to configuration $\mathbf{c}$ if $\mathbf{s} \le \mathbf{c}$ (pointwise inequality), i.e., sufficiently many copies of each reactant are present. If applicable reaction $(\mathbf{s}, \mathbf{t})$ occurs when the system is in configuration $\mathbf{c} = (c_1, \ldots, c_m)$, the new configuration is $\mathbf{c}' = (c_1 - s_1 + t_1, \ldots, c_m - s_m + t_m)$. In this case we say that $\mathbf{c}'$ is *directly reachable* from $\mathbf{c}$ and write $\mathbf{c} \xrightarrow{r} \mathbf{c}'$. An *execution* $\mathcal{E} = \mathbf{c}_0, \ldots, \mathbf{c}_t$ of $\mathcal{C}$ is a sequence of configurations of $\mathcal{C}$ such that $\mathbf{c}_i$ is directly reachable from $\mathbf{c}_{i-1}$ for $1 \le i \le t$. We say that $\mathbf{c}_t$ is *reachable* from $\mathbf{c}_0$.

Stable CRN Computation of functions with a leader. Angluin et al. [3] introduced the concept of stable computation of boolean predicates by population protocols, and Chen et al. [8] adapted the notion to function computation by CRNs. While this paper focuses on two-dimensional domains, we present the following details in full generality.

Let $f : \mathbb{N}^k \to \mathbb{N}^\ell$ be a function. Formally, a *Chemical Reaction Network (CRN) for computing f with a leader* is $\mathcal{C} = (\mathcal{Z}, \mathcal{R}, \mathcal{I}, \mathcal{O}, L)$, where $\mathcal{Z}$ is a set of species, $\mathcal{R}$ is a set of reactions, $\mathcal{I} = \{X_1, X_2, \ldots, X_k\} \subseteq \mathcal{Z}$ is an ordered set of input species, $\mathcal{O} = \{Y_1, Y_2, \ldots, Y_\ell\} \subseteq \mathcal{Z}$ is an ordered set of output species and L is a leader species, $L \in \mathcal{Z} \setminus \mathcal{I}$.

Function computation on input $\mathbf{n} = (n_1, \ldots, n_k) \in \mathbb{N}^k$ starts from a *valid initial configuration* $\mathbf{c}_0$ of $\mathcal{C}$; namely a configuration in which the count of L is 1, the count of species X_i is n_i, and the count of any other species is 0. A *computation* is an execution of $\mathcal{C}$ from a valid initial configuration to a stable configuration. A configuration c is *stable* if for every $\mathbf{c}' \in \mathbb{N}^m$ reachable from $\mathbf{c}$, $\mathbf{c}(Y) = \mathbf{c}'(Y)$ for all $Y \in \mathcal{O}$. That is, once the system reaches configuration $\mathbf{c}$, the counts of the output species do not change. We say that $\mathcal{C}$ *stably computes* f if for every valid initial configuration $\mathbf{c}_0$ and for every configuration $\mathbf{c}$ reachable from $\mathbf{c}_0$, there exists a stable configuration $\mathbf{c}'$ reachable from $\mathbf{c}$ such that $f(\mathbf{c}_0(X_1), \ldots, \mathbf{c}_0(X_k)) = (\mathbf{c}'(Y_1), \ldots, \mathbf{c}'(Y_\ell))$.

Output-monotonic and output-oblivious CRNs. We say a CRN $\mathcal{C}$ is *output-oblivious* if it never consumes any of its output species, and *output-monotonic* if on all executions from a valid initial configuration, the count of any output species never decreases. As noted in the introduction, these notions are not equivalent. We say a function f is *output-oblivious* (*monotonic*) if there exists an output-oblivious (monotonic) CRN which stably computes f. Our results show that the set of output-oblivious functions and output-monotonic functions are the same.

2.2 Linear and Semilinear Sets; Lines, Grids, and Wedges

For a vector $\mathbf{v}$, let v_i denote its ith coordinate. Let $D \subseteq \mathbb{N}^2$ and let Π_1 and Π_2 denote the projection maps onto x and y axes, respectively. We say D is *two-way-infinite* if $|\Pi_1(D)| = |\Pi_2(D)| = \infty$, *one-way-infinite* if either $|\Pi_1(D)| = \infty$ or $|\Pi_2(D)| = \infty$ but not both, and finite if $|\Pi_1(D)| < \infty$ and $|\Pi_2(D)| < \infty$. Also, if $A, B \subseteq \mathbb{N}^2$ and $\mathbf{n} \in \mathbb{N}^2$ we let $A + B = \{a + b : a \in A, b \in B\}$ and $A + \mathbf{n} = A + \{\mathbf{n}\}$.

A set $E \subset \mathbb{N}^2$ is *linear* if $E = \{\sum_{i=1}^{t} \mathbf{x}_i \alpha_i + \mathbf{o} : \alpha_i \in \mathbb{N}\}$ for some $t \in \mathbb{N}$ and $\mathbf{x}_i, \mathbf{o} \in \mathbb{N}^2$. If $t = 1$ we say that E is a *line*. A set is *semilinear* if it is the finite union of linear sets.

A linear set $\mathcal{G} \subseteq \mathbb{N}^2$ is a *grid* if there exist $p, q \in \mathbb{N}$ and $\mathbf{o} \in \mathbb{N}^2$ such that $\mathcal{G} = \{(p, 0)\alpha_1 + (0, q)\alpha_2 : \alpha_i \in \mathbb{N}\} + \mathbf{o} = \{(p\alpha_1 + o_1, q\alpha_2 + o_2) : \alpha_i \in \mathbb{N}\}$. If both p and q are zero, the grid is simply the point $\mathbf{o}$. If $p > 0$ and $q = 0$, or $p = 0$ and $q > 0$, the grid is a one-way-infinite line with period p or q respectively. If $p = q > 0$ we say that the grid is periodic, with period p. We let $\mathcal{G}_p + \mathbf{o}$ be the grid $\{(\alpha_1 p, \alpha_2 p) : \alpha_i \in \mathbb{N}\} + \mathbf{o}$ and write $\mathcal{G}_p$ if $\mathbf{o} = (0, 0)$.

A *threshold set* is a linear set with the form $\{\mathbf{x} : \mathbf{x} \cdot \mathbf{v} \geq r\}$ (i.e., a halfspace) for some $\mathbf{v} \in \mathbb{Z}^2$ and $r \in \mathbb{Z}$. Let E be a two-way-infinite linear set of the form $\mathcal{G} \cap \mathcal{T}$, where $\mathcal{G}$ is a grid and $\mathcal{T}$ is a finite intersection of threshold sets. E is bounded by two lines (represented by threshold sets and/or the x or y axes; the points on these lines, if any, are in E). If the two bounding lines are parallel, E is the finite union of lines on $\mathcal{G}$, i.e., all points of each line lie on grid $\mathcal{G}$. Otherwise we call E a *wedge* on $\mathcal{G}$. For example, the sets $\{\mathbf{n} : n_1 \geq n_2\}$ and $\{(1, 1)\alpha_1 + (1, 2)\alpha_2 : \alpha_i \in \mathbb{N}\}$ are wedges on $\mathcal{G}_1$. Likewise, the two regions above and below the fissure line in Figure 2 are wedges on $\mathcal{G}_1$. More generally, we can intuitively think of a wedge as a pie-like slice of $\mathbb{N}^2 \cap \mathcal{G}$, except that pieces may be chopped off near the narrow "corner" that is closest to the origin. If the two bounding lines are the x and y axes, the wedge is all of $\mathcal{G}$. We can show the following characterization of semilinear sets.

► **Lemma 2.** *Every semilinear set can be represented as the finite union of points, lines on grids, and wedges on grids, with all grids having the same period.*

2.3 Semilinear, Semiaffine, Grid-Affine, and Fissure Functions

For a function $f : \mathbb{N}^2 \to \mathbb{N}$, the *restriction of f to domain $D \subseteq \mathbb{N}^2$* is the partial function $f|_D : D \to \mathbb{N}$ given by $f|_D(\mathbf{n}) = f(\mathbf{n})$ for all $\mathbf{n} \in D$. We say that $f : D \to \mathbb{N}$ is *(partial) affine* if $f(\mathbf{n}) = a_1 n_1 + a_2 n_2 + a_0$ for rational numbers a_0, a_1, and $a_2 \in \mathbb{Q}$. Function f is a *finite combination* of the finite set of functions $\{\varphi_1, \ldots, \varphi_k\}$ if $\mathrm{Dom}(f) = \bigcup_{i=1}^{k} \mathrm{Dom}(\varphi_i)$ and $f(\mathbf{n}) = \varphi_i(\mathbf{n})$ whenever $\mathbf{n} \in \mathrm{Dom}(\varphi_i)$. Throughout we write Dom_i in place of $\mathrm{Dom}(\varphi_i)$. We define semilinear functions using a characterization of Chen et al. [8]:

► **Definition 3** (Semilinear function [8]). A function $f : \mathbb{N}^2 \to \mathbb{N}$ is *semilinear* if and only if f is a finite combination of partial affine functions with linear domains.

We next define *semiaffine* functions, a refinement of Definition 3. Lemma 5 then states that semilinear and semiaffine functions are equivalent.

▶ **Definition 4** (Semiaffine function). Let $\mathcal{G}_p + \mathbf{o}$ be a periodic grid. A function $f : \mathcal{G}_p + \mathbf{o} \to \mathbb{N}$ is *semiaffine* if and only if f is a finite combination of partial affine functions whose domains are points, lines or wedges on grid $\mathcal{G}_p + \mathbf{o}$. A function $f : \mathbb{N}^2 \to \mathbb{N}$ is *semiaffine with period* $p \in \mathbb{N}^+$ if and only if f is a combination of semiaffine functions on grids of the form $\mathcal{G}_p + \mathbf{o}$.

▶ **Lemma 5.** *A function $f : \mathbb{N}^2 \to \mathbb{N}$ is semilinear if and only if f is semiaffine.*

Our main result, Theorem 1, shows that output-oblivious functions are exactly the following two special types of semiaffine functions. In the first special case, on each grid $\mathcal{G}_p + \mathbf{o}$, f is restricted to be an affine (rather than a more general semiaffine) function.

▶ **Definition 6** (Grid-affine function). A function $f : \mathbb{N}^2 \to \mathbb{N}$ is *grid-affine* if and only if for some $p \in \mathbb{N}^+$, f is a combination of affine functions on points and on grids of period p.

A function $f : D \to \mathbb{N}$ is *increasing* if $f(\mathbf{n}) \leq f(\mathbf{n}')$ for all $\mathbf{n} \leq \mathbf{n}'$, where $\mathbf{n}, \mathbf{n}' \in D$. Doty and Hajiaghayi [11] observed that an output-oblivious function must be increasing. Accordingly, we hereafter focus on increasing functions.

▶ **Definition 7** (Fissure function). Let $\mathcal{G}$ be a two-way-infinite grid. An increasing semiaffine function $f : \mathcal{G} \to \mathbb{N}$ is a *partial fissure function* if for some $\mathbf{o} \in \mathbb{N}^2$, f can be represented as follows for all $\mathbf{n} \geq \mathbf{o}$:

$$f(\mathbf{n}) = \begin{cases} \varphi_A(\mathbf{n}), & \text{if } \varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) \leq -k, \\ \varphi_{-i}(\mathbf{n}) = \varphi_A(\mathbf{n}) - d_{-i}, & \text{if } \varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = -i, 1 \leq i < k, \\ \varphi_i(\mathbf{n}) = \varphi_B(\mathbf{n}) - d_i, & \text{if } \varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = i, 0 \leq i < k, \\ \varphi_B(\mathbf{n}), & \text{if } \varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) \geq k. \end{cases} \tag{1}$$

where $\varphi_A(\mathbf{n}) = A_0 + A_1 n_1 + A_2 n_2$, $\varphi_B(\mathbf{n}) = B_0 + B_1 n_1 + B_2 n_2$, for integers A_0 and B_0, nonnegative rationals A_1, A_2, B_1 and B_2, and nonnegative integers $d_{-k}, \ldots, d_{-1}, d_0, d_1, \ldots, d_k$. For $-k \leq i \leq k$, we refer to the line $\varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = i$ as a *fissure line* and call it L_i. Moreover, $\varphi_A < \varphi_B$ on Dom_A and $\varphi_B < \varphi_A$ on Dom_B; thus $A_1 > B_1$ and $B_2 > A_2$. We say $f : \mathbb{N}^2 \to \mathbb{N}$ is a *(complete) fissure function* if f is a combination of partial fissure functions on grids of period p.

3 Proof of Sufficiency in Theorem 1

This section shows that if an increasing semilinear function $f : \mathbb{N}^2 \to \mathbb{N}$ is either a grid-affine function or a fissure function, then f is output-oblivious. We do this in three lemmas. Lemma 8 shows that an increasing affine function whose domain is a grid is output-oblivious. Lemma 10 shows that a partial fissure function is output-oblivious. Finally, Lemma 11 shows that if f is increasing and is a combination of partial output-oblivious functions defined on grids, we can stitch together the CRNs for the partial functions to obtain an output-oblivious CRN for f.

▶ **Lemma 8.** *Let $\mathcal{G}$ be a grid. Any increasing affine function $f : \mathcal{G} \to \mathbb{N}$ is output-oblivious.*

Proof. We consider the case that $\mathcal{G} = \{(p, 0)\alpha_1 + (0, q)\alpha_2 : \alpha_i \in \mathbb{N}\} + \mathbf{o}$ is two-way-infinite; the cases when $\mathcal{G}$ is a point or a line are simpler. Let $f(\mathbf{n}) = a_1 n_1 + a_2 n_2 + a_0$, where $a_1, a_2 \in \mathbb{Q}^+$ and $a_0 \in \mathbb{Q}$. Since $\mathcal{G}$ is two-way-infinite and f is increasing, a_1 and a_2 are nonnegative. On input $\mathbf{n} = (n_1, n_2) \in \mathcal{G}$, i.e., given n_1 copies of X_1 and n_2 copies of X_2, the following CRN will produce $f(\mathbf{n})$ copies of Y:

$$\begin{array}{lll} L + o_1X_1 + o_2X_2 & \to L' + (a_1o_1 + a_2o_2 + a_0)Y & // \text{ base case} \\ L' + pX_1 & \to L' + a_1pY & \\ L' + qX_2 & \to L' + a_2qY & \end{array}$$

Note that the first reaction must produce a non-negative and integral number of Y's since $f(\mathbf{o}) \in \mathbb{N}$. Likewise, $a_1p \in \mathbb{N}$ since $a_1p = f(\mathbf{o} + (p, 0)) - f(\mathbf{o})$, and similarly for a_2q. Finally, the CRN is clearly output-oblivious since the output species Y is never a reactant. ◀

We show in Lemma 10 below that any partial fissure function is output-oblivious. First we describe some useful structure pertaining to partial fissure functions $f : \mathcal{G} \to \mathbb{N}$. We can represent such a fissure function as $f(\mathbf{n}) = \min\{\varphi_A(\mathbf{n}), \varphi_B(\mathbf{n})\} - d_i$, where d_i is determined by the fissure line L_i on which $\mathbf{n}$ resides, and $d_i = 0$ if i is not on a fissure line; this formulation is not identical to but is equivalent to that of Definition 7. As noted in that definition, it must be that $A_1 > B_1$ and $B_2 > A_2$, since $\varphi_A < \varphi_B$ on Dom_A and vice versa.

For all integers i, let L_i be the line $\varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = i$. All of these lines, which include the $2k - 1$ "fissure lines" L_i, $-k < i < k$, have the same slope. In addition to the fissure lines, our CRN construction will also refer to the lines L_i for i in the range $[k, \dots, K - 1]$, where $K = k + d_{\max} - 1$. We call these the *lower boundary lines*, and we call the lines L_i for i in the range $[-K + 1, \dots, -k]$ the *upper boundary lines*. Note that $(0, 0)$ is on the line $L_{A_0 - B_0}$ and more generally, if point $\mathbf{p}$ is on line L_i then $(A_1 - B_1)p_1 - (B_2 - A_2)p_2 = i - A_0 + B_0$. For $\mathbf{n} \in \mathcal{G}$ let $M(\mathbf{n}) = (\varphi_A(\mathbf{n}), \varphi_B(\mathbf{n}))$. The next lemma shows that $M(\mathbf{n}) \in \mathbb{N}^2$ for all sufficiently large $\mathbf{n} \in \mathcal{G}$, even though Dom_A and Dom_B are proper subsets of $\mathcal{G}$.

▶ **Lemma 9.** *Let $\varphi : D \to \mathbb{N}$ be a partial affine function, where D is a wedge domain on $\mathcal{G}$. Let $\mathbf{m}$ be a minimal point of D. Then $\varphi(\mathbf{n}) \in \mathbb{N}$ on all $\mathbf{n} \in \mathcal{G}$ with $\mathbf{n} \geq \mathbf{m}$.*

We let $\mathcal{P}$ be the set of rational points $\mathbf{p}$ for which $M(\mathbf{p}) \in \mathbb{N}$ and let $\mathcal{Q}$ be the range of M with respect to domain $\mathcal{P}$. For $\mathbf{q} \in \mathcal{Q}$, let $M^{-1}(\mathbf{q})$ denote the inverse of M ($M^{-1}\mathbf{q}$ is unique since (A_1, A_2) and (B_1, B_2) are linearly independent). The following claim follows easily from the definition of M and will be useful later.

▶ **Claim 1.** *Let $z_1, z_1', z_2, z_2' \in \mathbb{N}$. If $z_1 \leq z_1'$ and $M^{-1}(z_1, z_2)$ is in Dom_B then $M^{-1}(z_1', z_2)$ is also in Dom_B. Similarly if $z_2 \leq z_2'$ and $M^{-1}(z_1, z_2)$ is in Dom_A then $M^{-1}(z_1, z_2')$ is also in Dom_A.*

▶ **Lemma 10.** *Any partial fissure function $f : \mathcal{G} \to \mathbb{N}$ is output-oblivious.*

Proof. For simplicity we assume that the grid $\mathcal{G}$ is $\mathbb{N}^2$, i.e., the period of the grid is 1 and the offset $\mathbf{o}$ is zero; it is straightforward to generalize to larger grid periods. With these assumptions, it must be that A_0 and B_0 are nonnegative integers, which slightly simplifies base cases of our construction.

The CRN input is represented as the initial counts of species X_1 and X_2, and $\mathbf{x} = (x_1, x_2)$ denotes the counts of X_1 and X_2 that have been consumed at any time. Rather than producing output $f(\mathbf{x})$ directly upon consumption of $\mathbf{x}$, our CRN produces $\varphi_A(\mathbf{x})$ copies of a species Z_1 and $\varphi_B(\mathbf{x})$ copies of a species Z_2, effectively computing the mapping M described above. Note that $\varphi_A(\mathbf{x})$ and $\varphi_B(\mathbf{x})$ are nonnegative integers by Lemma 9. The CRN works backwards from the quantities $\varphi_A(\mathbf{x})$ and $\varphi_B(\mathbf{x})$ to reconstruct $f(\mathbf{x})$. Roughly this is possible because $f(\mathbf{x})$ is "almost" the min of $\varphi_A(\mathbf{x})$ and $\varphi_B(\mathbf{x})$, and min is easy to compute. More precisely, we can assume that $f(\mathbf{x}) = \min\{\varphi_A(\mathbf{x}), \varphi_B(\mathbf{x})\} - d_i$, where d_i is determined by the fissure line L_i on which $\mathbf{n}$ resides, and $d_i = 0$ if i is not on a fissure line. In addition to the input, a leader L is also present initially. Other CRN molecules (not initially present) represent a state $[l_{\mathbf{x}}, l_{\mathbf{z}}, d]$ containing three components; we explain the components later. Our CRN has three types of reactions: *Z-producing*, *Z-consuming*, and *Y-producing* reactions. The first Z-producing reaction handles the base case, producing $(\varphi_A(0,0), \varphi_B(0,0)) = (A_0, B_0)$:

$$L \to L' + A_0Z_1 + B_0Z_2.$$

The remaining two Z-producing reactions consume X_1 and X_2 while producing Z_1 and Z_2. If L_i is the line containing $\mathbf{x}$, the first state component, $l_{\mathbf{x}}$, keeps track of $i \mod 2K-1$, where $K = k + d_{\max}$. If i is in the range $[-K+1, K-1]$ then $l_{\mathbf{x}}$ uniquely determines i. For convenience in what follows, we consider $l_{\mathbf{x}}$ to be in the range $[-K+1, K-1]$ rather than $[0, 2K-1]$. The reactions are as follows, where $*$ represents any state component value that is unchanged as a result of the reaction:

$$\begin{aligned} [l_{\mathbf{x}}, *, *] + X_1 &\to [l_{\mathbf{x}} + (A_1 - B_1) \mod 2K-1, *, *] + A_1Z_1 + B_1Z_2 \\ [l_{\mathbf{x}}, *, *] + X_2 &\to [l_{\mathbf{x}} + (A_2 - B_2) \mod 2K-1, *, *] + A_2Z_1 + B_2Z_2 \end{aligned}$$

We next describe the Z-consuming reactions. These reactions update the remaining two components of the state to keep track of which fissure or boundary line contains $M^{-1}(\mathbf{z})$, where $\mathbf{z} = (z_1, z_2)$ denotes the counts of (Z_1, Z_2) that have been consumed at any time. The reactions also track what is the *deficit*, i.e., the difference between the "true" output $f(M^{-1}(\mathbf{z}))$ and the current output y, i.e., number of copies of species Y that has been actually produced so far. Formally, all reactions maintain the following *state invariant*: if after any reaction the state is $[l_{\mathbf{x}}, l_{\mathbf{z}}, d]$ then

1. $l_{\mathbf{z}}$ is the index of the boundary or fissure line $L_{l_{\mathbf{z}}}$ that contains $M^{-1}(\mathbf{z})$, and $l_{\mathbf{z}}$ is in the range $-K+1 \le l_{\mathbf{z}} \le K-1$; and
2. $d = f(M^{-1}(\mathbf{z})) - y$ is the deficit in the number of y's produced, and is in the finite range $-d_{\max} \le d \le 2d_{\max} + 1$, where $d_{\max} = \max\{d_i \mid -k < i < k\}$.

Z-consuming reactions of the first type handle the base case when $\mathbf{n} = (0,0)$:

$$L' + A_0Z_1 + B_0Z_2 \to [A_0 - B_0, A_0 - B_0, \min\{A_0, B_0\} - d_{A_0 - B_0}].$$

Z-consuming reactions of the second type consume a copy of Z_1 and reactions of the third type consume a copy of Z_2. Upon consumption, the state components are updated to ensure that the state invariant holds.

$$\begin{aligned} [l_{\mathbf{x}}, l_{\mathbf{z}}, d] + Z_1 &\to [l_{\mathbf{x}}, l_{\mathbf{z}} + 1, d + d^+_{l_{\mathbf{z}}}], \quad -K < l_{\mathbf{z}} < k, d \le d_{\max} \text{ and} \\ [l_{\mathbf{x}}, l_{\mathbf{z}}, d] + Z_2 &\to [l_{\mathbf{x}}, l_{\mathbf{z}} - 1, d + d^-_{l_{\mathbf{z}}}], \quad -k < l_{\mathbf{z}} < K, d \le d_{\max}, \end{aligned}$$

where

$$d^+_{l_{\mathbf{z}}} = \begin{cases} d_{l_{\mathbf{z}}} - d_{l_{\mathbf{z}}+1}, & l_{\mathbf{z}} \ge 0 \\ d_{l_{\mathbf{z}}} - d_{l_{\mathbf{z}}+1} + 1, & l_{\mathbf{z}} < 0. \end{cases} \quad \text{and } d^-_{l_{\mathbf{z}}} = \begin{cases} d_{l_{\mathbf{z}}} - d_{l_{\mathbf{z}}-1} + 1, & l_{\mathbf{z}} \ge 0 \\ d_{l_{\mathbf{z}}} - d_{l_{\mathbf{z}}-1}, & l_{\mathbf{z}} < 0. \end{cases}$$

The deficit d can never exceed $2d_{\max}+1$ since the reactions are only applicable when $d \le d_{\max}$ and d can increase by at most $d_{\max} + 1$.

The Y-producing reactions produce output molecules of species Y, while maintaining the state invariant above, and ensuring that at the end of the computation the number of Ys produced equals $f(\mathbf{n})$. The first Y-producing reaction produces $d - d_{\max}$ copies of Y when d becomes greater than $d_{\max}$.

$$[*, *, d] \rightarrow [*, *, d_{\max}] + (d - d_{\max})Y, \text{ if } d > d_{\max}.$$

Before describing the remaining Y-producing reactions, we describe some properties of the system of reactions above. We say that Z-consumption *stalls* if none of the Z-consuming reactions are ever applicable again. Let $\mathbf{z}_s = (z_{1s}, z_{2s})$ be the counts of (Z_1, Z_2) consumed when Z-consumption stalls ($\mathbf{z}_s$ is independent of the order in which the reactions happen). The Y-producing reaction above ensures that the Z-consuming reactions are never stalled because d becomes too large. Also, the Z-consuming reactions don't stall if $l_{\mathbf{z}}$ is a fissure line and another Z_1 is or will eventually be available (and similarly if another Z_2 is or will eventually be available), because $ł_z$ changes by 1 upon consumption of Z_1 and so is still less than K.

Stalling happens when and only when one of the following (exclusive) cases arise. (i) All copies of both Z_1 and Z_2 have been consumed and no more will ever be produced, so $\mathbf{z}_s = (\varphi_A(\mathbf{n}), \varphi_B(\mathbf{n}))$. (ii) All copies of Z_2 have been consumed and no more will ever be produced, so $z_{2s} = \varphi_B(\mathbf{n})$ but $z_{1s} < \varphi_A(\mathbf{n})$. In this case, $M^{-1}(\mathbf{z}_s)$ is on a lower boundary line. To see why, note that if $M^{-1}(\mathbf{z}_s)$ were on a fissure or upper boundary line, then the Z-consuming reaction that consumes Z_1 would eventually be applicable, because $l_{\mathbf{z}}$ is in the proper range and at least one copy of Z_1 has yet to be consumed. (iii) All copies of Z_1 have been consumed and no more will ever be produced, so $z_{1s} = \varphi_A(\mathbf{n})$, but $z_{2s} < \varphi_B(\mathbf{n})$. In this case, the line $L_{l_{\mathbf{z}}}$ containing $M^{-1}(\mathbf{z}_s)$ must be an upper boundary line.

▶ **Claim 2.** $f(M^{-1}(\mathbf{z}_s)) = f(\mathbf{n})$.

Proof. This is trivial in case (i) when all Zs have been consumed and no more will be produced, since $M^{-1}(\mathbf{z}_s) = \mathbf{n}$. Consider case (ii) (case (iii) is similar). Then $M^{-1}(\mathbf{z}_s) = M^{-1}(z_{1s}, \varphi_B(n))$, $z_{1s} < \varphi_A(\mathbf{n})$, and the line containing $M^{-1}(\mathbf{z}_s)$ is a lower boundary line. By Claim 1, $\mathbf{n}$ must be in Dom_B, because $\mathbf{n} = M^{-1}(\varphi_A(\mathbf{n}), \varphi_B(\mathbf{n}))$ and $\varphi_A(\mathbf{n}) > z_{1s}$. Therefore, $f(M^{-1}(\mathbf{z}_s)) = \varphi_B(M^{-1}(\mathbf{z}_s)) = \varphi_B(\mathbf{n}) = f(\mathbf{n})$. ◀

We now return to the last three reactions of the CRN, which are Y-producing reactions; we will number them so that we can reference them later and refer to them as *deficit-clearing* reactions. The next reaction clears a positive deficit when both $\mathbf{x}$ and $M^{-1}(\mathbf{z})$ lie on the same fissure line:

$$[l_{\mathbf{x}}, l_{\mathbf{z}}, d] \rightarrow [l_{\mathbf{x}}, l_{\mathbf{z}}, 0] + dY \text{ if } -k < l_{\mathbf{x}} = l_{\mathbf{z}} < k. \tag{2}$$

The last two Y-producing reactions clear the deficit if $M^{-1}(\mathbf{z})$ is on a lower boundary line and for some nonnegative integer r, $M^{-1}(\mathbf{z} + (r, 0))$ is on a line L_l with $l = l_{\mathbf{x}} \mod 2K - 1$. If such an r exists, let r_1 be the smallest such integer and add the following reaction:

$$[l_{\mathbf{x}}, l_{\mathbf{z}}, d] + r_1 Z_1 \rightarrow [l_{\mathbf{x}}, l_{\mathbf{z}}, 0] + r_1 Z_1 + dY, \quad l_{\mathbf{z}} \geq k, l_{\mathbf{z}} + r_1 = l_{\mathbf{x}} \mod 2K - 1, d > 0. \tag{3}$$

We add a similar reaction when the line $L_{l_{\mathbf{z}}}$ containing $M^{-1}(\mathbf{z})$ is an upper boundary line, when an similarly-defined r_2 exists:

$$[l_{\mathbf{x}}, l_{\mathbf{z}}, d] + r_2 Z_2 \rightarrow [l_{\mathbf{x}}, l_{\mathbf{z}}, 0] + r_2 Z_2 + dY, \quad l_{\mathbf{z}} \leq -k, l_{\mathbf{z}} + r_2 = l_{\mathbf{x}} \mod 2K - 1, d > 0. \tag{4}$$

This completes the description of the CRN. We need one more claim in order to complete the proof of the lemma:

▶ **Claim 3.** *When Z-consumption stalls, the deficit is nonnegative.*

To complete the proof, we argue that once Z-consumption stalls, some deficit-clearing reaction will eventually be applicable, ensuring that the output eventually produced is $f(\mathbf{n})$. If $M^{-1}(\mathbf{z}_s)$ is on a fissure line then $M^{-1}(\mathbf{z}_s)$ must equal $\mathbf{n}$, in which case Y-producing reaction (2) is applicable. If $M^{-1}(\mathbf{z}_s)$ is on a boundary line then either (3) or (4) will be applicable once all inputs are consumed, since for some r, either $M^{-1}(\mathbf{z}_s + (r, 0)) = \mathbf{n}$ or $M^{-1}(\mathbf{z}_s + (0, r)) = \mathbf{n}$. Thus in all cases some Y-producing reaction eventually clears the deficit, ensuring that the output produced is $f(\mathbf{n})$. ◀

▶ **Lemma 11.** *(Stitching Lemma) Let Let f be an increasing function. If $f : \mathbb{N}^2 \to \mathbb{N}$ is a finite combination of output-oblivious functions whose domains are grids, then f is output-oblivious. Also if f is the min of a finite number of output-oblivious functions then f is output-oblivious.*

Proof Sketch. Let f be a finite combination of output-oblivious functions, say $f_1, f_2, \ldots, f_m$, whose domains are grids. We first describe the construction for the case that the domain Dom_j of f_j is a two-way-infinite grid for all $j, 1 \leq j \leq m$. Let the offset of the jth grid be $\mathbf{o}_j = (o_{j,1}, o_{j,2})$. On input $\mathbf{n}$, our CRN $\mathcal{C}$ first produces m distinct "inputs" $\mathbf{n}^{(\mathbf{j})} \in \mathbb{N}^2$ such that $\mathbf{n} \leq \mathbf{n}^{(\mathbf{j})}$ and $\mathbf{n} = \mathbf{n}^{(\mathbf{j})}$ if $\mathbf{n} \in \text{Dom}_j$. From these, $\mathcal{C}$ produces m "outputs" $y_j = f_j(\mathbf{n}^{(\mathbf{j})})$, using CRNs $\mathcal{C}_j$ for each f_j. Finally, $\mathcal{C}$ produces $y = \min\{y_1, \ldots, y_m\}$.

To see that such a $\mathcal{C}$ is correct, i.e., that $y = f(\mathbf{n})$, note that if $\mathbf{n} \in \text{Dom}_j$ then $y_j = f_j(\mathbf{n}^{(\mathbf{j})}) = f_j(\mathbf{n}) = f(\mathbf{n})$, since $\mathbf{n} = \mathbf{n}^{(\mathbf{j})}$, and if $\mathbf{n} \notin \text{Dom}_j$ then $y_j = f_j(\mathbf{n}^{(\mathbf{j})}) \geq f(\mathbf{n})$, since f is increasing and $\mathbf{n}^{(\mathbf{j})} \geq \mathbf{n}$. Thus $f(\mathbf{n}) = \min\{y_1, \ldots, y_m\} = y$. The details of producing the $\mathbf{n}^{(\mathbf{j})}$s and the output are found in the full version.

When f is the min of a finite number of output-oblivious functions, say $f_1, f_2, \ldots, f_m$, we can similarly stably compute each f_i using an output-oblivious CRN $\mathcal{C}_i$ such that the species for each $\mathcal{C}_i$ are distinct, and then take the min of the outputs as the result. ◀

We complete this section by proving the sufficiency (if) direction of Theorem 1.

Theorem 1 (if direction). *A semilinear function $f : \mathbb{N}^2 \to \mathbb{N}$ is output-oblivious if f is increasing and is either grid-affine or the minimum of finitely many fissure functions.*

Proof. First suppose that f is grid-affine. Then by Definition 6, f is a combination of affine functions $f_1, \ldots, f_m$ whose domains $\mathcal{G}_1, \ldots, \mathcal{G}_m$ respectively are grids. By Lemma 8, $f_i : \mathcal{G}_i \to \mathbb{N}$ is output-oblivious, $1 \leq i \leq m$. By Lemma 11, f is output-oblivious.

Otherwise f is the min of finitely many complete fissure functions $f_1, \ldots, f_m$. By Definition 7, each f_i is a combination of partial fissure functions on grids of period p, for some $p \in \mathbb{N}$. By Lemma 10, each of these partial fissure functions is output-oblivious. By Lemma 11, each f_i is output-oblivious and also f is output-oblivious. ◀

4 Proof of Necessity in Theorem 1

In this section we prove that if a function is output-monotonic, then it is either grid-affine or the minimum of finitely many fissure functions. In Section 4.1 we describe two conditions on a function which ensure that it is not output-oblivious. In Section 4.2 we show two technical results which are needed in Section 4.3, which contains the proof of necessity. The arguments made in this section pertain to output-monotonic functions, allowing us to both characterize this set of functions and output-oblivious functions since any output-oblivious function is clearly output-monotonic.

4.1 Impossibility Lemmas

The results of this section use Dickson's Lemma:

▶ **Lemma 12.** *(Dickson's Lemma [10]) Any infinite sequence in $\mathbb{N}^k$ has an infinite, non-decreasing subsequence.*

▶ **Lemma 13.** *Let $f : \mathbb{N}^2 \to \mathbb{N}$ be a semiaffine function. Suppose that $f = \varphi_i$ on Dom_i and $f = \varphi_j$ on Dom_j, where Dom_i and Dom_j lie on the same grid, Dom_i is a wedge domain and for some two-way-infinite line L in Dom_j, $\varphi_j > \varphi_i$ on L. Then f cannot be stably computed by an output-monotonic CRN.*

Proof. Suppose to the contrary that CRN $\mathcal{C}$ stably computes f. Either Dom_i is counter-clockwise to Dom_j or vice versa. Assume it is the latter; the proof is similar if the orientation of the domains is reversed.

Let $\{\mathbf{p}_k\}_{k\in\mathbb{N}}$, be an infinite sequence of points in Dom_i that are strictly increasing in the first dimension, and form a line which is not parallel to L (this is possible since Dom_i is a wedge domain). Let $\mathbf{c}_k$ be a stable configuration reached on a computation of $\mathcal{C}$ on input $\mathbf{p}_k$. Applying Dickson's Lemma, choose an infinite subsequence of $\{\mathbf{c}_k\}$ and renumber so that $\mathbf{c}_k \leq \mathbf{c}_{k+1}$ for all k. Let $\mathbf{n}_1, \mathbf{n}_2, \dots$ be another strictly increasing sequence in $\mathbb{N}^2$ such that $\mathbf{p}_k + \mathbf{n}_k \in L$ and $\mathbf{p}_{k'} + \mathbf{n}_k \in \mathrm{Dom}_i$ for $k' > k$. Such a sequence exists because $\{\mathbf{p}_k\}$ is not parallel to L, and L is two-way-infinite.

If $\mathcal{C}$ is correct, then on input $\mathbf{p}_k + \mathbf{n}_k$ some execution sequence of $\mathcal{C}$ first reaches configuration $\mathbf{c}_k$, which has $\varphi_i(\mathbf{p}_k)$ copies of Y, and then outputs $\varphi_j(\mathbf{p}_k + \mathbf{n}_k) - \varphi_i(\mathbf{p}_k)$ additional Ys (since $\mathbf{p}_k + \mathbf{n}_k \in \mathrm{Dom}_j$). Let $k' > k$. On input $\mathbf{p}_{k'} + \mathbf{n}_k$ (which is in Dom_i), $\mathcal{C}$ can output a number of Ys equal to:

$$\varphi_i(\mathbf{p}_{k'}) + \varphi_j(\mathbf{p}_k + \mathbf{n}_k) - \varphi_i(\mathbf{p}_k).$$

This occurs when $\mathcal{C}$ first produces stable output $\varphi_i(\mathbf{p}_{k'})$ while consuming input $\mathbf{p}_{k'}$ and reaching configuration $\mathbf{c}_{k'}$. Since $\mathbf{c}_{k'} \geq \mathbf{c}_k$, it can then follow the same execution that it would follow from $\mathbf{c}_k$ to produce $\varphi_j(\mathbf{p}_k + \mathbf{n}_k) - \varphi_i(\mathbf{p}_k)$ copies of Y (since all the necessary species are available). However, since $\varphi_j > \varphi_i$ on L, the number of Ys produced is greater than

$$\varphi_i(\mathbf{p}_{k'}) + \varphi_i(\mathbf{p}_k + \mathbf{n}_k) - \varphi_i(\mathbf{p}_k) = \varphi_i(\mathbf{p}_{k'} + \mathbf{n}_k)$$

(the equality here follows since φ_i is affine). Thus too many Y's can be output by C on input $\mathbf{p}_{k'} + \mathbf{n}_k$, and so C cannot be output-monotonic. ◀

▶ **Lemma 14.** *Let $f : \mathbb{N}^2 \to \mathbb{N}$ be a semiaffine function. Suppose that $f = \varphi_i$ on Dom_i and $f = \varphi_j$ on Dom_j, where Dom_i and Dom_j are wedge domains on the same grid $\mathcal{G}$ such that (i) $\varphi_i = \varphi_j$ on $\mathbb{N}^2$ and (ii) there is a two-way-infinite line $L \subset \mathcal{G}$ separating Dom_i and Dom_j, with $\varphi_L < \varphi_i$ $(= \varphi_j)$ on L. Then f cannot be stably computed by an output-monotonic CRN.*

4.2 Properties of Increasing Semiaffine Functions

Here we show several useful properties of increasing semiaffine functions $f : \mathbb{N}^2 \to \mathbb{N}$. For a partial affine function φ_i with domain Dom_i, write $\varphi_i(\mathbf{n}) = \langle \mathbf{a}_i, \mathbf{n} \rangle + a_{0,i}$ (where $\langle \cdot, \cdot \rangle$ denotes the standard inner product). We say that a line L is a *constant distance* from Dom_i if there exists some constant K such that for all $\mathbf{n} \in L$ there is some $\mathbf{c}(\mathbf{n})$ with $-(K, K) \leq \mathbf{c}(\mathbf{n}) \leq (K, K)$ and $\mathbf{n} + \mathbf{c}(\mathbf{n}) \in \mathrm{Dom}_i$.

▶ **Lemma 15.** *Let $f : \mathbb{N}^2 \to \mathbb{N}$ be an increasing semiaffine function. Suppose that $f = \varphi_i$ on Dom_i and $f = \varphi_j$ on Dom_j, where φ_i and φ_j are partial affine functions, Dom_i and Dom_j are two-way infinite domains, and some line L in Dom_j is a constant distance from Dom_i. Then there exists some $\kappa \in \mathbb{Q}$ such that $\langle \mathbf{a}_i, \mathbf{n} \rangle = \langle \mathbf{a}_j, \mathbf{n} \rangle + \kappa$ for all $\mathbf{n} \in L$.*

▶ **Lemma 16.** *Let $f : \mathbb{N}^2 \to \mathbb{N}$ be an increasing semiaffine function. Suppose that $f = \varphi_i$ on Dom_i and $f = \varphi_j$ on Dom_j, where φ_i and φ_j are partial affine functions. If there exist two non-parallel lines $I, L \subset \mathrm{Dom}_i$ which are both a constant distance from Dom_j, then $\mathbf{a}_i = \mathbf{a}_j$.*

What follows is an easy consequence of the previous lemma.

▶ **Lemma 17.** *Let $f : \mathbb{N}^2 \to \mathbb{N}$ be an increasing semiaffine function. If there exists a two-way-infinite domain D on which φ_i and φ_j are well-defined such that $f(\mathbf{n}) = \varphi_i(\mathbf{n}) = \varphi_j(\mathbf{n})$ on all $\mathbf{n} \in D$, then $\varphi_i = \varphi_j$.*

4.3 Proof of Necessity

We first provide the main argument in the proof of necessity – that the desired result holds on individual grids.

▶ **Lemma 18.** *Let $f : \mathbb{N}^2 \to \mathbb{N}$ be an increasing semiaffine function with period p. Then f is output-monotonic only if for any large enough offset $\mathbf{o}$, $f|_{\mathcal{G}_p+\mathbf{o}}$ is either affine or the minimum of finitely many partial fissure functions.*

Proof. Fix a representation of f as a semiaffine function with period p. Choose $\mathbf{o}$ large enough so that, if $\mathcal{G} = \mathcal{G}_p + \mathbf{o}$, no domains of f that are points or one-way-infinite domains overlap $\mathcal{G}$, and also no two-way-infinite domains of f cross. Assume that $f|_{\mathcal{G}}$ is not affine; we need to show that $f|_{\mathcal{G}}$ is the minimum of partial fissure functions. Assume that the representation of f on $\mathcal{G}$ minimizes the number of wedge domains.

Consider all line domains in the representation of $f|_{\mathcal{G}}$, plus all two-way-infinite lines that define the top or bottom boundaries of wedge domains. Partition these lines into maximal sets of parallel lines. For each such set s, we define a function $f_s : \mathcal{G} \to \mathbb{N}^2$, and show in Claim 4 that each f_s is a partial fissure function. Let $\mathcal{S}$ be the set of all of the sets s of parallel lines. We show in Claim 5 that $f|_{\mathcal{G}} = \min_{s\in\mathcal{S}} f_s$, completing the proof of Lemma 18.

Definition of f_s. Fix any maximal set s of parallel lines. Without loss of generality we assume that s has at least three lines (we can add additional domains to f's representation if needed to ensure this). Some line of s, say $L_{A,s}$, defines the lower boundary of a wedge domain; we assume that this is top line of s (we can remove line domains of s above $L_{A,s}$ from f's representation if needed to ensure this). Let $f = \varphi_{A,s}$ on this wedge domain, where $\varphi_{A,s}$ is a partial affine function. In what follows, we drop the subscript s when referring to these and other domains and functions. Let Dom_A be the wedge of points of $\mathcal{G}$ that lie on or above L_A. (Note that Dom_A may not be a domain of $f|_{\mathcal{G}}$ since lines from some other set s_j may lie above the lines of s.) Similarly, we can assume that the bottom line, say L_B, of s defines the upper boundary of a wedge domain. Let $f = \varphi_B (= \varphi_{B,s})$ on this wedge domain and let Dom_B be the wedge of points of $\mathcal{G}$ that lie on or below L_B. We can assume without loss of generality (by adding more lines if necessary and further adjusting which lines are L_A and L_B) that any line L that lies between, and is parallel to, L_A and L_B is a (possibly empty) domain of $f|_G$. Number these lines $L_1, L_2, \ldots$, say from top to bottom, and let φ_i be the partial affine function that agrees with f on line L_i.

Let $f_s : \mathcal{G} \to \mathbb{N}$ be the following function associated with set s:

$$f_s(\mathbf{n}) = \begin{cases} \varphi_A(\mathbf{n}), & \text{if } \mathbf{n} \in \mathrm{Dom}_A, \\ \varphi_i(\mathbf{n}), & \text{if } \mathbf{n} \in L_i, 2 \le i \le |s| - 1, \\ \varphi_B(\mathbf{n}), & \text{if } \mathbf{n} \in \mathrm{Dom}_B\,. \end{cases}$$

▶ **Claim 4.** *f_s is a partial fissure function.*

Proof. Note that the lines L_i are parallel to, and have constant distance from, both L_A and L_B. By Lemma 15, for some $d_{i,A} \in \mathbb{Q}$ we have that $\varphi_i(\mathbf{n}) = \varphi_A(\mathbf{n}) - d_{i,A}$ for all $\mathbf{n}$ on line L_i. That is, L_i is the set of all $\mathbf{n}$ such that $\varphi_A(\mathbf{n}) - \varphi_i(\mathbf{n}) = d_{i,A}$. Since $\varphi_i(\mathbf{n})$ and $\varphi_A(\mathbf{n}) \in \mathbb{N}$ for all $\mathbf{n}$ in $\mathcal{G} \cap L_i$, $d_{i,A}$ must be in $\mathbb{Z}$. Also, since f is output-monotonic, Lemma 13 shows that $\varphi_i \le \varphi_A$ on L_i and so $d_{i,A}$ must be in $\mathbb{N}$. By reasoning similar to that in the last paragraph, $\varphi_i(\mathbf{n}) = \varphi_B(\mathbf{n}) - d_{i,B}$ for all $\mathbf{n}$ on line L_i, for some $d_{i,B} \in \mathbb{N}$. It follows that L_i is the set of points $\mathbf{n} \in \mathcal{G}$ for which $\varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = k$, for some $k \in \mathbb{Z}$. It follows (by potentially adding yet more lines if necessary so that the number of lines L_i that lie above the line $\varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = 0$ is equal to the number of lines of s that lie below the line $\varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = 0$) that we can represent $f_s(\mathbf{n})$ as in Definition 7.

It remains to show that $\varphi_A < \varphi_B$ on Dom_A and $\varphi_B < \varphi_A$ on Dom_B. Suppose that there exists a point $\mathbf{n} \in \mathrm{Dom}_A$ such that $\varphi_A(\mathbf{n}) \ge \varphi_B(\mathbf{n})$. If there are only finitely many such points in any dimension, then we may disregard them by taking $\mathbf{o}$ sufficiently large. Hence, we may assume that if there is one such point then there are infinitely many and they form a two-way-infinite domain, D. Let $E = \{\mathbf{n} \in D : \varphi_A(\mathbf{n}) > \varphi_B(\mathbf{n})\}$ and $F = \{\mathbf{n} \in D : \varphi_A(\mathbf{n}) = \varphi_B(\mathbf{n})\}$. Note that $E \cup F = D$. We consider three cases. If E and F are both one-way-infinite, then one is a horizontal line and the other a vertical line. As above, we may disregard these points by taking $\mathbf{o}$ large enough. If E is two-way-infinite we can find a two-way-infinite line $L \subset \mathrm{Dom}_A$ such that $\varphi_A > \varphi_B$ on L. By Lemma 13, this contradicts the fact that f is output-monotonic. Otherwise, F is two-way-infinite. By Lemma 16 we see that $\varphi_A = \varphi_B$. Furthermore, for some i, we have $\varphi_i < \varphi_A$. Otherwise, $d_{i,A} = 0$ for all i and consequently the number of wedge domains of $f|_{\mathcal{G}}$ can be reduced by merging the domains Dom_A, Dom_B and the L_i contradicting our assumption that the representation of f on $\mathcal{G}$ minimizes the number of wedge domains. However, φ_A, φ_B and φ_i then meet the conditions of Lemma 14, a contradiction. The proof that $\varphi_B < \varphi_A$ on Dom_B is similar. ◀

▶ **Claim 5.** $f|_{\mathcal{G}} = \min_{s \in \mathcal{S}} f_s$.

Proof. Let $\mathbf{n} \in \mathrm{Dom}(f_s)$, so that $f|_{\mathcal{G}}(\mathbf{n}) = f_s(\mathbf{n})$. Suppose to the contrary that $f_{s'}(\mathbf{n}) < f_s(\mathbf{n})$ for some $s' \in \mathcal{S}$. We consider the case where the lines of s' lie above those of s; the case where the lines of s' lie below those of s is similar. Hence $f_{s'}(\mathbf{n}) = \varphi_{A,s'}(\mathbf{n})$. We consider three distinct cases based on the partitioning of $\mathrm{Dom}(f_s)$.

1. $\mathbf{n} \in \mathrm{Dom}_{A,s}$ so $f_s(\mathbf{n}) = \varphi_{A,s}(\mathbf{n})$. Applying the same argument as in the proof of Claim 4, since $\varphi_{A,s'}(\mathbf{n}) < \varphi_{A,s}(\mathbf{n})$, there is a line $L \subset \mathrm{Dom}_{A,s}$ such that $\varphi_{A,s'} < \varphi_{A,s}$ on L. Thus, $\varphi_{A,s'}$ and $\varphi_{A,s}$ and their respective domains meet the condition of Lemma 13, contradicting the fact that f can be stably computed by an output-monotonic CRN.
2. $\mathbf{n} \in \mathrm{Dom}_{i,s}$ for some i with $2 \le i \le |s| - 1$, so $f_s(\mathbf{n}) = \varphi_{i,s}(\mathbf{n})$. Since f_s is a fissure function, we have $\varphi_{i,s}(\mathbf{n}) = \varphi_{A,s}(\mathbf{n}) - d_i$, for some $d_i \in \mathbb{N}$, hence $f_s(\mathbf{n}) = \varphi_{i,s}(\mathbf{n}) \le \varphi_{A,s}(\mathbf{n}) \le \varphi_{A,s'}(\mathbf{n}) = f_{s'}(\mathbf{n})$. Clearly then, it cannot be the case that $f_{s'}(\mathbf{n}) < f_s(\mathbf{n})$.
3. $\mathbf{n} \in \mathrm{Dom}_{B,s}$. This is similar to Case 1. ◀

Since we get a contradiction in all three cases, we conclude that $f_s(\mathbf{n})$ must be less than or equal to $f_{s'}(\mathbf{n})$ for all $s' \in \mathcal{S}$, and the claim is proved.

This completes the proof of Lemma 18. ◀

Lemma 18 describes properties of increasing semiaffine functions on one grid. We now turn to properties of such functions across grids. The proof of the next lemma builds on that of Lemma 18 and uses Lemma 16 to relate f across grids.

▶ **Lemma 19.** *Let $f : \mathbb{N}^2 \to \mathbb{N}$ be an output-monotonic semiaffine function with period p. Let $\mathbf{o}$ be large enough that for all $\mathbf{o}' \geq \mathbf{o}$, $f|_{\mathcal{G}_p+\mathbf{o}'}$ is either affine or the minimum of finitely many partial fissure functions. If f is affine on $\mathcal{G} = \mathcal{G}_p + \mathbf{o}$, then $f|_{\mathcal{G}_p+\mathbf{o}'}$ must also be affine for all $\mathbf{o}' \geq \mathbf{o}$, and thus f is grid-affine. Otherwise f is the minimum of finitely many fissure functions.*

We complete this section by proving the necessity (only if) direction of Theorem 1, strengthening it slightly so that it applies also to output-monotonic functions.

Theorem 1 (only if direction). *A semilinear function $f : \mathbb{N}^2 \to \mathbb{N}$ is output-monotonic only if f is increasing and is either grid-affine or the minimum of finitely many fissure functions.*

Proof. By Lemma 5, f is a semiaffine function with period p. By Lemmas 18 and 19, f is either grid-affine or the minimum of fissure functions. ◀

5 Conclusions and Future Work

Here we have characterized the class of functions $f : \mathbb{N}^2 \to \mathbb{N}$ that can be stably computed by output-oblivious and output-monotonic stochastic chemical reaction networks (CRNs) with a leader. A natural next step for future work is to generalize the result to functions $f : \mathbb{N}^k \to \mathbb{N}$ for $k > 2$; we are optimistic that many of the building blocks that we introduce here for the two-dimensional case will generalize to the multi-dimensional case.

Another natural question is to determine what can be computed when there is no leader. By similar reasoning to that of Chalk et al. [7] for mass-action CRNs, such functions must be super-additive, but whether only the super-additive semilinear functions have output-oblivious stochastic CRNs remains to be determined.

Yet another direction for future work is to determine whether, for some functions, there is a provable gap between the time needed to stably compute the functions with an output-oblivious CRN and the time needed by a CRN that is not restricted to be output-oblivious. Yet other directions are to better understand output-oblivious CRN computation when errors are allowed, and whether it is possible to "repair" a CRN that is not output-oblivious so that composition is possible.

References

1 Dan Alistarh, Rati Gelashvili, and Milan Vojnović. Fast and exact majority in population protocols. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, pages 47–56. ACM, 2015.

2 Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. In *Dolev S. (eds) Distributed Computing (DISC), Lecture Notes in Computer Science*, volume 4167, pages 61–75. Springer, Berlin, Heidelberg, 2006.

3 Dana Angluin, James Aspnes, and David Eisenstat. Stably computable predicates are semilinear. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 292–299, New York, NY, USA, 2006. ACM Press. doi:10.1145/1146381.1146425.

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

5 Adam Arkin, John Ross, and Harley H McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells. *Genetics*, 149(4):1633–1648, 1998.

6 Luca Cardelli and Attila Csikász-Nagy. The cell cycle switch computes approximate majority. *Scientific reports*, 2:656, 2012.

7 Cameron Chalk, Niels Kornerup, Wyatt Reeves, and David Soloveichik. Composable Rate-Independent Computation in Continuous Chemical Reaction Networks. In Milan Ceska and David Safránek, editors, *Computational Methods in Systems Biology*, pages 256–273, Cham, 2018. Springer International Publishing.

8 Ho-Lin Chen, David Doty, and David Soloveichik. Deterministic function computation with chemical reaction networks. *Natural Computing*, 13(4):517–534, December 2014.

9 Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui, and Eric Ruppert. Secretive birds: Privacy in population protocols. In *International Conference On Principles Of Distributed Systems*, pages 329–342. Springer, 2007.

10 Leonard Eugene Dickson. Finiteness of the Odd Perfect and Primitive Abundant Numbers with n Distinct Prime Factors. *American Journal of Mathematics*, 35(4):413–422, 1913. URL: http://www.jstor.org/stable/2370405.

11 David Doty and Monir Hajiaghayi. Leaderless deterministic chemical reaction networks. *Natural Computing*, 14(2):213–223, 2015.

12 Allen Hjelmfelt, Edward D Weinberger, and John Ross. Chemical implementation of neural networks and Turing machines. *Proceedings of the National Academy of Sciences*, 88(24):10983–10987, 1991.

13 Hua Jiang, Marc D Riedel, and Keshab K Parhi. Digital signal processing with molecular reactions. *IEEE Design and Test of Computers*, 29(3):21–31, 2012.

14 Nils E Napp and Ryan P Adams. Message passing inference with chemical reaction networks. In *Advances in neural information processing systems*, pages 2247–2255, 2013.

15 Lulu Qian and Erik Winfree. Scaling up digital circuit computation with DNA strand displacement cascades. *Science*, 332(6034):1196–1201, 2011.

16 David Soloveichik, Matthew Cook, Erik Winfree, and Jehoshua Bruck. Computation with finite stochastic chemical reaction networks. *Natural Computing*, 7, 2008.

17 David Soloveichik, Georg Seelig, and Erik Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences*, 107(12):5393–5398, 2010.

18 Chris Thachuk and Anne Condon. Space and energy efficient computation with DNA strand displacement systems. In *International Workshop on DNA-Based Computers*, pages 135–149. Springer, 2012.

19 David Yu Zhang, Rizal F Hariadi, Harry MT Choi, and Erik Winfree. Integrating DNA strand-displacement circuitry with DNA tile self-assembly. *Nature communications*, 4:1965, 2013.