Composable Computation in Leaderless, Discrete Chemical Reaction Networks

Hooman Hashemi
The University of British Columbia, Vancouver, Canada

Ben Chugg
Stanford University, CA, USA
benchugg@stanford.edu

Anne Condon
The University of British Columbia, Vancouver, Canada
condon@cs.ubc.ca

Abstract

We classify the functions $f : \mathbb{N}^d \to \mathbb{N}$ that are stably computable by leaderless, output-oblivious discrete (stochastic) Chemical Reaction Networks (CRNs). CRNs that compute such functions are systems of reactions over species that include d designated input species, whose initial counts represent an input $\mathbf{x} \in \mathbb{N}^d$, and one output species whose eventual count represents $f(\mathbf{x})$. Chen et al. showed that the class of functions computable by CRNs is precisely the semilinear functions. In output-oblivious CRNs, the output species is never a reactant. Output-oblivious CRNs are easily composable since a downstream CRN can consume the output of an upstream CRN without affecting its correctness. Severson et al. showed that output-oblivious CRNs compute exactly the subclass of semilinear functions that are eventually the minimum of quilt-affine functions, i.e., affine functions with different intercepts in each of finitely many congruence classes. They call such functions the output-oblivious functions. A leaderless CRN can compute only superadditive functions, and so a leaderless output-oblivious CRN can compute only superadditive, output-oblivious functions. In this work we show that a function $f : \mathbb{N}^d \to \mathbb{N}$ is stably computable by a leaderless, output-oblivious CRN if and only if it is superadditive and output-oblivious.

2012 ACM Subject Classification Theory of computation → Models of computation; 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.DNA.2020.3

Funding *Hooman Hashemi*: Supported by an NSERC Discovery Grant.
Ben Chugg: Supported by an NSERC Undergraduate Research Award.
Anne Condon: Supported by an NSERC Discovery Grant.

Acknowledgements This work benefited greatly from conversations with Eric Severson and David Doty. Thanks also to David Haley and Eric Severson for help in generating the figures.

1 Introduction

Chemical Reaction Networks (CRNs) have proven to be very valuable as a programming language for describing how computations can ensue when molecules react. There is now a rich complexity theory of computation with the CRN model, as well as the closely related population protocol model of distributed computing [2, 4, 7, 10, 11, 17]. This theory helps us understand what types of computational or engineered dynamic processes are possible with molecules, since CRNs can be "compiled" down to DNA strand displacement systems, which in turn can be implemented with real DNA strands in a test tube [5, 15, 18, 19].

26th International Conference on DNA Computing and Molecular Programming (DNA 26).
Editors: Cody Geary and Matthew J. Patitz; Article No. 3; pp. 3:1–3:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

It is natural to ask: If CRNs C and C' compute functions f and f', respectively, can we compose the CRNs to compute the composition $f' \circ f$? In this paper we study this question for leaderless, discrete CRNs, resolving an open question of Chugg et al. [9], Severson et al. [16], and Chalk et al. [6]. Here we first describe the CRN model, background and motivation for the work, and then describe our result in more detail.

We focus on discrete CRNs (also called stochastic CRNs), which are described as a finite set of chemical reactions among abstract species. Discrete CRNs *stably* compute functions $f : \mathbb{N}^d \to \mathbb{N}$ in the following sense. An input $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{N}^d$ is represented by initial counts of d designated molecular species. A single copy of a so-called leader molecule may also be present initially. Reactions of the CRN ensue, changing the species counts over time. Eventually, regardless of the order of reactions, the count of a designated output species Y equals $f(\mathbf{x})$ and does not subsequently change. See Figure 1. Here and throughout, we assume without loss of generality that the range of f is $\mathbb{N}$, since functions that map $\mathbb{N}^d$ to $\mathbb{N}^l$ for some $l > 1$ can be computed by first cloning l distinct copies of the inputs, and then for each $1 \leq i \leq l$, computing the ith output from the ith copy of the inputs.

(a)		(b)		(c)	
$X_1' + X_1'$	$\to 2Y'$			X_1	$\to X_1' + X_1''$
$X_1' + X_2'$	$\to 2Y'$	X_2''	$\to 2Y''$	X_2	$\to X_2' + X_2''$
$X_2' + X_2'$	$\to 2Y'$			$Y' + Y''$	$\to Y$

Figure 1 Examples of Chemical Reaction Networks (CRNs) for stable function computation. (a) A CRN C_1 for $f(x_1, x_2) = x_1 + x_2 + ((x_1 + x_2) \mod 2)$, with inputs X_1', X_2' and output Y'. (b) A CRN C_2 for $f'(x_1, x_2) = 2x_2$, with inputs X_1'', X_2'' and output Y''. (The input X_1'' does not appear in the reaction.) (c) A CRN C for the function $\min\{f(x_1, x_2), 2x_2\}$. C converts its inputs X_1, X_2 to those needed by CRNs C_1 and C_2 of parts (a) and (b), and then computes the function $\min\{f(x_1, x_2), 2x_2\}$ from the outputs of C_1 and C_2, demonstrating function composition. All three CRNs are leaderless.

Exactly the semilinear predicates and functions are stably computable by discrete CRNs [2, 7]. Such functions are linear on each of a finite number of semilinear domains – subsets of $\mathbb{N}^d$ that are defined using $\geq$ or mod. See Figure 2.

Let C and C' be discrete CRNs that stably compute functions $f : \mathbb{N}^d \to \mathbb{N}$ and $f' : \mathbb{N} \to \mathbb{N}$. Suppose furthermore that C is *output-oblivious*: That is, the output species of C is not a reactant of any reaction of C. This condition ensures that outputs produced by C can be consumed as inputs by a downstream CRN, without affecting the correctness of C. Then if the output species of C is the input species of C', and there is no other species common to C and C', the CRN $C \cup C'$ computes $f' \circ f$.

More generally, suppose that CRNs $C_1, C_2, \ldots, C_{d'}$ stably compute the functions f_1, f_2, $\ldots$, $f_{d'} : \mathbb{N}^d \to \mathbb{N}$, and CRN C' stably computes $f' : \mathbb{N}^{d'} \to \mathbb{N}$. Suppose also that the C_i are output-oblivious, the output of C_i is the ith input to C' and there is no other species common to the CRNs. Then $C_1 \cup C_2 \ldots C_{d'} \cup C'$ computes $f'(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_{d'}(\mathbf{x}))$. For example, combining the reactions of the CRN of Figure 1 parts (a), (b) and (c) results in a CRN to compute the function $f'(x_1, x_2) = \min\{f(x_1, x_2), 2x_2\}$.

If a function f is stably computable by an output-oblivious CRN with a leader, we say that f is *obliviously-computable*. Obliviously-computable functions must be nondecreasing, because a CRN on input $\mathbf{x} + \mathbf{x}'$ can produce $f(\mathbf{x})$ Y's (by ignoring inputs representing $\mathbf{x}'$), and if Y's are never consumed, the stable output $f(\mathbf{x} + \mathbf{x}')$ that is eventually produced must then be at least $f(\mathbf{x})$. However, not all nondecreasing semilinear functions are obliviously-computable, the max function being an interesting counterexample. Chugg et al. [9] characterized the

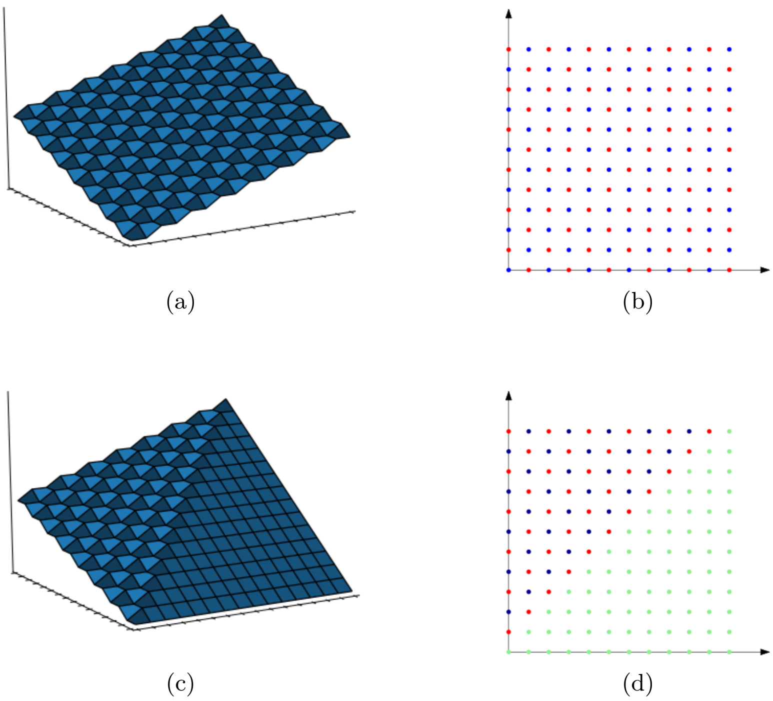


Figure 2 Illustrations of quilt-affine functions with domain $\mathbb{N}^2$. (a) The function $h(\mathbf{x}) = x_1 + x_2 - ((x_1 + x_2) \mod 2)$. (b) Domains of the function h of part (a). $h(\mathbf{x}) = x_1 + x_2$ on the domain $\text{Dom}_1 = \{\mathbf{x} \in \mathbb{N}^2 \mid x_1 + x_2 = 0 \pmod 2\}$, shown in blue. Dom_1 is linear since it equals $\{\alpha_1(2,0) + \alpha_2(0,2) + \alpha_3(1,1) + (0,0) \mid \alpha_1, \alpha_2, \alpha_3 \in \mathbb{N}\}$. Also, $h(\mathbf{x}) = x_1 + x_2 - 1$ on the domain $\text{Dom}_2 = \{\mathbf{x} \in \mathbb{N}^2 \mid x_1 + x_2 = 1 \pmod 2\}$, shown in red. The domain Dom_2 is the union of two linear sets, namely $\{\alpha_1(2,0) + \alpha_2(0,2) + \alpha_3(1,1) + (0,1) \mid \alpha_1, \alpha_2, \alpha_3 \in \mathbb{N}\}$ and $\{\alpha_1(2,0) + \alpha_2(0,2) + \alpha_3(1,1) + (1,0) \mid \alpha_1, \alpha_2, \alpha_3 \in \mathbb{N}\}$. (c) The function $f(\mathbf{x}) = \min\{h(x_1, x_2), 2x_2\}$. (d) Domains of the function f of part (c). $f(\mathbf{x}) = 2x_2$ on the domain $\text{Dom}_3 = \{\mathbf{x} \in \mathbb{N}^2 \mid x_2 + 1 \leq x_1\}$, shown in green. Dom_3 is linear since it equals $\{\alpha_1(1,0) + \alpha_2(1,1) + (1,0) \mid \alpha_1, \alpha_2 \in \mathbb{N}\}$. Also, $f(x_1, x_2) = h(x_1, x_2)$ on the semilinear domains $\text{Dom}'_1 = \text{Dom}_1 \cap \{\mathbf{x} \in \mathbb{N}^2 \mid x_1 \leq x_2\}$ and $\text{Dom}'_2 = \text{Dom}_2 \cap \{\mathbf{x} \in \mathbb{N}^2 \mid x_1 \leq x_2\}$, shown in red and blue.

subclass of obliviously-computable functions with two inputs, i.e., functions $f : \mathbb{N}^2 \to \mathbb{N}$. Severson et al. [16] gave a general characterization of obliviously-computable functions $f : \mathbb{N}^d \to \mathbb{N}$, for any d; such functions are eventually the min of *quilt-affine* functions, defined as nondecreasing linear functions with a periodic intercept, see Figure 2. See Section 2 for formal definitions of quilt-affine and obliviously-computable functions.

The results of Chugg et al. and Severson et al. described so far concern discrete, output-oblivious CRNs with leaders. What about leaderless CRNs? Output-oblivious functions computed by a leaderless CRN C must be superadditive, i.e., $f(\mathbf{x}) + f(\mathbf{x}') \geq f(\mathbf{x} + \mathbf{x}')$. This is because on input $\mathbf{x} + \mathbf{x}'$, reactions of a leaderless CRN could be used to independently compute both $f(\mathbf{x})$ and $f(\mathbf{x}')$, resulting in $f(\mathbf{x}) + f(\mathbf{x}')$ output molecules, so this quantity must be less than or equal to the eventual stable output, namely $f(\mathbf{x} + \mathbf{x}')$. This raises the question: Is the class of functions $f : \mathbb{N}^d \to \mathbb{N}$ that can be stably computed by leaderless output-oblivious CRNs exactly the superadditive obliviously-computable functions? Severson et al. showed that this is indeed the case when $d = 1$, but the more general case was left as an open problem. In this paper we show that the answer is "yes" for all d:

▶ **Theorem 1.** *Functions that are stably computable by leaderless output-oblivious CRNs are exactly the superadditive obliviously-computable functions.*

Our proof of Theorem 1 has two parts. First, building on the previous work of Severson et al. and Chugg et al., we provide in Claim 5 a new characterization of superadditive, obliviously-computable functions as the minimum of superadditive quilt-affine functions on *well-ordered* domains, which we define in Section 2. Then in Claim 14 we construct a leaderless, output-oblivious CRN for superadditive, obliviously-computable functions, using the well-ordered domain representation.

Our result has strong parallels with that of Chalk et al. [6] who studied composability of function-computing CRNs for the *continuous* (also called mass-action) CRN model. In this model, real-valued species concentrations, rather than discrete species counts, evolve over time, according to a finite set of reactions. Earlier, Chen et al. [8] showed that continuous CRNs can stably (i.e., regardless of actual reaction rates) compute positive-continuous, piecewise rational linear functions. Chalk et al. showed that such functions are obliviously-computable by continuous CRNs if and only if they are superadditive. However, the proof techniques for the discrete and continuous CRN models are quite different.

2 The CRN Model and Obliviously-Computable Functions

Following a summary of useful notation, we describe Chemical Reaction Networks (CRNs), stable CRN function computation, and output-oblivious function computation. We then describe the result of Severson et al. [16] that characterizes the class of functions that are stably computable by output-oblivious CRNs with a leader, i.e., obliviously-computable functions, in terms of quilt-affine functions. Finally, we provide a new, alternative characterization of obliviously-computable functions that is useful for our main results.

2.1 Notation

We use $\mathbb{N}$ to denote the set of nonnegative integers, $\mathbb{N}_+$ the positive integers, $\mathbb{Z}$ the integers, $\mathbb{Q}$ the rationals, and $\mathbb{Q}_{\geq 0}$ the nonnegative rationals. Where d is understood, we use boldface to represent d-dimensional vectors $\mathbf{x} \in \mathbb{N}^d$, and x_i to denote the ith component of $\mathbf{x}$, $1 \leq i \leq d$. We write $\mathbf{x} \leq \mathbf{x}'$ to denote that $x_i \leq x'_i$, for all $i, 1 \leq i \leq d$, and $\mathbf{x} < \mathbf{x}'$ to denote that $\mathbf{x} \leq \mathbf{x}'$ and for some $i, 1 \leq i \leq d$, $x_i < x'_i$. For $1 \leq i \leq d$, we let $\mathbf{e}_i$ denote the d-dimensional unit vector $(e_{i1}, \ldots, e_{id})$ in which all components are zero except that $e_{ii} = 1$. We denote the d-dimensional vector of all zero's by $\mathbf{0}$.

For $d, p \in \mathbb{N}_+$, $\mathbb{Z}^d/p\mathbb{Z}^d$ denotes the additive group of $\mathbb{Z}^d$ modulo p. Each element of $\mathbb{Z}^d/p\mathbb{Z}^d$ is a congruence class of the form $\{\mathbf{n} + p\mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^d\}$ for some $\mathbf{n} \in \mathbb{N}^d$, and we denote this set by $\overline{\mathbf{n}}$.

2.2 Chemical Reaction Networks and Stable Function Computation

A discrete Chemical Reaction Network (CRNs) is specified as a finite set $\mathcal{Z} = \{Z_1, \ldots, Z_m\}$ of *species*, plus a finite set of $\mathcal{R}$ of *reactions* $(\mathbf{s}, \mathbf{t}) = ((s_1, \ldots, s_m), (t_1, \ldots, t_m)) \in \mathbb{N}^{\mathcal{Z}} \times \mathbb{N}^{\mathcal{Z}}$ of the form

$$\sum_{k: s_k > 0} s_k Z_k \rightarrow \sum_{k: t_k > 0} t_k Z_k,$$

where for at least one j, $s_j \neq t_j$. The species Z_k with $s_k > 0$ are the *reactants*, which are *consumed*, while those with $t_k > 0$ are the *products*. (A species may be both a reactant and product of the same reaction). A *configuration* $\mathbf{c} \in \mathbb{N}^m$ describes counts of species in $\mathcal{Z}$, and

$\mathbf{c}(Z)$ denotes the count of species $Z \in \mathcal{Z}$. Reaction $(\mathbf{s}, \mathbf{t})$ is *applicable* to configuration $\mathbf{c}$ if $\mathbf{s} \leq \mathbf{c}$, i.e., sufficiently many copies of each reactant are present. Application of the reaction to $\mathbf{c}$ results in the configuration $\mathbf{c}' = \mathbf{c} - \mathbf{s} + \mathbf{t}$, and we write $\mathbf{c} \to \mathbf{c}'$. If $\mathbf{c}_0 \to \mathbf{c}_1 \to \dots \mathbf{c}_t$ then we say that $\mathbf{c}_t$ is *reachable* from $\mathbf{c}_0$ and call $\mathbf{c}_0 \to \mathbf{c}_1 \to \dots \mathbf{c}_t$ an *execution* of the CRN.

A CRN $\mathcal{C}$ to stably compute a function $f : \mathbb{N}^d \to \mathbb{N}$ has designated input species, say $X_1, \dots, X_d$, a designated output species, say Y, and may or may not have a designated leader species, $L \in \mathcal{Z} \setminus \mathcal{I}$. Leaderless function computation on input $\mathbf{x} \in \mathbb{N}^d$ starts from a valid initial configuration $\mathbf{c}_0 = \mathbf{c}_0(\mathbf{x})$, where $\mathbf{c}_0(X_i) = x_i$ for $1 \leq i \leq d$, and the count of any other species is 0. CRN computation with a leader differs only in that the initial count of the leader species L is 1, i.e., $\mathbf{c}_0(L) = 1$. We say that $\mathcal{C}$ *stably computes* f if for every valid initial configuration $\mathbf{c}_0 = \mathbf{c}_0(\mathbf{x})$ for some $\mathbf{x}$, and for every configuration $\mathbf{c}$ reachable from $\mathbf{c}_0$, there exists a stable configuration $\mathbf{c}'$ reachable from $\mathbf{c}$ such that $f(\mathbf{x}) = \mathbf{c}'(Y)$. Here, $\mathbf{c}'$ is *stable* if for every $\mathbf{c}'' \in \mathbb{N}^m$ reachable from $\mathbf{c}'$, $\mathbf{c}'(Y) = \mathbf{c}''(Y)$. That is, once configuration $\mathbf{c}'$ is reached, the count of the output species does not change. Stable computation with a leader is defined in the same way, except that in the initial configuration the count of a designated leader species L is 1.

Chen et al. [7] (building on related work of Angluin et al. [2, 4] on predicate computation by population protocols) showed that exactly the *semilinear* functions are stably computable by CRNs. A semilinear function is the union of partial affine functions on linear domains. A domain $E \subset \mathbb{N}^d$ is *linear* if $E = \{\sum_{\mathbf{z} \in F} \alpha_{\mathbf{z}} \mathbf{z} + \mathbf{o} : \alpha_{\mathbf{z}} \in \mathbb{N}\}$ for some finite set $F \subset \mathbb{N}^d$ and $\mathbf{o} \in \mathbb{N}^d$. Thus, if $E_1, E_2, \dots, E_m$ are linear sets, $\cup_{i=1}^m E_i = \mathbb{N}^d$, and for $1 \leq i \leq m$ $f_i : E_i \to \mathbb{N}$ is a partial affine function, then the function $f : \mathbb{N}^d \to \mathbb{N}$ where $f(\mathbf{x}) = f_i(\mathbf{x})$ if $\mathbf{x} \in E_i$ is semilinear. Figure 2 shows examples of linear sets and semilinear functions, illustrating show the union of linear sets can be defined using $\geq$ or mod. Doty and Hajiaghayi [11] showed that leaderless CRNs also stably compute the semilinear functions.

2.3 Obliviously-Computable Functions As Quilt-Affine Functions

A CRN $\mathcal{C}$ is *output-oblivious* if no reaction consumes the output species. A function f is *obliviously-computable* if some output-oblivious CRN with a leader stably computes f. A subclass of the obliviously-computable functions are the *leaderless obliviously-computable* functions, that can be stably computed by leaderless output-oblivious CRNs.

Severson et al. [16] defined a *quilt-affine function* $h : \mathbb{N}^d \to \mathbb{Z}$ to be a nondecreasing function that is the sum of a rational linear function and a periodic function. That is, for some $\nabla_h \in \mathbb{Q}^d_{\geq 0}$, called the gradient of h, some $p \in \mathbb{N}^+$, called the period, and some $B : \mathbb{Z}^d / p\mathbb{Z}^d \to \mathbb{Q}$, called the periodic intercept,

$$h(\mathbf{x}) = \nabla_h \cdot \mathbf{x} + B(\overline{\mathbf{x}}).$$

For example, the 2D function $h(\mathbf{x}) = x_1 + x_2 - ((x_1 + x_2) \mod 2)$ of Figure 2 is quilt-affine, since it can be written as $h(\mathbf{x}) = (1, 1) \cdot (x_1, x_2) + B(\overline{\mathbf{x}})$, where $B(0, 0) = B(1, 1) = 0$ and $B(0, 1) = B(1, 0) = -1$. Severson et al. [16] proved the following result.

▶ **Theorem 2.** (Severson et al. [16]) *A function $f : \mathbb{N}^d \to \mathbb{N}$ is obliviously-computable if and only if it satisfies the following three properties:*

(i) *f is* nondecreasing*, i.e., $f(\mathbf{x}) \leq f(\mathbf{x}')$ for all $\mathbf{x} \leq \mathbf{x}'$.*

(ii) *There exist (nondecreasing) quilt-affine functions $h_1, \dots, h_m : \mathbb{N}^d \to \mathbb{N}$ and $\mathbf{k}_f \in \mathbb{N}^d$ such that for all $\mathbf{x} \geq \mathbf{k}_f$, $f(\mathbf{x}) = \min_i\{h_i(\mathbf{x})\}$.*

(iii) *All fixed-input restrictions of f are obliviously-computable. Here, a fixed-input restriction of f is a function on $d-1$ inputs defined as*

$$f_{[x_i \to j]}(\mathbf{x}) = f(x_1, x_2, \ldots, x_{i-1}, j, x_{i+1}, \ldots, x_d),$$

for some $1 \le i \le d$ and $j \in \mathbb{N}$.

2.4 Obliviously-Computable Functions As Well-Ordered Quilt-Affine Functions

Here we adapt Severson et al.'s result to obtain a slightly different characterization of obliviously-computable functions, as the union of *partial* quilt-affine functions over *well-ordered* domains sets. This result lays the foundation for the rest of the paper. Claim 4 in Section 3 demonstrates that these partial quilt-affine functions may be assumed to be superadditive which, coupled with Theorem 2, implicitly proves one direction of Theorem 1. Additionally, the well-ordered domain sets will be further refined by the CRN construction in Section 4, enabling a quilt-affine function to be expressed simply as a piecewise affine function. A *partial quilt-affine function* is simply a quilt-affine function that is defined only over a subset of $\mathbb{N}^d$.

Next we define well-ordered domain sets. Let $\mathbf{w} \in \mathbb{N}^d$ be fixed and let $\mathbf{0} \le \mathbf{o} \le \mathbf{w}$. Let

$$\mathrm{Dom}_{\mathbf{o}}\ (= \mathrm{Dom}_{\mathbf{o},\mathbf{w}}) = \{\mathbf{x} \in \mathbb{N}^d \mid\ \mathbf{x} \ge \mathbf{o} \text{ and } x_i = o_i \text{ if } o_i < w_i\}. \quad (1)$$

The sets $\mathrm{Dom}_{\mathbf{o}}$ for $\mathbf{0} \le \mathbf{o} \le \mathbf{w}$, are disjoint and their union is $\mathbb{N}^d$. We call the set of sets $\{\mathrm{Dom}_{\mathbf{o}} \mid \mathbf{0} \le \mathbf{o} \le \mathbf{w}\}$ a *well-ordered domain set*, and we denote this set by $\mathcal{WO}_{\mathbf{w}}$. The sets are ordered in the sense that if $\mathbf{x} \in \mathrm{Dom}_{\mathbf{o}}$, $\mathbf{x}' \in \mathrm{Dom}_{\mathbf{o}'}$ and $\mathbf{x} \le \mathbf{x}'$ then $\mathbf{o} \le \mathbf{o}'$. Figure 3 (b) shows a well-ordered domain set for $\mathbb{N}^2$ where $\mathbf{w} = (4, 4)$. We will later use the following property of well-ordered domains:

▶ **Lemma 3.** *Let $\mathrm{Dom}_{\boldsymbol{o}}$ and $\mathrm{Dom}_{\boldsymbol{o}'}$ be domains of a well-ordered set defined by $\boldsymbol{w}$, and let $\mathrm{Dom}_{\boldsymbol{o}''}$ be the domain containing $\boldsymbol{o} + \boldsymbol{o}'$. Then for any $\mathbf{x} \in \mathrm{Dom}_{\boldsymbol{o}}$, and $\mathbf{x}' \in \mathrm{Dom}_{\boldsymbol{o}'}$, $\mathbf{x}'' = \mathbf{x} + \mathbf{x}' \in \mathrm{Dom}_{\boldsymbol{o}''}$.*

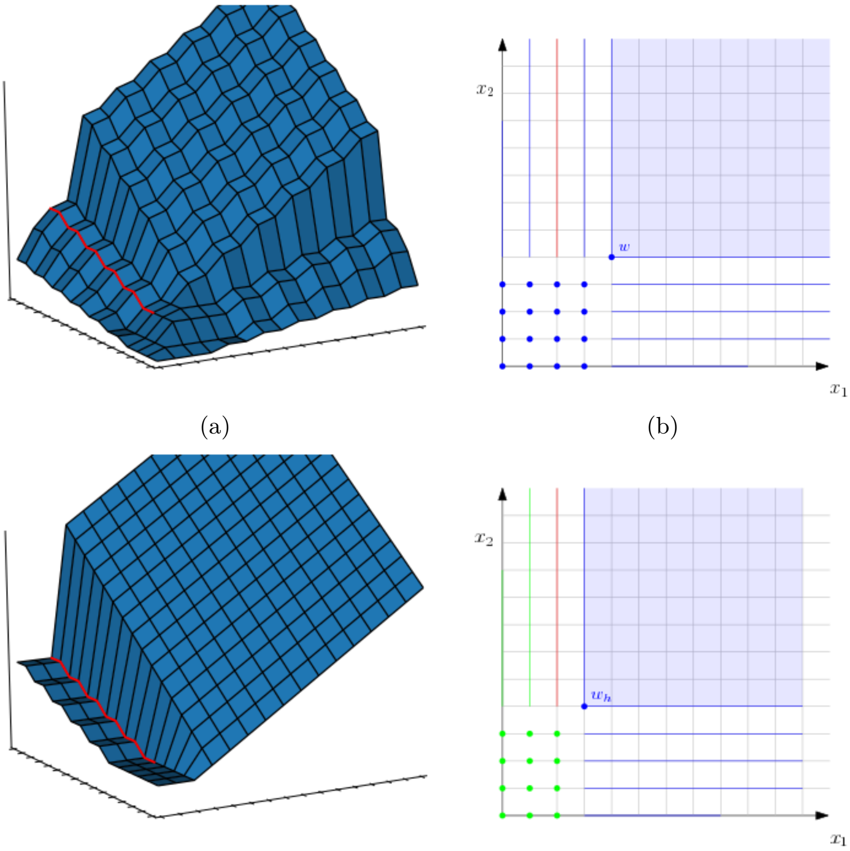
Proof. Since $\mathbf{x} \in \mathrm{Dom}_{\mathbf{o}}$ and $\mathbf{x}' \in \mathrm{Dom}_{\mathbf{o}'}$ we have that $\mathbf{x} \ge \mathbf{o}$ and $\mathbf{x}' \ge \mathbf{o}'$. Additionally, since $\mathbf{o} + \mathbf{o}' \in \mathrm{Dom}_{\mathbf{o}''}$, we have that $\mathbf{x} + \mathbf{x}' \ge \mathbf{o} + \mathbf{o}' \ge \mathbf{o}''$. So $\mathbf{x}''$ satisfies the first condition of membership in $\mathrm{Dom}_{\mathbf{o}''}$.

It remains to show that if $o''_i < w_i$, then $x_i = o''_i$. So suppose that $o''_i < w_i$. Then it must also be the case that $o_i < w_i$ and $o'_i < w_i$, that $o''_i = o_i + o'_i$, and that $x_i = o_i$ and $x'_i = o'_i$. The result follows. ◀

A *well-ordered quilt-affine function* is the finite union of partial quilt-affine functions, each of which is defined on a domain of a well-ordered domain set.

▷ Claim 4. Any obliviously-computable function is the minimum of a finite number of nondecreasing, well-ordered quilt-affine functions.

Proof. First, from the characterization of obliviously-computable functions of Severson et al. [16] given in Theorem 2 above, we identify a finite set of partial quilt-affine functions $\mathcal{H}$, as follows. We include in $\mathcal{H}$ the functions $h_1, h_2, \ldots, h_m$, each with domain $\mathrm{Dom}_{h_i} = \{\mathbf{x} \in \mathbb{N}^d \mid \mathbf{x} \ge \mathbf{k}_f\}$, described in property (ii) of Theorem 2.3. Then we recursively augment $\mathcal{H}$ by considering each of the fixed-input restrictions $f_{[x_i \to j]}$ of f of part (iii) of the definition, for each choice of $j < k_{f,i}$, and adding the functions corresponding to $f_{[x_i \to j]}$ from property (ii)



(a) (b) (c) (d)

Figure 3 (a) A well-ordered, superadditive function f with domain set $\mathcal{WO}_{\mathbf{w}}$ for $\mathbf{w} = (4, 4)$. Here, $f(x, y) = h(x, y) = y - (y \mod 2)$ on the red line and $f = 2y - (y \mod 2) + 2x - (x \mod 2) - 8$ on the large 2 dimensional area. The red line corresponds to the domain $\text{Dom}_{(3,4)}$. (b) The well-ordered domain set for the function f of part (a), with $\mathbf{w} = (4, 4)$. There is one 2D domain, eight 1D domains, and twelve 0D domains, i.e., points. (c) The function h_{WO} obtained from h via the construction of Claim 4. (d) The three domains Dom_h (in red), Dom_{big} (in blue) and $\text{Dom}_{\text{small}}$ (in green), for the function h of part (c). Here, $w_h = (3, 4)$.

of Theorem 2. There are d levels of recursion; the functions that are recursively added to $\mathcal{H}$ have at least one and up to d fixed inputs, and the remaining (non-fixed) inputs are lower bounded by some constant. Thus, for each function h added to $\mathcal{H}$, the domain of h has the form

$$\text{Dom}_h = \{\mathbf{x} \in \mathbb{N}^d \mid x_i = k_{h,i} \text{ if } i \in D_h \text{ and } x_i \geq k_{h,i} \text{ otherwise}\}, \tag{2}$$

for some $\mathbf{k}_h \in \mathbb{N}^d$ and $D_h \subseteq [1, \ldots, d]$. We can assume without loss of generality that all functions $h \in \mathcal{H}$ have the same period, since we can always take the least common multiple of the periods and redefine each h with respect to this least common multiple.

For each such $h \in \mathcal{H}$ we will construct a nondecreasing, well-ordered quilt-affine function $h_{\text{WO}} : \mathbb{N}^d \to \mathbb{N}$ such that $h_{\text{WO}}(\mathbf{x}) = h(\mathbf{x})$ for all $\mathbf{x} \in \text{Dom}_h$, and also $f(\mathbf{x}) \leq h_{\text{WO}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^d - \text{Dom}_h$. Then $f = \min_{\{h \in \mathcal{H}\}} h_{\text{WO}}$, and the claim follows.

We'll use the following notation when describing h_{WO}. Let $\nabla_h = (\lambda_{h,1}, \lambda_{h,2}, \ldots, \lambda_{h,k}) \in \mathbb{Q}^d$ be the gradient of h, let $\lambda_{\max} = \lceil \max_{h \in \mathcal{H}, 1 \leq i \leq d} \{\lambda_{h,i}\} \rceil$ and let

$$\nabla_{\max} = (\lambda_{\max}, \ldots, \lambda_{\max}).$$

Similarly, let B_h be the periodic intercept of h and let

$$B_{\max} = \left\lceil \max_{h \in \mathcal{H}, \mathbf{x} \in \mathbb{N}^d} \{B_h(\overline{\mathbf{x}} \mod p)\} \right\rceil.$$

We partition $\mathbb{N}^d$ into three domains:

- Dom_h, defined in Equation (2), where $\mathbf{k}_h \in \mathbb{N}^d$ and $D_h \subseteq [1, \ldots, d]$.
- $\mathrm{Dom}_{\text{small}} = \{\mathbf{x} \in \mathbb{N}^d \mid \ x_i \leq k_{h,i},\ 1 \leq i \leq d\} - \mathrm{Dom}_h$;
- $\mathrm{Dom}_{\text{big}} = \mathbb{N}^d - \mathrm{Dom}_{\text{small}} - \mathrm{Dom}_h$.

Also, for $\mathbf{x} \in \mathbb{N}^d$, we let

$$\mathrm{pr}(\mathbf{x}) = (\mathrm{pr}(x_1), \mathrm{pr}(x_2), \ldots, \mathrm{pr}(x_d)),$$

where $\mathrm{pr}(x_i) = k_i$ if $x_i \leq k_i$ and $\mathrm{pr}(x_i) = x_i$ otherwise. Note that for $\mathbf{x} \in \mathrm{Dom}_{\text{small}}$ we have $\mathrm{pr}(\mathbf{x}) \in \mathrm{Dom}_h$. We can now define h_{WO} as follows.

$$h_{\text{WO}}(\mathbf{x}) = \begin{cases} h(\mathbf{x}), & \text{for all } \mathbf{x} \in \mathrm{Dom}_h, \\ \nabla_{\max} \cdot \mathbf{x} + B_{\max}, & \text{for all } \mathbf{x} \in \mathrm{Dom}_{\text{big}}, \text{ and} \\ h(\mathrm{pr}(\mathbf{x})), & \text{for all } \mathbf{x} \in \mathrm{Dom}_{\text{small}}. \end{cases}$$

Figure 3 shows an example of the construction of h_{WO} from h.

First we show that $f(\mathbf{x}) \leq h_{\text{WO}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^d$. There are three cases, depending on whether $\mathbf{x}$ is in Dom_h, $\mathrm{Dom}_{\text{big}}$, or $\mathrm{Dom}_{\text{small}}$. (1) By definition, for $\mathbf{x} \in \mathrm{Dom}_h$ we have $f(\mathbf{x}) \leq h(\mathbf{x}) = h_{\text{WO}}(\mathbf{x})$. (2) For $\mathbf{x} \in \mathrm{Dom}_{\text{small}}$, we know that $\mathbf{x} \leq \mathrm{pr}(\mathbf{x})$ and so $f(\mathbf{x}) \leq f(\mathrm{pr}(\mathbf{x}))$. Also, $\mathrm{pr}(\mathbf{x}) \in \mathrm{Dom}_h$, and so we know from case (1) that $f(\mathrm{pr}(\mathbf{x})) \leq h_{\text{WO}}(\mathrm{pr}(\mathbf{x}))$. (3) For $\mathbf{x} \in \mathrm{Dom}_{\text{big}}$ we know that $f(\mathbf{x}) = h'(\mathbf{x})$ for some $h' \in \mathcal{H}$, and also by our choice of $\nabla_{\max}$ and $B_{\max}$ we have that $h'(\mathbf{x}) \leq \nabla_{\max} \cdot \mathbf{x} + B_{\max} = h_{\text{WO}}(\mathbf{x})$. Putting these together, we have that

$$f(\mathbf{x}) = h'(\mathbf{x}) \leq h_{\text{WO}}(\mathbf{x}).$$

Next we show that h_{WO} is non-decreasing, that is, $h_{\text{WO}}(\mathbf{x}) \leq h_{\text{WO}}(\mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}' \in \mathbb{N}^d$ with $\mathbf{x} \leq \mathbf{x}'$. We consider the possible cases for the domains of $\mathbf{x}$ and $\mathbf{x}'$:

1. $\mathbf{x} \in \mathrm{Dom}_h$ and $\mathbf{x}' \in \mathrm{Dom}_h$. Then $h_{\text{WO}}(\mathbf{x}) \leq h_{\text{WO}}(\mathbf{x}')$ since $h_{\text{WO}} = h$ on Dom_h and h is nondecreasing.
2. $\mathbf{x} \in \mathrm{Dom}_{\text{big}}$ and $\mathbf{x}' \in \mathrm{Dom}_{\text{big}}$. Then

$$h_{\text{WO}}(\mathbf{x}) = \nabla_{\max}(\mathbf{x}) + B_{\max} \leq \nabla_{\max}(\mathbf{x}') + B_{\max} = h_{\text{WO}}(\mathbf{x}').$$

3. $\mathbf{x} \in \mathrm{Dom}_h$ and $\mathbf{x}' \in \mathrm{Dom}_{\text{big}}$. Then

$$h_{\text{WO}}(\mathbf{x}) = \nabla_h(\mathbf{x}) + B(\overline{\mathbf{x}}) \leq \nabla_{\max}(\mathbf{x}) + B_{\max} \leq \nabla_{\max}(\mathbf{x}') + B_{\max} = h_{\text{WO}}(\mathbf{x}').$$

4. $\mathbf{x} \in \mathrm{Dom}_{\text{small}}$ and $\mathbf{x}' \in \mathrm{Dom}_{\text{small}}$. Then $\mathrm{pr}(\mathbf{x}) \leq \mathrm{pr}(\mathbf{x}')$ and both $\mathrm{pr}(\mathbf{x})$ and $\mathrm{pr}(\mathbf{x}')$ are in Dom_h, so

$$h_{\text{WO}}(\mathbf{x}) = h_{\text{WO}}(\mathrm{pr}(\mathbf{x})) \leq h_{\text{WO}}(\mathrm{pr}(\mathbf{x}')) = h_{\text{WO}}(\mathbf{x}'),$$

where the inequality holds because of case 1.

5. $\mathbf{x} \in \mathrm{Dom}_{\text{small}}$ and $\mathbf{x}' \in \mathrm{Dom}_h$. Then $\mathrm{pr}(\mathbf{x}) \in \mathrm{Dom}_h$ and $\mathrm{pr}(\mathbf{x}) \leq \mathbf{x}'$, so

$$h_{\text{WO}}(\mathbf{x}) = h_{\text{WO}}(\mathrm{pr}(\mathbf{x})) = h(\mathrm{pr}(\mathbf{x})) \leq h(\mathbf{x}') = h_{\text{WO}}(\mathbf{x}').$$

6. $\mathbf{x} \in \mathrm{Dom}_{\text{small}}$ and $\mathbf{x}' \in \mathrm{Dom}_{\text{big}}$. Then

$$h_{\text{WO}}(\mathbf{x}) = h_{\text{WO}}(\mathrm{pr}(\mathbf{x})) \leq \nabla_{\max}(\mathbf{x}) + B_{\max} \leq \nabla_{\max}(\mathbf{x}') + B_{\max} = h_{\text{WO}}(\mathbf{x}').$$

Finally, we show that h_{WO} is a well ordered quilt-affine function with offset $\mathbf{w}_h$, where we define $\mathbf{w}_h \in \mathbb{N}^d$ as $w_{h,i} = k_{h,i}$ if $i \in D_h$ and $w_{h,i} = k_{h,i} + 1$ otherwise. Consider any $\mathbf{o} \leq \mathbf{w}_h$. We need to show that h_{WO} is quilt-affine on the domain $\text{Dom}_{\mathbf{o}}$ (defined in Equation (1)). There are three cases:

1. If $\mathbf{o} = \mathbf{k}_h$ ($\leq \mathbf{w}_h$) then $\text{Dom}_{\mathbf{o}} = \text{Dom}_h$. By construction, $h_{\text{WO}} = h$ on Dom_h, and h is quilt-affine.
2. If $\mathbf{o} \leq \mathbf{k}_h$ but $\mathbf{o} \neq \mathbf{k}_h$, then $\mathbf{o}$ is in $\text{Dom}_{\text{small}}$. Let $\mathbf{o} = \mathbf{k}_h - \mathbf{k}'_h$, where $\mathbf{k}'_h \in \mathbb{N}^d$. For each $\mathbf{x} \in \text{Dom}_{\mathbf{o}}$ we have $\mathbf{x} \in \text{Dom}_{\text{small}}$, and so also $\text{pr}(\mathbf{x}) = \mathbf{x} + \mathbf{k}'_h \in \text{Dom}_h$. Therefore,

$$\begin{aligned} h_{\text{WO}}(\mathbf{x}) &= h_{\text{WO}}(\text{pr}(\mathbf{x})) \\ &= h(\text{pr}(\mathbf{x})) \\ &= h(\mathbf{x} + \mathbf{k}'_h) \\ &= \nabla_h \cdot (\mathbf{x} + \mathbf{k}'_h) + B(\overline{\mathbf{x} + \mathbf{k}'_h}) \\ &= \nabla_h \cdot \mathbf{x} + \nabla_h \cdot \mathbf{k}'_h + B(\overline{\mathbf{x} + \mathbf{k}'_h}) \\ &= \nabla_h \cdot \mathbf{x} + B'(\overline{\mathbf{x}}), \end{aligned}$$

   where $B'(\overline{\mathbf{x}}) = \nabla_h \cdot \mathbf{k}'_h + B(\overline{\mathbf{x} + \mathbf{k}'_h})$. Thus h_{WO} is quilt-affine.
3. If $\mathbf{o} \in \text{Dom}_{\text{big}}$, then since all $\mathbf{x} \geq \mathbf{o}$ are in Dom_{big}, the function h_{WO} on $\text{Dom}_{\mathbf{o}}$ is affine and therefore quilt-affine with period p. ◁

3 Superadditive, Obliviously-Computable Functions as Quilt-Affine Functions

In Claim 4, we showed that an obliviously-computable function f can be represented as the min of finitely many well-ordered quilt-affine functions. However, even if f is superadditive, the quilt-affine functions constructed in Claim 4 may not be superadditive. In this section we strengthen that result to show in Claim 5 that if f is superadditive, then f is the min of finitely many *superadditive* well-ordered quilt-affine functions, thereby proving the first half of our main result, Theorem 1.

▷ Claim 5. Any superadditive, obliviously-computable function is the minimum of a finite number of superadditive, well-ordered quilt-affine functions.

Proof. Let $f : \mathbb{N}^d \to \mathbb{N}$ be a superadditive, obliviously-computable function. From Claim 4, we know that $f = \min\{h_{\text{WO}}\}$, where each of the finitely many $h_{\text{WO}} : \mathbb{N}^d \to \mathbb{N}$ is a non-decreasing, well-ordered quilt-affine function. Let p be the period of the functions f and the h_{WO}'s. Since the h_{WO}'s may not be superadditive, we construct a superadditive, well-ordered quilt-affine function h_{S} from each h_{WO}, such that $f = \min\{h_{\text{S}}\}$.

With respect to some fixed h_{WO} and its well-ordered domain representation, say $\mathcal{WO}_{\mathbf{w}}$, we first partition the well-ordered domains into new types of domains that we will call *patches*. Then we define a superadditive function h_{S} as the union of partial affine functions on patches, such that $f(\mathbf{x}) \leq h_{\text{S}}(\mathbf{x}) \leq h_{\text{WO}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^d$. Finally we further partition the patches into well-ordered domains to show that h_{S} is well-ordered quilt-affine, completing the proof of the claim.

We define a patch as follows. Let $\overline{\mathbf{n}}$ be a congruence class mod p, i.e., $\overline{\mathbf{n}} = \{\mathbf{n} + p\mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^d\}$, where $\mathbf{n} \in \mathbb{N}^d$. The patch defined by a *corner* $\mathbf{q} \in \mathbb{N}^d \cap \overline{\mathbf{n}}$, a finite *set of excluding points* $Q \subset \mathbb{N}^d$, and $\overline{\mathbf{n}}$ is

$$P(\mathbf{q}, Q, \overline{\mathbf{n}}) = \{\mathbf{x} \in \mathbb{N}^d \cap \overline{\mathbf{n}} \mid \mathbf{q} \leq \mathbf{x} \text{ and } \mathbf{x} \ngeq \mathbf{q}', \forall \mathbf{q}' \in Q\}.$$

Figure 6 of the appendix illustrates a patch, and our overall transformation from h_{WO} to h_{S}.

For each domain Dom of the well-ordered representation of h_{WO} and each congruence class $\overline{\mathbf{n}}$ in $\mathbb{Z}^d/p\mathbb{Z}^d$, we cover $\text{Dom} \cap \overline{\mathbf{n}}$ with a finite number patches as follows. Initially, let the set Q of excluding points be the set of offsets of domains of h_{WO} that are greater than the offset of Dom. This ensures that only points in Dom are included in the constructed patches. While not all of $\text{Dom} \cap \overline{\mathbf{n}}$ is covered, select from the uncovered points the lexicographically first minimal point $\mathbf{q}$ that minimizes $h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q})$. Here, by minimal $\mathbf{q}$ we mean that there is no point $\mathbf{q}' < \mathbf{q}$, $\mathbf{q}' \in \text{Dom} \cap \overline{\mathbf{n}}$ with $h_{\text{WO}}(\mathbf{q}') - f(\mathbf{q}') \leq h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q})$, and if $\mathbf{q}_1$ and $\mathbf{q}_2$ are two distinct such minimal points then the lexicographically first one is the one with the smaller value at the first index between 1 and d where the two points differ. Since $h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q}) \geq 0$, the minimum exists if $\text{Dom} \cap \overline{\mathbf{n}}$ is not empty. Create the patch $P(\mathbf{q}, Q, \overline{\mathbf{n}})$. Then add $\mathbf{q}$ to Q (so that future patches exclude points in already-created patches), and repeat until all points of $\text{Dom} \cap \overline{\mathbf{n}}$ are covered.

Since the above algorithm is deterministic, for a given a patch corner, the associated set of excluding points and congruence class are uniquely determined, so we simply refer to a patch by its corner. Moreover, the number of patches generated by the algorithm is finite. To see why, we use the following lemma from Angluin et al., which is in turn a corollary of Higman's Lemma [13].

▶ **Lemma 6.** (Angluin et al. [2], Higman [13].) *Every subset of* $\mathbb{N}^d$ *under the inclusion ordering* $\leq$ *has finitely many minimal elements.*

The algorithm selects patch corners $\mathbf{q}$ with nondecreasing value of $h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q})$. The function f is bounded above by h_{WO}. So a lower bound for $h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q})$ is 0. If $\mathbf{x}_0$ is the minimum point of $\text{Dom} \cap \overline{\mathbf{n}}$, then when $\mathbf{x}_0$ is selected as a patch corner the algorithm must terminate. So the upper bound for $h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q})$ is $h_{\text{WO}}(\mathbf{x}_0) - f(\mathbf{x}_0)$. Since $h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q})$ is always integral, there are at most $h_{\text{WO}}(\mathbf{x}_0) - f(\mathbf{x}_0)$ different values for $h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q})$ during the algorithm. Consider the set of points $\mathbf{q}$ in $\mathbb{N}^d$ with the same value $h_{\text{WO}}(\mathbf{q}) - f(\mathbf{q})$. By Lemma 6 this set has a finite number of minimal points. So the number of patches produced by the algorithm is equal to the sum of the sizes of these finite minimal point sets, summed over the finite different values in the range $0, \ldots, h_{\text{WO}}(\mathbf{x}_0) - f(\mathbf{x}_0)$. Thus the algorithm terminates after a finite number of steps, when run on each $\text{Dom} \cap \overline{\mathbf{n}}$, and $\mathbb{N}^d$ is covered by the union of all the patches, taken over all domains of $\mathcal{WO}_{\mathbf{w}}$ and congruence classes $\overline{\mathbf{n}}$.

We define $h_{\text{S}} : P(\mathbf{q}, Q, \overline{\mathbf{n}}) \to \mathbb{N}$ by $h_{\text{S}}(\mathbf{x}) = h_{\text{WO}}(\mathbf{x}) - h_{\text{WO}}(\mathbf{q}) + f(\mathbf{q})$. If $\mathbf{q}$ is in domain Dom of h_{WO}'s well-ordered representation, where on domain $\text{Dom} \cap \overline{\mathbf{n}}$ we have that $h_{\text{WO}}(\mathbf{x})$ is the affine function $h_{\text{WO}}(\mathbf{x}) = \nabla \cdot \mathbf{x} + b$, then we can write

$$h_{\text{S}}(\mathbf{x}) = \nabla \cdot \mathbf{x} + b - h_{\text{WO}}(\mathbf{q}) + f(\mathbf{q}). \tag{3}$$

That is, $h_{\text{S}} : P(\mathbf{q}, Q, \overline{\mathbf{n}}) \to \mathbb{N}$ is an affine function with gradient ∇ and intercept $b - h_{\text{WO}}(\mathbf{q}) + f(\mathbf{q})$. Finally, we define $h_{\text{S}} : \mathbb{N}^d \to \mathbb{N}$ to be the union of these partial affine functions on patches. Next we prove several useful properties of h_{S}.

▶ **Lemma 7.** *For each patch corner* $\boldsymbol{q}$*,* $h_S(\boldsymbol{q}) = f(\boldsymbol{q})$*.*

Proof. Follows directly from the definition of h_{S}, since $h_{\text{S}}(\mathbf{q}) = h_{\text{WO}}(\mathbf{q}) - h_{\text{WO}}(\mathbf{q}) + f(\mathbf{q})$. ◀

▶ **Lemma 8.** *For all* $\mathbf{x} \in \mathbb{N}^d$*,* $h_S(\mathbf{x}) \leq h_{wo}(\mathbf{x})$*.*

Proof. Let $\mathbf{x}$ be in the patch with corner $\mathbf{q}$. Then $h_{\text{S}}(\mathbf{x}) = h_{\text{WO}}(\mathbf{x}) - h_{\text{WO}}(\mathbf{q}) + f(\mathbf{q}) \leq h_{\text{WO}}(\mathbf{x})$, since $h_{\text{WO}}(\mathbf{q}) \geq f(\mathbf{q})$. ◀

▶ **Lemma 9.** *For all* $\mathbf{x} \in \mathbb{N}^d$*,* $f(\mathbf{x}) \leq h_S(\mathbf{x})$*.*

Proof. Let $\mathbf{x}$ be in the patch with corner $\mathbf{q}$. Then by our choice of $\mathbf{q}$, $h_{\mathrm{WO}}(\mathbf{q}) - f(\mathbf{q}) \leq h_{\mathrm{WO}}(\mathbf{x}) - f(\mathbf{x})$. Rearranging the terms, we have that $f(\mathbf{x}) \leq h_{\mathrm{WO}}(\mathbf{x}) - h_{\mathrm{WO}}(\mathbf{q}) + f(\mathbf{q}) = h_{\mathrm{S}}(\mathbf{x})$. ◀

▶ **Lemma 10.** *Let $\mathbf{x}, \mathbf{x}' \in \mathbb{N}^d$ and let $\mathbf{x} \leq \mathbf{x}'$. Then the gradient of h_S on the patch containing $\mathbf{x}$ is less than or equal to the gradient of h_S on the patch containing $\mathbf{x}'$.*

Proof. Suppose that $\mathbf{x}$ and $\mathbf{x}'$ are in domains $\mathrm{Dom}_{\mathbf{o}}$ and $\mathrm{Dom}_{\mathbf{o}'}$ in the well-ordered domain representation of h_{WO}. Then since $\mathbf{x} \leq \mathbf{x}'$, the gradient of h_{WO} on $\mathrm{Dom}_{\mathbf{o}}$ is less than or equal to the gradient of h_{WO} on $\mathrm{Dom}_{\mathbf{o}'}$ (the construction of Claim 4 satisfies this property). By construction of h_{S} in Equation (3), the gradient of h_{S} on a patch equals the gradient of h_{WO} in the domain containing the patch, and so the lemma follows. ◀

▶ **Lemma 11.** *Let $\mathbf{x}, \mathbf{x}' \in \mathrm{Dom} \cap \overline{\mathbf{n}}$, for some $\mathrm{Dom} \in \mathcal{WO}_{\mathbf{w}}$ and congruence class $\overline{\mathbf{n}}$. Suppose also that $\mathbf{x} \leq \mathbf{x}'$. Then the intercept of h_S on $\mathbf{x}$ is less than or equal to the intercept of h_S on $\mathbf{x}'$.*

Proof. The stated conditions of the lemma on $\mathbf{x}$ and $\mathbf{x}'$ imply that either $\mathbf{x}$ and $\mathbf{x}'$ are in the same patch, or the patch containing $\mathbf{x}$ is constructed *after* the patch containing $\mathbf{x}'$. The intercepts of h_{S} on patches within $\mathrm{Dom} \cap \overline{\mathbf{n}}$ are nonincreasing in the order of patch construction. ◀

▶ **Lemma 12.** *h_S is superadditive.*

Proof. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be in patches $\mathbf{q}_1$ and $\mathbf{q}_2$, respectively. Then $\mathbf{q}_1 + \mathbf{q}_2 \leq \mathbf{x}_1 + \mathbf{x}_2$ and $\mathbf{q}_1 + \mathbf{q}_2$ and $\mathbf{x}_1 + \mathbf{x}_2$ are in the same congruence class. Also, by Lemma 3, the points $x_1 + x_2$ and $q_1 + q_2$ lie in the same domain of h_{WO}. Let $\mathbf{x}_1 + \mathbf{x}_2$ be in the patch with corner $\mathbf{q}$.

On the patches with corners $\mathbf{q}_1$, $\mathbf{q}_2$, and $\mathbf{q}$, let $h_{\mathrm{S}}(\mathbf{x}_1) = \nabla_1 \cdot \mathbf{x} + b_1$, $h_{\mathrm{S}}(\mathbf{x}_2) = \nabla_2 \cdot \mathbf{x} + b_2$, and $h_{\mathrm{S}}(\mathbf{x}) = \nabla \cdot \mathbf{x} + b$, respectively. By Lemma 10, $\nabla_1 \leq \nabla$ and $\nabla_2 \leq \nabla$. Also, we have that

$$\begin{aligned} h_{\mathrm{S}}(\mathbf{q}_1) + h_{\mathrm{S}}(\mathbf{q}_2) &= f(\mathbf{q}_1) + f(\mathbf{q}_2) && \text{(by Lemma 7)} \\ &\leq f(\mathbf{q}_1 + \mathbf{q}_2) && \text{(since } f \text{ is superadditive)} \\ &\leq h_{\mathrm{S}}(\mathbf{q}_1 + \mathbf{q}_2) && \text{(by Lemma 9)} \\ &\leq \nabla \cdot (\mathbf{q}_1 + \mathbf{q}_2) + b, \end{aligned}$$

where the last inequality follows by Lemmas 10 and 11. Then

$$\begin{aligned} h_{\mathrm{S}}(\mathbf{x}_1) + h_{\mathrm{S}}(\mathbf{x}_2) &= h_{\mathrm{S}}(\mathbf{x}_1) - h_{\mathrm{S}}(\mathbf{q}_1) + h_{\mathrm{S}}(\mathbf{x}_2) - h_{\mathrm{S}}(\mathbf{q}_2) + h_{\mathrm{S}}(\mathbf{q}_1) + h_{\mathrm{S}}(\mathbf{q}_2) \\ &= \nabla_1 \cdot (\mathbf{x}_1 - \mathbf{q}_1) + \nabla_2 \cdot (\mathbf{x}_2 - \mathbf{q}_2) + h_{\mathrm{S}}(\mathbf{q}_1) + h_{\mathrm{S}}(\mathbf{q}_2) \\ &\leq \nabla \cdot (\mathbf{x}_1 - \mathbf{q}_1) + \nabla \cdot (\mathbf{x}_2 - \mathbf{q}_2) + \nabla \cdot (\mathbf{q}_1 + \mathbf{q}_2) + b \\ &= \nabla \cdot (\mathbf{x}_1 + \mathbf{x}_2) + b \\ &= h_{\mathrm{S}}(\mathbf{x}_1 + \mathbf{x}_2). \end{aligned}$$

◀

▶ **Lemma 13.** *h_S is well-ordered quilt-affine.*

Proof. Define $\mathbf{w}'$ to be the vector whose ith component w_i' is $\max_{\mathbf{q}} q_i$, rounded up to be 0 mod p. The domain set $\mathcal{WO}_{\mathbf{w}'}$ is a refinement of the original domain set $\mathcal{WO}_{\mathbf{w}}$ of h_{WO}'s representation. Let $\mathrm{Dom}_{\mathbf{o}'}$ be one of the domains of $\mathcal{WO}_{\mathbf{w}'}$ (where $\mathbf{o}' \leq \mathbf{w}'$), and let $\mathrm{Dom}_{\mathbf{o}'} \subset \mathrm{Dom}_{\mathbf{o}}$, where $\mathrm{Dom}_{\mathbf{o}} \in \mathcal{WO}_{\mathbf{w}}$.

Fix any congruence class $\overline{\mathbf{n}}$ of $\mathbb{Z}^d/pZ^d$. If $\mathrm{Dom}_{\mathbf{o}'} \cap \overline{\mathbf{n}}$ is not empty, let $\mathbf{m}$ be the smallest point in $\mathrm{Dom}_{\mathbf{o}'} \cap \overline{\mathbf{n}}$. Let $\mathbf{q}$ be the corner of the patch containing $\mathbf{m}$. Note that $\mathbf{q}$ is in $\mathrm{Dom}_{\mathbf{o}}$.

We claim that $\mathrm{Dom}_{\mathbf{o}'} \cap \overline{\mathbf{n}}$ is contained in the patch with corner $\mathbf{q}$. This is trivially true if $\mathrm{Dom}_{\mathbf{o}'} \cap \overline{\mathbf{n}}$ is finite, and thus a single point. Consider the case where $\mathrm{Dom}_{\mathbf{o}'}$ is infinite. Let

$\mathbf{x} \in \text{Dom}_{\mathbf{o}'} \cap \overline{\mathbf{n}}$ and let $\mathbf{q}'$ be the corner of the patch containing $\mathbf{x}$. We claim that $\mathbf{q}' \leq \mathbf{m}$. To see why, note that if $x_i > m_i$ then it must be that $m_i \geq w'_i$ and by our choice of $\mathbf{w}'$, $q'_i \leq m_i$. Otherwise, $x_i \leq m_i$ and so $q'_i \leq x_i \leq m_i$. But then $\mathbf{q} = \mathbf{q}'$, since $\mathbf{q}$ is the corner of the patch containing $\mathbf{m}$. Therefore all of $\text{Dom}_{\mathbf{o}'} \cap \overline{\mathbf{n}}$ is in the patch with corner $\mathbf{q}$. It follows that h_S on domain $\text{Dom}_{\mathbf{o}'} \cap \overline{\mathbf{n}}$ is a single affine function, namely that associated with the patch with corner $\mathbf{q}$. Moreover, the gradient of this function is the gradient of the function h_{WO} on domain $\text{Dom}_{\mathbf{o}} \in \mathcal{WO}_{\mathbf{w}}$. Since this is true for any congruence class $\overline{\mathbf{n}}$, the function h_S on domain $\text{Dom}_{\mathbf{o}'}$ is a quilt-affine function whose gradient is the same as that of f on $\text{Dom}_{\mathbf{o}}$, completing the proof. ◀

From Lemmas 8 and 9, we have that $f = \min\{h_S\}$ where the min is taken over a finite number of functions h_S. Moreover, from Lemma 12, each h_S is superadditive and from Lemma 13, each h_S is well-ordered quilt-affine. The proof of Claim 5 follows. ◁

4 A Leaderless Output-Oblivious CRN for Superadditive, Obliviously-Computable Functions

Here we show the second half of our main result, Theorem 1, by constructing a leaderless, output-oblivious CRN for any superadditive, well-ordered quilt-affine function.

▷ Claim 14. Any superadditive, well-ordered quilt-affine function can be stably computed by a leaderless, output-oblivious CRN.

Proof. Let $f : \mathbb{N}^d \to \mathbb{N}$ be a superadditive, obliviously-computable function. From Claim 5, we know that $f = \min\{h_S\}$, where each of the finitely many h_S is a superadditive, well-ordered quilt-affine function. Below we show that any such function has a leaderless, output-oblivious CRN, say $\mathcal{C}_{h_S}$. A leaderless, output-oblivious CRN for f can then be obtained from the $\mathcal{C}_{h_S}$'s via the following steps: (i) for each function h_S, create a unique replica $X_{h_S,i}$ of each input species X_i ; (ii) adapt $\mathcal{C}_{h_S}$ by replacing input species X_i by the replica $X_{h_S,i}$, in every reaction and for each i and replacing the output species Y of $\mathcal{C}_{h_S}$ with Y_{h_S} in every reaction; and (iii) adding the reaction $\sum_{h_S} Y_{h_S} \to Y$, which implements the min function.

Fix any superadditive, well-ordered quilt-affine function h, and a representation of h with well-ordered domain set $\mathcal{WO}_{\mathbf{w}}$ and period $p \in \mathbb{N}^+$. To simplify our proof we will assume without loss of generality that $p > 1$. Recall that there is one domain $\text{Dom}_{\mathbf{o}}$ in h's representation for each $\mathbf{o} \leq \mathbf{w}$. We partition these domains by taking intersections with congruence classes mod p. For each $\text{Dom}_{\mathbf{o}} \in \mathcal{WO}_{\mathbf{w}}$ and each congruence class $\overline{\mathbf{x}}$ of $\mathbb{Z}^d/p\mathbb{Z}^d$ such that $\text{Dom}_{\mathbf{o}} \cap \overline{\mathbf{x}}$ is non-empty, let $\mathbf{m} = \mathbf{m}(\mathbf{o}, \overline{\mathbf{x}})$ be the minimum point in the subdomain $\text{Dom}_{\mathbf{o}} \cap \overline{\mathbf{x}}$, and denote this subdomain by $\text{Dom}'_{\mathbf{m}}$. Let $\mathcal{N}$ be the set of all such $\mathbf{m}$. By our assumption that $p > 1$, it must be that all unit vectors $\mathbf{e}_i$ are in $\mathcal{N}$, $1 \leq i \leq d$. Since h is quilt-affine with period p, we have that $h(\mathbf{x})$ on $\text{Dom}'_{\mathbf{m}}$ is a partial affine function, which we denote by $h_{\mathbf{m}}(\mathbf{x}) = \nabla_{\mathbf{m}}(\mathbf{x}) + b_{\mathbf{m}}$, where $\nabla_{\mathbf{m}} = \nabla_{\mathbf{o}}$ if $\mathbf{m} = \mathbf{m}(\mathbf{o}, \overline{\mathbf{x}})$.

Our CRN has input species $X_1, X_2, \ldots, X_d$ and an output species Y. We will use $\mathbf{x}$ to denote the vector of counts of input species consumed, and y to denote the number of Y's produced, during an execution of the CRN. Our CRN also has a *leader* species $L_{\mathbf{m}}$ for $\mathbf{m} \in \mathcal{N}$, and a *distance* species $P_{\mathbf{m},i}$ for each $\mathbf{m} \in \mathcal{N}$ and each $i \in \{1, \ldots, d\}$. We will use $\#L_{\mathbf{m}}$ and $\#P_{\mathbf{m},i}$ to denote counts of leader and distance species, during an execution of the CRN.

The leader and distances species will track how much input has been consumed by reactions. To build intuition on how this works, it may be helpful first to imagine that there is just one leader. In this case, if the input $\mathbf{x}$ consumed so far is in domain $\text{Dom}'_{\mathbf{m}}$, then our reactions will ensure that the leader is $L_{\mathbf{m}}$ and that for $1 \leq i \leq d$, $\#P_{\mathbf{m},i} = (x_i - m_i)/p$, i.e., the distance of the consumed input $\mathbf{x}$ from $\mathbf{m}$, along the ith dimension. (Since $\mathbf{x} \in \mathrm{Dom}'_{\mathbf{m}}$, $x_i - m_i$ is a multiple of p.) Thus,

$$\mathbf{x} = \mathbf{m} + p \sum_{1 \leq i \leq d} \#P_{\mathbf{m},i} \times \mathbf{e}_i.$$

Generalizing to the leaderless scenario, consumption of input will produce many leaders; we can imagine that the consumed input is distributed over many domains $\mathrm{Dom}_{\mathbf{m}}$. Our reactions will ensure that a generalization of the above equality holds:

$$\mathbf{x} = \sum_{\mathbf{m} \in \mathcal{N}} \left(\#L_{\mathbf{m}} \times \mathbf{m} + p \sum_{i \in \{1,\dots,d\}} \#P_{\mathbf{m},i} \times \mathbf{e}_i \right), \tag{4}$$

and we call the term on the right hand side of this invariant the *input value* of the CRN configuration. The invariant trivially holds initially since both $\mathbf{x}$ and the input value are $\mathbf{0}$. Our reactions will also maintain the following *output invariant*:

$$y = \sum_{\mathbf{m} \in \mathcal{N}} \left(\#L_{\mathbf{m}} \times h(\mathbf{m}) + \sum_{i \in \{1,\dots,d\}} \#P_{\mathbf{m},i} \times \nabla_{\mathbf{m},i} \right). \tag{5}$$

We call the term on the right hand side of this invariant the *output value*. Initially both y and the output value are 0. We will show that once our CRN stabilizes, the output value is the function h applied to the input value, and so these invariants ensure that $y = h(\mathbf{x})$ upon stabilization.

Our CRN has three types of reactions. We next describe these, and show that each respects the input and output invariants. Figure 4, included in the appendix, shows an example of a function h, a quilt-affine representation and the partitioning of the quilt-affine domains (via intersections with congruence classes), and Figure 5, also in the appendix, illustrates part of our CRN construction for the function of Figure 4.

4.1 Input-Consuming Reactions

These reactions consume inputs and produce leader species. There is one reaction for each $i, 1 \leq i \leq d$:

$$X_i \to L_{\mathbf{e}_i} + h(\mathbf{e}_i) \times Y.$$

This reaction consumes input $\mathbf{e}_i$, and recall that by our assumption that $p > 1$, $\mathbf{e}_i \in \mathcal{N}$. So, no distance species are needed to ensure that the input invariant holds. Producing $h(\mathbf{e}_i)$ Y's ensures that the output invariant holds.

4.2 Merge Reactions

Merge reactions reduce the number of leader species, effectively electing a single leader:

$$L_{\mathbf{m}} + L_{\mathbf{m}'} \to L_{\mathbf{m}''} + \delta \times Y + \sum_{j \in \{1..d\}} \delta'_j P_{\mathbf{m}'',j}.$$

Here, $\mathbf{m}''$ is chosen such that $\mathrm{Dom}'_{\mathbf{m}''}$ contains $\mathbf{m} + \mathbf{m}'$. To ensure that the input invariant holds upon a merge reaction, we choose $\delta'_j = (n_j + n'_j - n''_j)/p$. Plugging this value into the input invariant (4) shows that the input value is unchanged, which is necessary since no input is consumed. To ensure that the output invariant holds, we set δ to be equal to the change in the output value as a result of the reaction (increase due to addition of products minus decrease due to removal of reactants):

$$\begin{aligned}\delta &= h(\mathbf{m}'') + p\textstyle\sum_{j\in\{1..d\}} \delta'_j \times \nabla_{\mathbf{m}'',j} - h(\mathbf{m}) - h(\mathbf{m}')\\ &= h(\mathbf{m}+\mathbf{m}') - h(\mathbf{m}) - h(\mathbf{m}').\end{aligned}$$

In this case, δ is non-negative because f is superadditive.

4.3 Exchange Reactions

The exchange reactions ensure that, once there is a single leader molecule, say $L_{\mathbf{m}}$, eventually all of the distance species $P_{\mathbf{m}',i}$ are such that $\mathbf{m}' = \mathbf{m}$. Let $\mathbf{m}$ and $\mathbf{m}'$ be in $\mathcal{N}$, with $\mathbf{m} \neq \mathbf{m}'$. Let $\mathbf{m}$ and $\mathbf{m}'$ be in the well-ordered domains of $\mathcal{WO}_{\mathbf{w}}$ with offsets $\mathbf{o}$ and $\mathbf{o}'$, respectively. Recall that $\nabla_{\mathbf{m}} = \nabla_{\mathbf{o}}$ and $\nabla_{\mathbf{m}'} = \nabla_{\mathbf{o}'}$. Suppose without loss of generality that $\mathbf{o} \leq \mathbf{o}'$ (in which case $\nabla_{\mathbf{o}} \leq \nabla_{\mathbf{o}'}$), and that if $\mathbf{o} = \mathbf{o}'$ then $\mathbf{m} \leq \mathbf{m}'$. Then we add the following reactions, for $1 \leq i \leq d$:

$$L_{\mathbf{m}} + P_{\mathbf{m}',i} \to L_{\mathbf{m}} + \delta \times Y + P_{\mathbf{m},i}.$$

Each exchange reaction preserves the input invariant because the input value is unchanged and no input is consumed. To ensure that the output invariant holds, we set δ to equal the change in the output value as a result of the reaction (increase due to addition of products minus decrease due to removal of reactants):

$$\begin{aligned}\delta &= h(\mathbf{m}) + p\nabla_{\mathbf{m},i} - h(\mathbf{m}) - p\nabla_{\mathbf{m}',i}\\ &= p(\nabla_{\mathbf{m},i} - \nabla_{\mathbf{m}',i})\\ &= p(\nabla_{\mathbf{o},i} - \nabla_{\mathbf{o}',i})\\ &\geq 0, && \text{since } \nabla_{\mathbf{o}} \geq \nabla_{\mathbf{o}'}.\end{aligned}$$

This completes the description of the reactions of the CRN.

4.4 Correctness

A "leader dominance" invariant that is maintained by all reactions is that for any $P_{\mathbf{m}',i}$ with positive count, there is also some leader $L_{\mathbf{m}}$ with positive count, such that if $\mathrm{Dom}_{\mathbf{o}}$ and $\mathrm{Dom}_{\mathbf{o}'}$ are the well-ordered domains containing $\mathbf{m}$ and $\mathbf{m}'$, respectively then $\mathbf{o} \geq \mathbf{o}'$. The input consuming and exchange reactions trivially maintain this invariant. Consider a merge reaction with reactants $L_{\mathbf{m}}$ and $L_{\mathbf{m}'}$ that produces $L_{\mathbf{m}''}$. Suppose that $\mathbf{m}$, $\mathbf{m}'$, and $\mathbf{m}''$ are in the well-ordered domains with offsets $\mathrm{Dom}_{\mathbf{o}}$, $\mathrm{Dom}_{\mathbf{o}'}$ and $\mathrm{Dom}_{\mathbf{o}''}$, respectively. Then by Lemma 3, since $\mathrm{Dom}'_{\mathbf{m}''}$ contains $\mathbf{m}+\mathbf{m}'$, $\mathrm{Dom}_{\mathbf{o}''}$ must contain $\mathbf{o}+\mathbf{o}'$. Therefore, $\mathbf{o}'' = (\mathbf{o}+\mathbf{o}') \diamond \mathbf{w}$, where we use $\diamond$ to denote the element-wise min. So it must be that $\mathbf{o}'' \geq \mathbf{o}'$, and the leader dominance invariant must hold upon a merge reaction.

Next we show that the CRN stabilizes. First note that eventually all input species are consumed by the input-consuming reactions, at which point no more leaders will be produced. Also, eventually there is exactly one leader, because of the merge reactions. At this point, the only possible reactions are exchange reactions. Each exchange reaction reduces the number of $P_{\mathbf{m}',i}$ with $\mathbf{m}' \neq \mathbf{m}$. By the leader dominance invariant, this number will eventually reach zero, at which point no more exchange reactions are possible.

Suppose that, once no more reactions are possible, the leader is $L_{\mathbf{m}}$, in which case the only distance species with count greater than zero are species $P_{\mathbf{m},i}$ for some i. As a result, we have that

$$\begin{aligned} y &= h(\mathbf{m}) + p\textstyle\sum_{i\in 1..d} \#P_{\mathbf{m},i} \times \nabla_{\mathbf{m},i} && \text{from the output invariant} \\ &= h(\mathbf{m})\nabla_{\mathbf{n}}p(\textstyle\sum_{i\in 1..d} \#P_{\mathbf{m},i} \times \mathbf{e}_i + \mathbf{m} - \mathbf{m}) \\ &= h(\mathbf{m}) + \nabla_{\mathbf{m}} \times (\mathbf{x} - \mathbf{m}) && \text{from the input invariant} \\ &= h(\mathbf{x}). \end{aligned}$$

This ensures that the output is correct once the CRN has stabilized, completing the proof. ◁

5 Conclusion

We have classified the functions $f : \mathbb{N}^d \to \mathbb{N}$ which are stably computable by CRNs that are (a) leaderless, and (b) never consume their own output. This result sheds light on the fundamental limitations of discrete CRNs. Indeed, together with previous work on CRNs with leaders [16], this has completed the classification of functions which are stably computable by output-oblivious CRNs – with and without leaders. Such results inform the larger question of composability in this model of computation, and to what extent such systems can be comprised of smaller, modular components.

While composition with guaranteed correctness seems dubious for functions which are not output-oblivious, we emphasize that there are nevertheless routes to composition with a high probability of correctness. Phase-clocks for example, a ubiquitous tool in population protocols (e.g., [3, 12, 1]), may be used to prohibit a CRN from being activated for some number of time steps. Kosowski and Uznański recently demonstrated how to build hierarchies of phase clocks; these could be leveraged to construct an arbitrarily long series of CRN compositions [14].

A question raised by our results is the extent to which the theory of discrete and continuous CRNs can be reconciled. As mentioned in the introduction, our results mirror those for continuous CRNs, but our techniques are quite distinct. It would be useful to know whether and under what conditions certain statements apply to both models. Is their a theoretical framework allowing both continuous and discrete CRNs to be studied simultaneously?

A separate question is whether CRNs which compute output-oblivious functions, but are not themselves output-oblivious, can be augmented with reactions to make them so. For CRNs implemented as strand displacement systems, for instance, it may be easier to add reactions than to change the underlying network entirely. Understanding the limitations of being able to edit in this way would shed light on the possibility of building CRNs incrementally instead of requiring that the design be understood beforehand.

References

1 Dan Alistarh, James Aspnes, and Rati Gelashvili. Space-optimal majority in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2221–2239. SIAM, 2018.

2 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.

3 Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199, 2008.

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

5 Stefan Badelt, Seung Woo Shin, Robert F. Johnson, Qing Dong, Chris Thachuk, and Erik Winfree. A general-purpose CRN-to-DSD compiler with formal verification, optimization, and simulation capabilities. In Robert Brijder and Lulu Qian, editors, *DNA Computing and Molecular Programming*, pages 232–248, Cham, 2017. Springer International Publishing.

6 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.

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

8 Ho-Lin Chen, David Doty, and David Soloveichik. Rate-independent computation in continuous chemical reaction networks. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ITCS 2014, pages 313–326, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2554797.2554827.

9 Ben Chugg, Hooman Hashemi, and Anne Condon. Output-oblivious stochastic chemical reaction networks. In Jiannong Cao, Faith Ellen, Luis Rodrigues, and Bernardo Ferreira, editors, *22nd International Conference on Principles of Distributed Systems, OPODIS 2018, December 17-19, 2018, Hong Kong, China*, volume 125 of *LIPIcs*, pages 21:1–21:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.OPODIS.2018.21.

10 Matthew Cook, David Soloveichik, Erik Winfree, and Jehoshua Bruck. Programmability of chemical reaction networks. *Algorithmic Bioprocesses*, pages 543–584, 2009.

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

12 Leszek Gąsieniec and Grzegorz Staehowiak. Fast space optimal leader election in population protocols. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2653–2667. SIAM, 2018.

13 G Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 3(2):326–336, 1952.

14 Adrian Kosowski and Przemysław Uznański. Population protocols are fast. *arXiv preprint arXiv:1802.06872*, 2018.

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

16 Eric E. Severson, David Haley, and David Doty. Composable computation in discrete chemical reaction networks. In Peter Robinson and Faith Ellen, editors, *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 14–23. ACM, 2019. doi:10.1145/3293611.3331615.

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

18 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.

19 David Zhang and Georg Seelig. Dynamic DNA nanotechnology using strand-displacement reactions. *Nature chemistry*, 3:103–13, February 2011. doi:10.1038/nchem.957.

A Appendix

$$h(x_1, x_2) = \begin{cases} x_1, & x_2 = 0 \\ x_2, & x_1 = 0 \\ h'(x_1, x_2), & x_1 \geq 1, x_2 \geq 1. \end{cases}$$

(a) A superadditive, output-oblivious function h, where $h'(x_1, x_2) = 2x_1 + 2x_2 - ((x_1 + x_2) \bmod 2)$.

$$h(x_1, x_2) = \begin{cases} 0, & (x_1, x_2) \in \mathrm{Dom}_{00} = \{(0,0)\} \\ x_1, & (x_1, x_2) \in \mathrm{Dom}_{01} = \{(x_1, 0) + (1,0) \mid x_1 \in \mathbb{N}\} \\ x_2, & (x_1, x_2) \in \mathrm{Dom}_{10} = \{(0, x_2) + (0,1) \mid x_2 \in \mathbb{N}\} \\ h'(x_1, x_2), & (x_1, x_2) \in \mathrm{Dom}_{11} = \{(x_1, x_2) + (1,1) \mid x_1, x_2 \in \mathbb{N}\}. \end{cases}$$

(b) A well-ordered, quilt-affine representation of h. The domain set $\mathcal{WO}_{\mathbf{w}}$ has period 2, $\mathbf{w} = 11$ and contains four domains $\mathrm{Dom}_{\mathbf{o}}$ as shown, for $\mathbf{o} \in \{00, 01, 10, 11\}$.

$$h(x_1, x_2) = \begin{cases} 0, & (x_1, x_2) \in \mathrm{Dom}'_{00} = \mathrm{Dom}_{00} \cap \overline{00} \\ x_1, & (x_1, x_2) \in \mathrm{Dom}'_{01} = \mathrm{Dom}_{01} \cap \overline{01} \\ x_1, & (x_1, x_2) \in \mathrm{Dom}'_{02} = \mathrm{Dom}_{01} \cap \overline{00} \\ x_2, & (x_1, x_2) \in \mathrm{Dom}'_{10} = \mathrm{Dom}_{10} \cap \overline{10} \\ x_2, & (x_1, x_2) \in \mathrm{Dom}'_{20} = \mathrm{Dom}_{10} \cap \overline{00} \\ 2x_1 + 2x_2, & (x_1, x_2) \in \mathrm{Dom}'_{11} = \mathrm{Dom}_{11} \cap \overline{11} \\ 2x_1 + 2x_2 - 1, & (x_1, x_2) \in \mathrm{Dom}'_{12} = \mathrm{Dom}_{11} \cap \overline{10} \\ 2x_1 + 2x_2 - 1, & (x_1, x_2) \in \mathrm{Dom}'_{21} = \mathrm{Dom}_{11} \cap \overline{01} \\ 2x_1 + 2x_1, & (x_1, x_2) \in \mathrm{Dom}'_{22} = \mathrm{Dom}_{11} \cap \overline{00} \end{cases}$$

(c) Representation of h on nonempty domains of the form $\mathrm{Dom}'_{\mathbf{n}} = \mathrm{Dom}'_{\mathbf{n}(\mathbf{o},\overline{\mathbf{z}})} = \mathrm{Dom}_{\mathbf{o}} \cap \overline{\mathbf{z}}$, for each congruence class $\overline{\mathbf{z}}$ of $\mathbb{Z}^2/2\mathbb{Z}^2$, where $\overline{\mathbf{z}} = \{2(x_1, x_2) + \mathbf{z} \mid x_1, x_2 \in \mathbb{N}\}$ for each $\mathbf{z} = 00, 01, 10, 11$, and $\mathbf{n} = \mathbf{n}(\mathbf{o}, \overline{\mathbf{z}})$ is the minimum point in $\mathrm{Dom}_{\mathbf{o}} \cap \overline{\mathbf{z}}$.

Figure 4 (a) A superadditive, output-oblivious function h. (b) Quilt-affine representation of h. (c) Representation of h used in our leaderless CRN construction. Here as in Figure 5, we use strings to denote vectors, e.g. 11 denotes $(1, 1)$.

Input-consuming Reactions	Sample Merge Reactions (involving L_{01} or L_{11})	Sample Exchange Reactions (involving L_{11} or L_{22})
	$L_{01} + L_{10} \to L_{11} + 2Y$	
	$L_{01} + L_{x1} \to L_{x2}, x \in \{0, 1\}$	$L_{11} + P_{01,x} \to P_{11,x} + 2Y$
	$L_{01} + L_{21} \to L_{22} + 2Y$	$L_{11} + P_{10,x} \to P_{11,x} + 2Y$
	$L_{01} + L_{02} \to L_{01} + P_{01,2}$	
$X_1 \to L_{10} + Y$	$L_{01} + L_{x2} \to L_{x1} + 2Y + P_{x1,2}, x \in \{1, 2\}$	$L_{22} + P_{01,x} \to P_{22,x} + 2Y$
$X_2 \to L_{01} + Y$		$L_{22} + P_{10,x} \to P_{22,x} + 2Y$
	$L_{11} + L_{01} \to L_{21}$	$L_{22} + P_{11,x} \to P_{22,x}$
	$L_{11} + L_{11} \to L_{22}$	$L_{22} + P_{21,x} \to P_{22,x}$
	$L_{11} + L_{21} \to L_{12} + P_{21,1}$	$L_{22} + P_{12,x} \to P_{22,x}$
	$L_{11} + L_{12} \to L_{21} + P_{21,2}$	
	$L_{11} + L_{22} \to L_{11} + 2Y + P_{11,x}, x \in \{1, 2\}$	

Figure 5 Sample reactions of the leaderless, output-oblivious CRN for the function h of Figure 4, obtained from our construction of Claim 14.

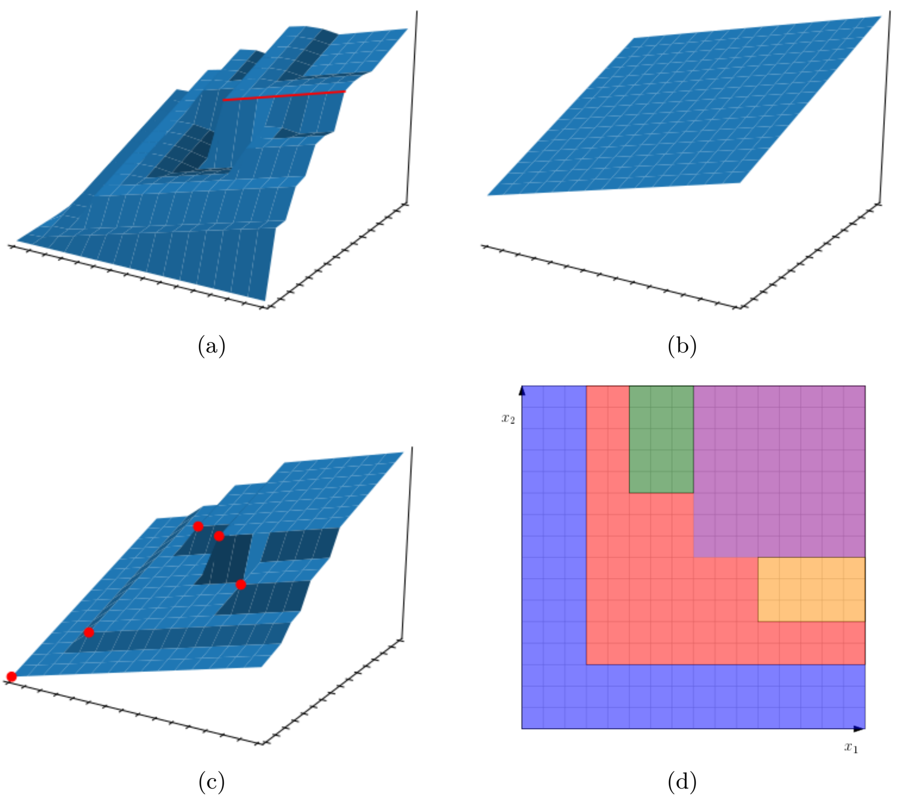


Figure 6 (a) An output-oblivious function $f(\mathbf{x})$. (b) By Claim 4, the function f of part (a) can be written as $f = \min\{h_{\mathrm{WO}}\}$, where each of the finitely many functions h_{WO} is nondecreasing, well-ordered quilt-affine. One of these functions is shown here. This function happens to be quite simple, with period 1 and one domain, namely $\mathbb{N}^2$, and $f = h_{\mathrm{WO}}$ on the red line shown in part (a). (c) The superadditive, obliviously-computable function h_{S} that is derived from the function h_{WO} of part (b) via the construction of Claim 5. Patch corners are shown as red dots. The function h_{S} has the same gradient as h_{WO} on each patch, but has different intercepts. (d) Each coloured region is a patch on $\mathbb{N}^2$ (i.e., the congruence class has period 1). These patches correspond to the corners of part (c).