# Sufficiency Condition for Output-Oblivious Chemical Reaction Networks

## and Run-time Analysis

by

Hooman Hashemi

B.Sc., Sharif University of Technology, 2017

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

November 2019

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Sufficiency Condition for Output-Oblivious Chemical Reaction Networks and Run-time Analysis

| | | |
|---|---|---|
| submited by | Hooman Hashemi | in partial fulfillment of the requirements for |
| the degree of | Master of Science | |
| in | Computer Science | |

**Examining Committee:**

Anne Condon, Computer Science
Supervisor

Jim Little, Computer Science
Supervisory Committee Member

# Abstract

This thesis provides a sufficiency condition for 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 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).
This result complements the necessity condition obtained and proven by other contributors.
The run-time analysis provided in the end adds more detail to the proof and the construction.

# Lay Summary

Principles of computation are relevant not only for today's silicon-based computers, but also for programming molecules to compute in novel ways that are not found in nature. Chemical reaction networks have traditionally been used to study extant chemical systems, but are also now being harnessed to describe molecular programs that compute mathematical functions. A desirable property of such programs is that they be composable, that is, the output of one chemical program serves as the input to a second program. In this thesis we contribute to knowledge about what functions can be stably computed in a composable way by chemical reaction networks. Chemical reaction networks are in definition close to population protocols which are another model that is used for distributed computation. Population protocols model parallel systems that each agent has a state and the states change by interaction between agents. Because of the similarity our results may be also applicable here.

# Preface

The research conducted in this article and the problem specified here were defined by professor Anne Condon and the student Ben Chugg.
I was one of the main contributors to the solution of the larger research problem. All content provided in the main chapters of this article was solved solely by me (that is excluding introduction and related work).
The research presented in this thesis has led to the following publication. In this publication I have acted as a coauthor, providing assistance and contributing to the final proof.

- Ben Chugg, Hooman Hashemi, and Anne Condon, 22nd International Conference on Principles of Distributed Systems, Conference on Principles of Distributed Systems (OPODIS 2018)
  Appears in the proceeding as: "Output-Oblivious Stochastic Chemical Reaction Networks"

For this publication I proved the sufficiency condition, and come up with the structure called semiaffine functions which helped to connect the two sides of the proof.
Most of the writing in the paper was done by the other two coauthors.

# Table of Contents

## Appendix: Proofs of Lemmas

# List of Figures

# Acknowledgements

I would like to take this opportunity to acknowledge and express my sincere gratitude toward my research supervisor professor Anne Condon, for her constant support, unparalleled kindness and consistent guidance. Her support exceeded expectation and I'm really thankful for this.

I want to thank professor Jim Little, for supervising my program. He helped me and guided me through my studies. He supported me to enter a second field of study and helped me to gain experience.

Finally special thanks to my family for their patience during my program. Without their support I would have not been able to be where I am today.

# Dedication

*To my family,*
*for their constant love and support.*

# Chapter 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, 20]. They are closely related to Population Protocols [1, 3, 4, 10], another distributed computing model; these models have found applications in areas as diverse as signal processing [16], graphical models [17], neural networks [15], and modeling cellular processes [5, 6]. CRNs can simulate Universal Turing Machines [2, 20]. 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., [18, 21–23]) 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.

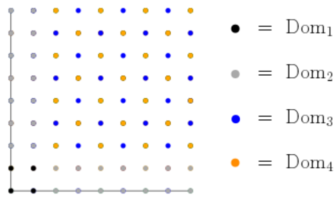
It is thus natural to ask: What functions can be stably computed in an *output-oblivious manner*, in which outputs are never reactants, without using diff-representation?

This question is the focus of this work. Doty and Hajiaghayi [13] 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 that can be stably computed by stochastic CRNs with leaders.

## 1.1 Our Results

In this work we present the sufficiency condition for a characterization of 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 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.

Later Severson et al. [19] characterized the functions with more than two inputs that can be stably computed in an output-oblivious manner.




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

Figure 1.1: 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.

Their results follow a simpler path; They do not use the notation of fissure functions which we will introduce later.

Our results also hold for Population Protocols, since stable function-computing CRNs can be translated into Population Protocols and vice versa. Section 2.1 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 Chapter 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 $f_i$, $1 \le i \le k$, 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 1.1 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 1.2.

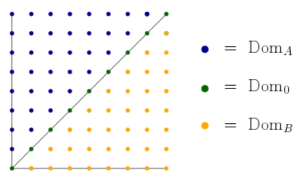
However, constructing CRNs for other fissure functions appears to be significantly trickier than that shown in Figure 1.2. Consider the function $f(n_1, n_2) = 2n_1 + 3n_2 + 2$ if $n_1 > n_2$, $f(n_1, n_2) = 3n_1 + 2n_2 + 2$ if $n_1 < n_2$ and $f(n_1, n_2) = 5n_1$ on the "fissure line" $n_1 = n_2$. The simple line-tracking mechanism of the CRN of Figure 1.2 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 1.2, 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 and the focus of this thesis is a construction that can handle such fissure functions, as well as functions with multiple parallel fissure lines.

In Section 2.2 we briefly present results on the negative side obtained by the other authors. 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, I state the main result jointly obtained with other authors here (see Section 2.1 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. A function is output-monotonic if it is stably computable by a CRN whose output count never decreases but unlike an output-




(a)

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

(b)

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

(c)

Figure 1.2: A simple fissure function $f : \mathbb{N}^2 \to \mathbb{N}$. On Figure 1.2a the three domains of $f$ are illustrated. There is one “fissure line” called $\mathrm{Dom}_0$, and two “wedge” domains called $\mathrm{Dom}_A$ and $\mathrm{Dom}_B$ (“A” is above and “B” is below the fissure line). The function value on each of these domains is specified in Figure 1.2b. 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 $\mathrm{Dom}_0$. In Figure 1.2c 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 $\mathrm{Dom}_0$, 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 $\mathrm{Dom}_A$ or $\mathrm{Dom}_B$ 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 $\varphi_A$ does not depend on $n_2$ (and the function $\varphi_B$ does not depend on $n_1$).

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 thesis is structured as follows. Section 2.1 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 next section and chapter we prove Theorem 1, with Section 2.2 giving a general outline that any function which is stably computable by an output-oblivious CRN obeys certain properties and Chapter 3 providing explicit constructions of CRNs. Finally in Chapter 4 an analysis of the time complexity of functions computable by output-oblivious CRNs is provided.

# Chapter 2

# Related Work

## 2.1 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.

### 2.1.1 Chemical Reaction Networks (CRNs)

CRNs specify possible behaviors 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 $1 \leq j \leq m$, $s_j \neq t_j$. 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 $\mathbf{s} \leq \mathbf{c}$ (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 $1 \leq i \leq t$. 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 work 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.

#### 2.1.2 Linear and Semilinear Sets; Lines, Grids, and Wedges

For a vector $\mathbf{v}$, let $v_i$ denote its $i$th coordinate. Let $D \subseteq \mathbb{N}^2$ and let $\Pi_1$ and $\Pi_2$ 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 semilinear set with the form $\{\mathbf{n} : \mathbf{n} \cdot \mathbf{v} \geq r\}$ (i.e., a halfspace) for some $\mathbf{v} \in \mathbb{Z}^2$ and $r \in \mathbb{Z}$ [4]. 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 lines parallel to the x or y axes; the points on these lines, if any, are in $E$). (Note that the boundary of a threshold set $\{\mathbf{n} : (n_1, n_2) \cdot (v_1, v_2) \geq r\}$ can be written as the linear set $\{(v_2/k, -v_1/k)\alpha + (|\min\{0, r/v_2\}|, \max\{0, r/v_1\}) : \alpha \in \mathbb{N}, k = \gcd(v_1, v_2)\}$ or similarly if it does not pass any points on the axes. For example, $\langle n_1, n_2 \rangle \cdot \langle -3, 2 \rangle = -1$ is the set $(2,3)\alpha + (1,0)$. If a line is infinite then $(v_2, -v_1) \geq (0,0)$.) 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 1.2 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; see Section 3.1.1 for the proof and a more formal definition of a wedge.

**Lemma 1.** *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.*

### 2.1.3 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 $\mathrm{Dom}_i$ in place of $\mathrm{Dom}(\varphi_i)$. We define semilinear functions using a characterization of Chen et al. [8]:

**Definition 1** (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 1. Lemma 2 then states that semilinear and semiaffine functions are equivalent. The proof is in Section 3.1.1.

**Definition 2** (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 2.** *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 3** (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 [13] observed that an output-oblivious function must be increasing. We hereafter focus on increasing functions.

**Definition 4** (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 $\mathbf{n} \geq \mathbf{o}$:*

$$f(\mathbf{n}) = \begin{cases} \varphi_A(\mathbf{n}), & \text{if } \varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) \leq -k, \\ \varphi_{-i}(\mathbf{n}) = \varphi_A(\mathbf{n}) - d_{-i}, & \text{if } \varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = -i, 1 \leq i < k, \\ \varphi_i(\mathbf{n}) = \varphi_B(\mathbf{n}) - d_i, & \text{if } \varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) = i, 0 \leq i < k, \\ \varphi_B(\mathbf{n}), & \text{if } \varphi_A(\mathbf{n}) - \varphi_B(\mathbf{n}) \geq k. \end{cases} \tag{2.1}$$

*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* $\mathrm{Dom}_A$ *and*

$\varphi_B < \varphi_A$ *on* $\mathrm{Dom}_B$*; 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* $p$*.*

## 2.2 A Summary of the Necessity Proof for Theorem 1

Here I summarize the necessity proof that, if a function is output-monotonic, then it is either grid-affine or the minimum of finitely many fissure functions. The proof of this direction was obtained by my co-authors with small contributions from me. For details of the proofs, see Chugg et al. [9]. First I describe two conditions on a function which ensure that it is not output-oblivious. Then I explain the general idea of the necessity proof.

### 2.2.1 Impossibility Lemmas

**Lemma 3.** *Let* $f : \mathbb{N}^2 \to \mathbb{N}$ *be a semiaffine function. Suppose that* $f = \varphi_i$ *on* $\mathrm{Dom}_i$ *and* $f = \varphi_j$ *on* $\mathrm{Dom}_j$*, where* $\mathrm{Dom}_i$ *and* $\mathrm{Dom}_j$ *lie on the same grid,* $\mathrm{Dom}_i$ *is a wedge domain and for some two-way-infinite line* $L$ *in* $\mathrm{Dom}_j$*,* $\varphi_j > \varphi_i$ *on* $L$*. Then* $f$ *cannot be stably computed by an output-monotonic CRN.*

**Lemma 4.** *Let* $f : \mathbb{N}^2 \to \mathbb{N}$ *be a semiaffine function. Suppose that* $f = \varphi_i$ *on* $\mathrm{Dom}_i$ *and* $f = \varphi_j$ *on* $\mathrm{Dom}_j$*, where* $\mathrm{Dom}_i$ *and* $\mathrm{Dom}_j$ *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* $\mathrm{Dom}_i$ *and* $\mathrm{Dom}_j$*, with* $\varphi_L < \varphi_i$ $(= \varphi_j)$ *on* $L$*. Then* $f$ *cannot be stably computed by an output-monotonic CRN.*

The general idea of these two lemmas is to find a configuration that allows all possible reaction sequences of another configuration for any same amount of remaining inputs. And then find a set of remaining inputs and a reaction sequence of the later configuration that outputs more than the answer in the former configuration. To find such two configurations it is enough to find two settings such that the count of all species in one setting is greater than the other. To find a reaction sequence that causes a problem, it is enough to limit the search of the two increasing settings to pairs that can cause problems. An infinite sequences of such pairs exists in the functions mentioned in the lemmas and by some property of infinite sequences [11] an infinite sequence of configurations has an increasing subsequence.

### 2.2.2 General Outline of the Necessity Proof

The proof analyzes each grid of the semiaffine function separately. For a large enough offset $o$ the threshold lines divide the points larger than $o$ to wedge domains and line domains sorted by threshold line slopes. The wedges and lines adjacent to each set of parallel lines can define partial fissure functions because of the two impossibility lemma. The minimum of these fissure functions is equal to $f$ on all of these domains.

## 2.3 The General Case

As mentioned in the results section, Severson et al. [19] characterized the functions with more than two inputs that can be stably computed in an output-oblivious manner. They do not use the notation of fissure functions. Instead, they express output-oblivious functions for large enough inputs as minimum of grid-affine functions (or more accurately minimum of quilt-affine functions within different infinite domains). The main idea of their approach is that with large enough periods of grid-affine functions, the periods will be larger than fissure widths and the anomalies of fissure lines can be considered as periodic characteristics of a grid-affine function. By this assumption all output-oblivious functions can be expressed as minimum of grid-affine functions where each fissure and each wedge is defined by a separate grid-affine function. For example for the fissure function in the figure 1.2 the minimum of the following increasing grid-affine functions will result in the desired fissure function,

$$
\begin{aligned}
f_1(\mathbf{n}) &= n_1 + 1, \\
f_2(\mathbf{n}) &= \begin{cases} \frac{n_1}{2} + \frac{n_2}{2}, & n_1 = n_2 \mod 2, \\ \frac{n_1}{2} + \frac{n_2}{2} + \frac{1}{2}, & n_1 \neq n_2 \mod 2, \end{cases} \\
f_3(\mathbf{n}) &= n_2 + 1.
\end{aligned}
$$

A simple CRN can be constructed for each of these functions and the minimum can be calculated with a simple operation.

For the inputs with more than two dimensions, their solution is recursive. That is by fixing any number of inputs to constants, for large enough values of the other input variables the function is minimum of grid-affine functions.

One of the advantages of this approach to the fissure function approach is that fissure functions are not as well defined as the minimum of grid-affine functions in higher dimensions. For example a higher dimensional fissure function can have a fissure of complex shape (that is the cross sections of fissures are 2D in a 3D space and can be any 2D shape) or even may have an infinite number of fissure lines in one direction (The 2D cross section of the fissure can be infinite in one direction).

# Chapter 3

# Sufficiency Condition for Stable Computability by Output-Oblivious CRNs

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 10 shows that an increasing affine function whose domain is a grid is output-oblivious. Lemma 12 shows that a partial fissure function is output-oblivious. Finally, Lemma 13 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 $f$.

## 3.1 Semiaffine Functions

Semiaffine functions as defined in the preliminaries chapter help to break down the semilinear functions with periodic structures into a set of non-periodic functions defined all over certain grids. This helps both the construction of CRNs for such functions by first constructing the non-periodic functions and then combining the CRNs to create the original semilinear function, and also helps to limit the impossibility arguments in the necessity condition to a non-periodic context.

### 3.1.1 Semilinear Functions are Semiaffine

The CRN provided in the next sections is designed for Semiaffine functions. In this section we shortly summarize that the semilinear functions are semiaffine and therefore the CRNs constructed for semiaffine functions also work for the semilinear functions. For a complete proof see [9].

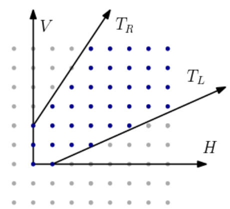



Figure 3.1: An example of a wedge domain.

**Proof of Lemma 1**

Here we prove Lemma 1. To do this, we first prove, via a sequence of lemmas, that every linear set is in fact a finite union of sets of the form $\mathcal{G} \cap \mathcal{T}$, where $\mathcal{G}$ is a grid and $\mathcal{T}$ is a finite intersection of threshold sets, where moreover the grids have the same period. We use several useful properties of semilinear sets due to Angluin et al. [4].

A *horizontal* threshold set has the form $\{\mathbf{n} : n_2 \geq r\}$ for some $r \in \mathbb{Z}$, i.e., the set includes all the points above the horizontal line $n_2 = r$. A *vertical* threshold set is defined similarly, including all the points to the right of some vertical line. The *slope* of a threshold set $\{\mathbf{n} : \mathbf{n} \cdot v \geq r\}$ is the slope of the its bounding line $\{\mathbf{n} : \mathbf{n} \cdot \mathbf{v} = r\}$, which is $-v_1/v_2$.

**Definition 5.** *Let $H$ and $V$ be horizontal and vertical threshold sets, respectively. Let $T_U$ and $T_L$ be threshold sets such that the slope of $T_U$ is strictly greater than that of $T_L$. A set $D \subset \mathbb{N}^2$ is a* wedge domain *if it can be written as $\mathcal{G} \cap T_U \cap T_L \cap H \cap V$ for some grid $\mathcal{G}$. See Figure 3.1.*

A *modulo set* is a set of the form $\{\mathbf{n} : \mathbf{n} \cdot \mathbf{v} \equiv r \mod p\}$ for some $\mathbf{v} \in \mathbb{N}^2$, and $r, p \in \mathbb{N}$. We call $p$ the *period* of the set. Recall that a *threshold set* has the form $\{\mathbf{n} : \mathbf{n} \cdot \mathbf{v} \geq r\}$ for some $\mathbf{v} \in \mathbb{Z}^2$, $r \in \mathbb{Z}$. A *boolean combination* of sets refers to a combination of sets by union and intersection. The following lemma relates semilinear sets to modulo and threshold sets.

**Lemma 5** ([4]). *Every semilinear set can be represented as a finite boolean combination of modulo sets and threshold sets.*

Next, we make a simple observation concerning the period of grids.

**Lemma 6.** *Any grid $\mathcal{G} = \{(p\alpha_1, q\alpha_2) + (o_1, o_2) : \alpha_i \in \mathbb{N}\}$ can be written as a finite union of grids with the same period.*

*Proof.* Let $k$ be a multiple of both $p$ and $q$. For all $i, j \in \mathbb{N}$ such that $pi < k$

and $qi < k$, let $\mathbf{o}_{i,j} = (o_1 + pi, o_2 + qi)$. It is then easily verified that

$$\mathcal{G} = \bigcup_{i,j} \mathcal{G}_k + \mathbf{o}_{i,j}. \qquad \square$$

The next lemma demonstrates that modulo sets are effectively unions of grids in hiding. For example, the modulo set $M = \{\mathbf{n} : 2n_1 + x_2 \equiv 0 \mod 3\}$ can be written as $\mathcal{G}_3 \cup \mathcal{G}_3 + (1,1) \cup \mathcal{G}_3 + (2,2)$.

**Lemma 7** ([9]). *Any modulo set can be expressed as the finite union of grids.*

Since the representation of a semilinear set $E$ may also have intersections of modulo sets (according to Lemma 5) we must be able to write the intersection of modulo sets as grids as well. We do this by first converting the modulo sets into grids as per Lemma 7 and then reasoning about the intersection of grids. The next lemma allows us to write the intersection of these grids as the union of other grids. As an example, consider $A = \{(\alpha_1, 3\alpha_2) : \alpha_i \in \mathbb{N}\}$ and $B = \mathcal{G}_2 + (1,2)$. Here, $A \cap B$ is the grid $\{(2\alpha_1, 6\alpha_2) + (1,0) : \alpha_i \in \mathbb{N}\}$.

**Lemma 8** ([9]). *The intersection of two grids can be expressed as the finite union of grids.*

Finally, using distributivity of set operations can prove the main result.

**Lemma 9.** *Every semilinear set can be written as a finite union of sets of the form $\mathcal{G} \cap \mathcal{T}$, where $\mathcal{G}$ is a grid and $\mathcal{T}$ is a finite intersection of threshold sets. Moreover, we may assume that each grid has the same period.*

*Proof.* Let $L$ be semilinear. Using Lemma 5 express $L$ as a finite boolean combination of threshold and modulo sets. Expressing each modulo set as a finite union of grids according to Lemma, 7 and then using distributivity of set operations and Lemma 8 we can write $L = \bigcup_{j=1}^{N} \mathcal{G}^j \cap \mathcal{T}^j$ for some $N \in \mathbb{N}$ where each $\mathcal{T}^j$ is a finite *intersection* of threshold sets and $\mathcal{G}^j$ is a grid. Finally, applying Lemma 6, we may assume that each $\mathcal{G}^j$ has the same period. $\square$

It now remains only to remark that we have in fact proven Lemma 1.

*Proof of Lemma 1.* Let $L$ be semilinear and using the previous lemma write $L = \bigcup_{j=1}^{N} \mathcal{G}^j \cap \mathcal{T}^j$. Fix $M = \mathcal{G}^j \cap \mathcal{T}^j$. If $M$ is finite, then it can be written as the finite union of points. If $\mathcal{T}^j$ includes only the boundary of some threshold set, then $M$ is a line on the grid $\mathcal{G}^j$. Otherwise, $M$ is infinite and is not a line. It follows that $M$ can be written as the union of a wedge domain and finitely many points. □

*Proof of Lemma 2.* From Definition 1 (Chen et al.), $f$ can be represented by the partial functions $\{\varphi_1, \ldots, \varphi_m\}$ where $\text{Dom}_i$ is linear for each $i$. Applying Lemma 5 and Lemma 6, we can write $\text{Dom}_i$ as the union of semiaffine sets with the same period, say $p$. Therefore, each grid of period $p$ is covered by a distinct function $f_i$ which is defined by the union of all affine functions whose domain covers this grid. Thus, it follows that $f$ is a periodic combination of partial semiaffine functions on grids. □

## 3.2 Construction of Fissure Functions

**Lemma 10.** *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 $f(\mathbf{n}) = a_1n_1 + a_2n_2 + a_0$, 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 $Y$:

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

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 12 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 4. As noted in that definition, it must be that $A_1 > B_1$ and $B_2 > A_2$, since $\varphi_A < \varphi_B$ on $\text{Dom}_A$ 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 $[k, \ldots, K-1]$, 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 $[-K+1, \ldots, -k]$ 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 $\mathrm{Dom}_A$ and $\mathrm{Dom}_B$ are proper subsets of $\mathcal{G}$.

**Lemma 11.** *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 $\mathrm{Dom}_B$ then $M^{-1}(z_1', z_2)$ is also in $\mathrm{Dom}_B$. Similarly if $z_2 \leq z_2'$ and $M^{-1}(z_1, z_2)$ is in $\mathrm{Dom}_A$ then $M^{-1}(z_1, z_2')$ is also in $\mathrm{Dom}_A$.*

**Lemma 12.** *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 11. 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 $(\varphi_A(0,0), \varphi_B(0,0)) = (A_0, B_0)$:

$$L \to L' + A_0 Z_1 + B_0 Z_2.$$

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:

$$\begin{aligned} [l_\mathbf{x}, *, *] + X_1 &\to [l_\mathbf{x} + (A_1 - B_1) \mod 2K-1, *, *] + A_1 Z_1 + B_1 Z_2 \\ [l_\mathbf{x}, *, *] + X_2 &\to [l_\mathbf{x} + (A_2 - B_2) \mod 2K-1, *, *] + A_2 Z_1 + B_2 Z_2 \end{aligned}$$

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 $\mathbf{n} = (0,0)$:

$$L' + A_0 Z_1 + B_0 Z_2 \to [A_0 - B_0, A_0 - B_0, \min\{A_0, B_0\} - d_{A_0 - B_0}].$$

$Z$-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.

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

where

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

The deficit $d$ can never exceed $2d_{\max} + 1$ since the reactions are only applicable when $d \leq d_{\max}$ 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 $Y$s 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 $_z$ 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 $Z$s 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 $\mathrm{Dom}_B$, 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:

$$[l_{\mathbf{x}}, l_{\mathbf{z}}, d] \to [l_{\mathbf{x}}, l_{\mathbf{z}}, 0] + dY \text{ if } -k < l_{\mathbf{x}} = l_{\mathbf{z}} < k. \tag{3.1}$$

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:

$$\begin{aligned}[l_{\mathbf{x}}, l_{\mathbf{z}}, d] + r_1 Z_1 \to [l_{\mathbf{x}}, l_{\mathbf{z}}, 0] + r_1 Z_1 + dY,&\\ l_{\mathbf{z}} \geq k, l_{\mathbf{z}} + r_1 = l_{\mathbf{x}} \mod 2K - 1, d > 0.&\end{aligned} \tag{3.2}$$

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:

$$\begin{aligned}[l_{\mathbf{x}}, l_{\mathbf{z}}, d] + r_2 Z_2 \to [l_{\mathbf{x}}, l_{\mathbf{z}}, 0] + r_2 Z_2 + dY,&\\ l_{\mathbf{z}} \leq -k, l_{\mathbf{z}} + r_2 = l_{\mathbf{x}} \mod 2K - 1, d > 0.&\end{aligned} \tag{3.3}$$

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.*

### 3.2.1 Proof of Claim 3

*Proof.* Let $\mathbf{z}_p$ be the number of $(Z_1, Z_2)$ already consumed at the last time that the deficit is cleared before $Z$-consumption stalls, plus the number of $Z_1$ or $Z_2$ that are reactants of this last deficit clearing reaction, if any (i.e., $r_1$ if the reaction is (3.2) or $r_2$ if the reaction is (3.3)). Let $\mathbf{n}_p = (n_{1p}, n_{2p})$ be the counts of the inputs $(X_1, X_2)$ that have been consumed at this time. The deficit is nonnegative when $Z$-consumption stalls if $f(M^{-1}(\mathbf{z}_s)) \geq f(M^{-}(\mathbf{z}_p))$. We will show that $f(\mathbf{n}_p) \geq f(M^{-1}(\mathbf{z}_p))$. Since $f(M^{-1}(\mathbf{z}_s)) = f(\mathbf{n})$ by Claim 2, since $\mathbf{n} \geq \mathbf{n}_p$ and since $f$ is increasing on integer-valued points, we then have

$$f(M^{-1}(\mathbf{z}_s)) = f(\mathbf{n}) \geq f(\mathbf{n}_p) \geq f(M^{-}(\mathbf{z}_p)).$$

To show that $f(\mathbf{n}_p) \geq f(M^{-1}(\mathbf{z}_p))$, first suppose that $\mathbf{n}_p$ is on a fissure line. Note that $\mathbf{n}_p$ and $M^{-1}(\mathbf{z}_p)$ must be on the same fissure line, since deficit-clearing reactions can happen only in this case. Now suppose that the deficit-clearing reaction applied is (3.1), i.e., $\mathbf{x} = \mathbf{n}_p$, $\mathbf{z} = M^{-1}(\mathbf{z}_p)$ and both $\mathbf{x}$ and $\mathbf{z}$ are on the same fissure line. So

$$f(\mathbf{n}_p) = \min(\varphi_A(\mathbf{n}_p), \varphi_B(\mathbf{n}_p)) - d_{l_{\mathbf{x}}} \geq \min\{z_{1p}, z_{2p}\} - d_{l_{\mathbf{x}}} = f(M^{-1}(\mathbf{z}_p)).$$

Next suppose that the deficit-clearing reaction applied is (3.2). Now, the line $l$ containing $M^{-1}(\mathbf{z}_p)$ must be such that $l \geq K$; otherwise the condition that $l = l_{\mathbf{x}} \mod K$ would not hold. Then by our choice of $K$, which is at least $k + d_{\max}$, it must be that

$$(\varphi_A(\mathbf{n}_p), \varphi_B(\mathbf{n}_p)) \geq (z_{1p}, z_{2p} + d_{\max}).$$

Intuitively, this is because to "get back" to $\mathbf{n}_p$ from $M^{-1}(\mathbf{z}_p)$ requires consuming at least $d_{\max}$ more $Z_2$s. Also $z_{1p} - z_{2p} = l > K$, so

$$\varphi_A(\mathbf{n}_p) \geq z_{1p} \geq z_{2p} + K \geq \min\{z_{1p}, z_{2p}\} + (k + d_{\max}),$$

and so

$$\min\{\varphi_A(\mathbf{n}_p), \varphi_B(\mathbf{n}_p)\} \geq \min\{z_{1p}, z_{2p}\} + d_{\max}.$$

Therefore

$$f(\mathbf{n}_p) = \min\{\varphi_A(\mathbf{n}_p), \varphi_B(\mathbf{n}_p)\} - d_l \geq \min\{z_{1p}, z_{2p}\} + d_{\max} - d_l$$
$$\geq \min\{z_{1p}, z_{2p}\} = f(M^{-1}(\mathbf{z}_p).$$

Otherwise $\mathbf{n}_p$ is not on a fissure line (although $l_{\mathbf{x}}$ might be the index of a fissure line). In this case $f(\mathbf{n}_p) = \min(\varphi_A(\mathbf{n}_p), \varphi_B(\mathbf{n}_p))$. Then, regardless of which deficit-clearing reaction is applied, we have that

$$f(\mathbf{n}_p) = \min(\varphi_A(\mathbf{n}_p), \varphi_B(\mathbf{n}_p)) \geq \min\{z_{1p}, z_{2p}\} \geq f(M^{-1}(\mathbf{z}_p)).$$

Thus in every case we have that $f(\mathbf{n}_p) \geq f(M^{-1}(\mathbf{z}_p)$, and we are done. □

To complete the proof of Lemma 12, 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 (3.1) is applicable. If $M^{-1}(\mathbf{z}_s)$ is on a boundary line then either (3.2) or (3.3) 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 $f(\mathbf{n})$. □

## 3.3 Stitching Lemma

**Lemma 13.** *(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.* 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 $\text{Dom}_j$ of $f_j$ is a two-way-infinite grid for all $j, 1 \leq j \leq m$. Let the offset of the $j$th 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 explained after this.

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. For simplicity of notation, fix some $j$, let $\text{Dom}' = \text{Dom}_j$, let the base vectors of $\text{Dom}'$ be $(a'_1, 0)$ and $(0, a'_2)$, and let $\mathbf{o}' = (o'_1, o'_2)$ be its offset. Let $\mathbf{n}^{(\mathbf{j})} = \mathbf{n}' = (n'_1, n'_2)$ be the smallest element of $\text{Dom}_j$ such that $\mathbf{n} \leq \mathbf{n}'$. The following CRN $C'$, which has nine reactions in total, produces $n'_1$ copies of species $X'_1$ and $n'_2$ copies of species $X'_2$ from a leader $L'$:

$$\begin{array}{lll} L' & \to L'_{00} + o_1 X'_1 + o_2 X'_2 & \\ L'_{0b} + (o'_1+1)X_1 & \to L'_{1b} + a'_1 X'_1, & \text{for } b=0 \text{ and } b=1 \\ L'_{b0} + (o'_2+1)X_2 & \to L'_{b1} + a'_2 X'_2, & \text{for } b=0 \text{ and } b=1 \\ L'_{1b} + a'_1 X_1 & \to L'_{1b} + a'_1 X'_1, & \text{for } b=0 \text{ and } b=1 \\ L'_{b1} + a'_2 X_2 & \to L'_{b1} + a'_2 X'_2, & \text{for } b=0 \text{ and } b=1 \end{array}$$

To produce all inputs $\mathbf{n^{(1)}}, \ldots, \mathbf{n^{(m)}}$, $m$ copies of CRN $C'$ are needed, each with independent copies of the species; for example, species $X_1^{(j)}$ and $X_2^{(j)}$ are substituted for $X'_1$ and $X'_2$. Three additional reactions produce the leaders and input copies needed for each of these $m$ independent copies of $C'$:

$$\begin{array}{ll} L & \to L'_1 + \ldots L'_m \\ X_1 & \to X_1^{(1)} + \ldots X_1^{(m)} \\ X_2 & \to X_2^{(1)} + \ldots X_2^{(m)}. \end{array}$$

**Producing the Output $y$:** Let $Y_j$ be the output species of the output-oblivious CRN for $f_j$, so that its count $y_j = f_j(\mathbf{n^{(j)}})$. To produce $y = \min\{y_1, \ldots, y_m\}$ we need just one reaction:

$$Y_1 + \ldots + Y_m \to Y.$$

This completes the contruction for the case that all $\mathrm{Dom}_j$ are 2D grids.

We now modify CRN $C$ with additional reactions to handle the case that some of the $\mathrm{Dom}_j$ may be 0D grids (i.e., points) or 1D grids. The problem that can arise is that for some $j$, $\mathrm{Dom}' = \mathrm{Dom}_j$ may not have any point that is greater than input $\mathbf{n}$. We add reactions that generate a species $S'$ when this is the case. Specifically, if $\mathrm{Dom}'$ contains no base vector of the form $(a'_1, 0)$, we add

$$L'_{1b} \to S', \text{ for } b=0 \text{ and } b=1,$$

and we remove reactions above that involve $a'_1$. Similarly, if $\mathrm{Dom}'$ contains no base vector of the form $(0, a'_2)$, we add

$$L'_{b1} \to S', \text{ for } b=0 \text{ and } b=1.$$

and we remove reactions above that involve $a'_2$. Note that $S'$ is produced if and only if $\mathrm{Dom}'$ has no point $\mathbf{n'}$ such that $\mathbf{n} \le \mathbf{n'}$.

As before, $C$ contains distinct copies of these reactions for each $\mathrm{Dom}_j$ that is not 2D, producing one or two copies of species $S_j$ if and only if $\mathrm{Dom}_j$ has no point $\mathbf{n^{(j)}}$ such that $\mathbf{n} \le \mathbf{n^{(j)}}$.

Finally, we need to ensure that the ouput $y$ produced is the min of the $y_j$ taken over domains $\mathrm{Dom}_j$ for which $\mathbf{n^{(j)}} \in \mathrm{Dom}_j$. To do this, we replace the single reaction $Y_1 + \ldots + Y_m \to Y$ by $2^m$ new reactions, one for each subset $I$ of $\{1, \ldots, m\}$. The reaction corresponding to subset $I$ is of the form

$$Y_1' + \ldots + Y_m' \to Y + Y_1'' + \ldots + Y_m'',$$

where $Y_j' = Y_j$ if $j \in I$ and $Y_j' = S_j$ otherwise, and $Y_j''$ is some inactive species if $j \in I$ and $Y_j'' = S_j$ otherwise.

□

## 3.4 Final Condition

Finally we prove 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 3, $f$ 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 10, $f_i : \mathcal{G}_i \to \mathbb{N}$ is output-oblivious, $1 \leq i \leq m$. By Lemma 13, $f$ is output-oblivious.

Otherwise $f$ is the min of finitely many complete fissure functions $f_1, \ldots, f_m$. By Definition 4, each $f_i$ is a combination of partial fissure functions on grids of period $p$, for some $p \in \mathbb{N}$. By Lemma 12, each of these partial fissure functions is output-oblivious. By Lemma 13, each $f_i$ is output-oblivious and also $f$ is output-oblivious. □

## 3.5 Examples

### 3.5.1 Example for Claim 3

In Figure 3.2 a contour plot for a fissure function is provided, this fissure function is used in the next example which is displayed in Figure 3.3. The fissure function in these two figures is defined as follows.

$$f(\mathbf{n}) = \begin{cases} \varphi_A(\mathbf{n}) = 3n_1 + 2n_2 + 6, & \text{if } \mathbf{n} \in \mathrm{Dom}_A \text{ i.e. } \mathbf{n} \in L_i, i \leq -2, \\ \varphi_A(\mathbf{n}) - d_{-1} = 3n_1 + 2n_2 + 3, & \text{if } \mathbf{n} \in L_{-1}, \\ \varphi_A(\mathbf{n}) - d_0 = 3n_1 + 2n_2, & \text{if } \mathbf{n} \in L_0, \\ \varphi_B(\mathbf{n}) - d_1 = 2n_1 + 3n_2 + 1, & \text{if } \mathbf{n} \in L_1, \\ \varphi_B(\mathbf{n}) - d_2 = 2n_1 + 3n_2 + 3, & \text{if } \mathbf{n} \in L_2, \\ \varphi_B(\mathbf{n}) - d_3 = 2n_1 + 3n_2 + 3, & \text{if } \mathbf{n} \in L_3, \\ \varphi_B(\mathbf{n}) = 2n_1 + 3n_2 + 6, & \text{if } \mathbf{n} \in \mathrm{Dom}_B \text{ i.e. } \mathbf{n} \in L_i, 4 \leq i. \end{cases} \tag{3.4}$$

In some of the proofs and for the definition of lower and upper boundary lines, $L_{-2}$ and $L_{-3}$ are also considered as fissure lines with $d_i = 0$ instead of being in $\mathrm{Dom}_A$.
As displayed in purple in Figure 3.2, if the CRN returns from an state on a line far enough from the fissure lines to a fissure line the deficit change is going to be non negative. This is because $\min\{\varphi_A(n), \varphi_B(n)\} - f(n)$ is at most a constant and by choosing a far enough initial line $\min\{\varphi_A(n), \varphi_B(n)\}$ can increase arbitrary large in the return and more than this constant value, in turn this is because moving from two adjacent lines in the direction of fissure lines increases the min function exactly by one. Therefore $f(n)$ should not decrease. Because of this, in the case that the deficit-clearing reaction happens far from fissure lines, it is not going to encounter a problem. The next figure will argue that $f$ is not decreasing in other complex scenarios for deficit-clearing reactions.

In Figure 3.3 an example is provided for proof of Claim 3. In this example the same fissure function as in the previous figure is used. Note that in the first scenario of Figure 3.3 the deficit is negative at $M^{-1}(z) = (3, 0)$ with $f(3, 0) = 9$ because the deficit is cleared at $M^{-1}(z_p)$ where $f(M^{-1}(z_p)) = 10$. In case of a negative deficit we know that the CRN does not need to output immediately because we already know that $f(n)$ is larger than the current output.

In Figure 3.3 in both scenarios based on $l_z = 1$ we know $M^{-1}(z_p)$ lies on the line $L_1$ or one of the far lines. In scenario 1 because $l_x = l_z = 1$, the deficit clearing reaction is allowed. In such a scenario if $n_p$ and $M^{-1}(z_p)$ both lie on the fissure line, because $M(n_p) > z_p$ (the green area) we know $n_p$ is in the intersection of the line $L_1$ and the green area. All points in this intersection are greater than $M^{-1}(z_p)$ (the medium darkness area) and therefore $f(n_p) > f(M^{-1}(z_p))$ and deficit clearing reaction does not cause a problem. Note that the case that $n_p$ or $M^{-1}(z_p)$ are not on $L_1$ is explained in the previous paragraphs and the other figure. In the second scenario, although $f(n_p) = 9 < f(M^{-1}(z_p)) = 10$ and a deficit clearing reaction would produce more outputs than the answer, because the condition $l_x = l_z$ does not hold the deficit clearing reaction is not allowed.

### 3.5.2 Example for Lemma 9

Figure 3.4 shows an example for the Lemma 9. Here the conversion of domains is only shown for a linear set.

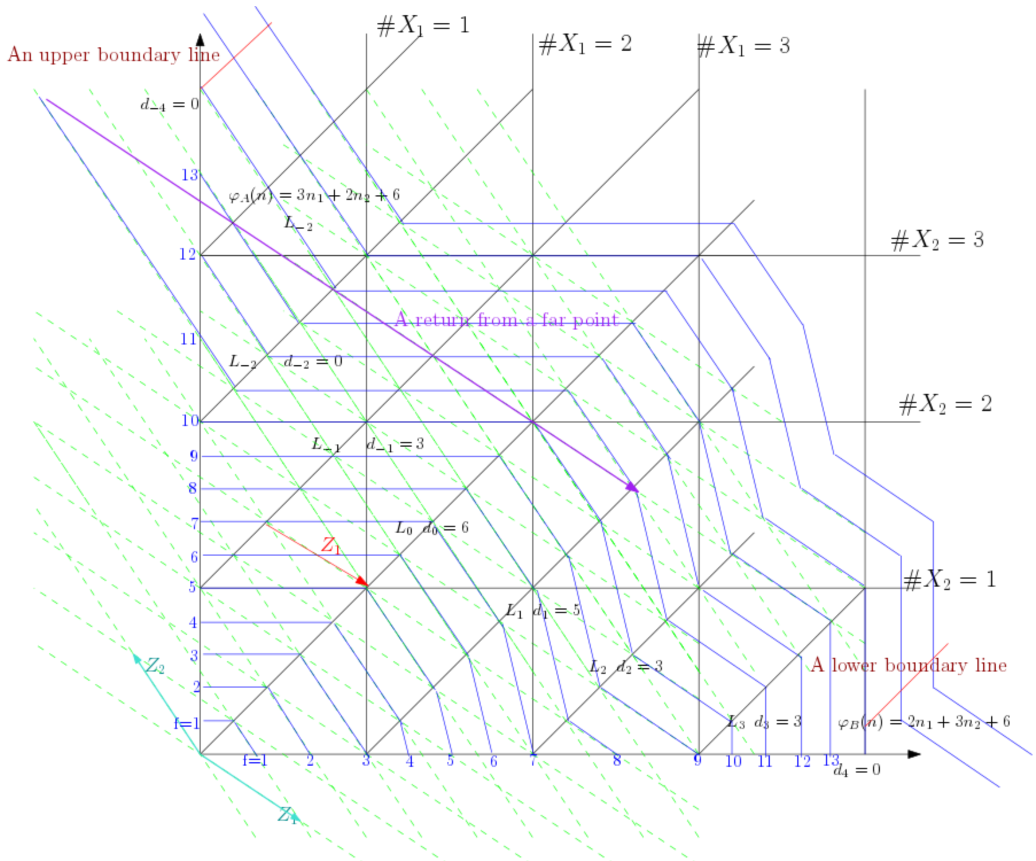



Figure 3.2: A contour plot for the fissure function used in the next example and defined in Equation 3.4. The red arrow shows a transition with negative deficit change. The function can not take back its output so it should not cause negative deficits which last until after when the execution is finished. The purple arrow displays a return to the fissure lines from a far line. If the initial line is far enough from the fissure lines, the change in the deficit is going to be positive or non-negative similar to the purple arrow.

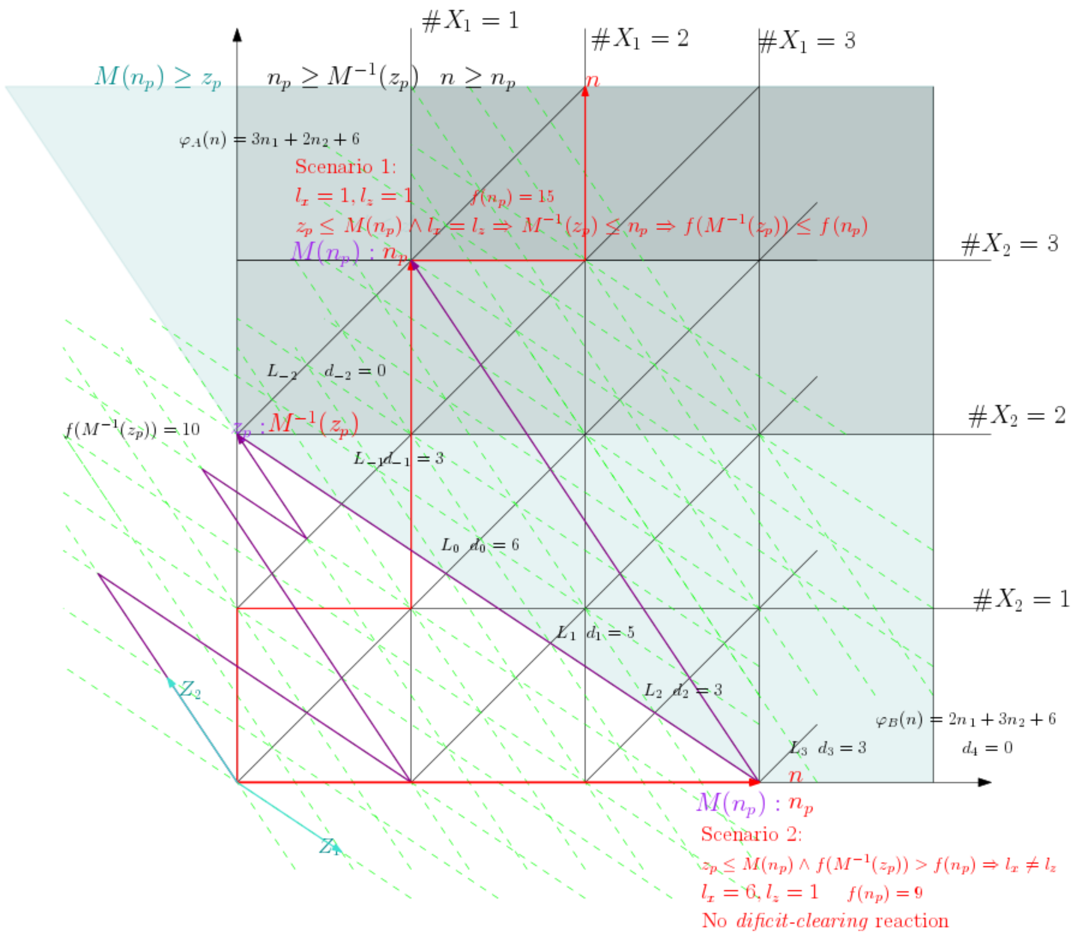



Figure 3.3: Similar to the proof of Claim 3, assume that $z_p$ is the number of $(Z_1, Z_2)$ already consumed and $n_p$ is the number of $(X_1, X_2)$. Two scenarios with the same path for $(Z_1, Z_2)$ consumption (purple) but different paths for $(X_1, X_2)$ consumption (red) are provided.

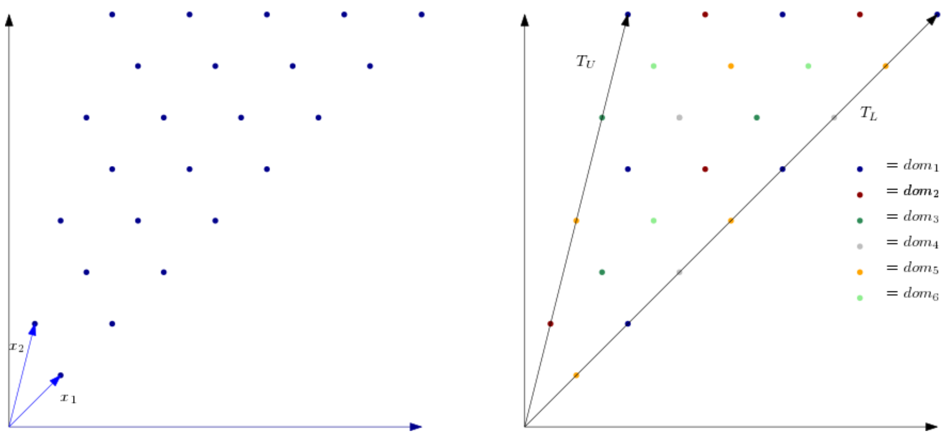



Figure 3.4: Based on Lemma 9, each semilinear domain can be expressed as the union of sets like $\mathcal{G}_i \cap \mathcal{T}_i$, where $\mathcal{G}_i$ is a grid with a period $p$ and $\mathcal{T}_i$ is an intersection of threshold sets and where all grids share the same period. In this example all $\mathcal{T}_i$ are the same but they can be different in the general case. The left figure shows a linear set with the base $\{x_1, x_2\}$ where $x_1 = (2, 2)$ and $x_2 = (1, 4)$. The right figure shows the same set expressed as union of $\mathcal{G}_i \cap \mathcal{T}$. Each grid in the right figure has a period of 6 and there are a total of 6 grids and $\mathcal{T}$ is the intersection of two threshold sets defined by $x \cdot (-2, 2) \geq 0$ and $x \cdot (4, -1) \geq 0$.

# Chapter 4

# Time Analysis

## 4.1 The Kinetic Model and Basics

The stochastic model of chemical kinetics describes interactions involving small number of molecules [14, 20]. This subsection is structured similar to David Doty's lecture notes [12]. In this analysis we assume that the probability of a reaction taking place follows an exponential distribution with expected value of $\frac{1}{\lambda}$.
The rate $\lambda$ depends on volume $v$ and the number of available reactant species.
Here the following formula is used for the rate $\lambda$ of a given reaction with reactants $S_1, ..., S_d$. This formula is with respect to a given configuration, in which the count of species $S_i$ is $\#S_i$.

$$\lambda = k.\frac{1}{v^{\|r\|-1}} \prod_{i=1}^{d} \frac{\#S_i!}{(\#S_i - r(S_i))!},$$

where $||r||$ is the count of the molecules consumed in the reaction and $r(S_i)$ is the count of the molecules of the $i$th species consumed in the reaction and $k$ is a constant independent of species count. We will assume that $k = 1$ to simplify the expressions.
Here the time complexity refers to the expected amount of time needed for the CRN to stabilize. In the examples this amount of time is shown by $T$ and the expected amount is shown by $E[T]$.
For example the rate for the reaction $\alpha : A \to B$ with $\#A = n$ is $\lambda = n$ and the time complexity is $\frac{1}{\lambda} = \frac{1}{n}$. This rate for the reaction $\alpha : A + A \to B$ is $\lambda = \frac{n(n-1)}{v}$.
A series of scenarios are analyzed by other authors. The following settings and their time complexities are included here.

1. $\#A = n$ initially in volume $n$:

$$\alpha : A \to B$$

The time for each reaction is independent of the previous ones. So,

$$E[T] = \sum E[T_i] = \sum \frac{1}{i} = \Theta(\ln n),$$

where $T_i$ is the amount of time needed for the reaction when $\#A = i$.

2. $\#A = \#B = 1$ initially in volume $n$:

$$\alpha : A + B \to C.$$

Only one reaction occurs with $\lambda = \frac{1*1}{n}$. So,

$$E[T] = n.$$

3. $\#A = \#B = n$ initially in volume $n$:

$$\alpha : A + B \to C.$$

The last reaction needs $\Theta(n)$ time, the rest of reactions are also within $\Theta(n)$. That is the final expected time is also $\Theta(n)$ as follows,

$$E[T] = \sum E[T_i] = \sum \frac{n}{i * i} = \Theta(n),$$

where $T_i$ is defined similar to scenario 1.

4. Minimum. in case of minimum with two species and $\#A = \#B$ this is the same as the pairing off reactions in scenario 3 and has an expected time of $\Theta(n)$. When $\#A > \#B$ the time of individual reactions with the same amount of $\#B$ decreases and therefore the time is $O(n)$. When the difference of $\#A$ and $\#B$ is large the time can be below $n$. For example when $\#A \geq 2 \times \#B$,

$$E[T] = \sum E[T_i] \leq \sum \frac{n}{(\#A - \#B)i} \leq \frac{n}{\frac{n}{3}} \sum \frac{1}{i} = O(\lg n).$$

Other cases similarly can be analyzed. Later we will show similar results for case-specific scenarios.

## 4.2 Output Oblivious CRN Time Analysis

### 4.2.1 Basics with Leader

#### Reaction with Leader

Assume $\#A = n$ and $\#L = 1$ initially in volume $n$.

$$\alpha : A + L \to B + L.$$

Similar to the leaderless scenario, if $T_i$ is the time needed for the reaction when count of $A$ is $i$, then $T_i$ are independent.

$$E[T] = \sum E[T_i] = \sum \frac{n}{i * 1} = \Theta(n \lg n).$$

#### Minimum with Multiple Species

Assume $\#A_1 = a_1$,$\#A_2 = a_2$,...,$\#A_k = a_k$ and $\#L = 1$ initially in volume $n$. Here we assume that no other reactions have the $A_i$ as reactants. We can rewrite the following reaction,

$$\alpha : A_1 + A_2 + ... + A_k + L \to B + L,$$

as the set of reactions,

$$L \to L_{(00..0)}$$

$$L_{(b_1..b_i=0..b_k)} + A_i \to L_{(b_1..b_i=1..b_k)}$$

$$L_{(11..1)} \to B + L_{(00..0)}.$$

This way we only use reactions that have two reactants and are faster.
In case that there are other reactions that want to use the species temporarily absorbed by the leader, it is still possible to construct a CRN with 2 reactants. In this case the other reactions also should be able to use the absorbed species. For example in the subsection allocated to fissure function time analysis two reactions use the same species and both are replaced by reactions with input size of 2.
The time $T_{ij}$ that one instance of the $i$th species with count $j$ is consumed by the leader when it is consumable by the leader and the time $T'_i$ that leader outputs the $i$th instance of $B$ when possible are all independent from each other. Although more than one reaction might be possible at a time, and this gives a final time which is less than the sum of the reaction times, we

know that at least one reaction is possible at a time so the sum of individual reaction times give an upper bound for the total time needed by CRN.

$$E[T] \leq \sum_{i,j} E[T_{ij}] + \sum_{i} E[T'_i] = (\sum_{i,j} \frac{n}{j}) + \min(a_1, .., a_k)$$

$$\leq (\sum_{i} n \log(\min(a_1, .., a_k))) + \min(a_1, .., a_k) = O(n \lg n).$$

The bound is tight when $\min(a_1, .., a_k)$ is $\Omega(n)$.
Although it is possible to construct CRNs that do not have the requirement that no other reactions should use the reactants of the replaced reaction, the stitching lemma's minimum operations meets this requirement and therefore no other reaction than the minimum reaction needs to be changed.

Comparing to the original minimum reaction with $k$ species (reaction $\alpha$), in case when $\#A_1 = \#A_2 = ... = \#A_k = n$ the reaction $\alpha$ should happen $n$ times and the reaction with $\#A_1 = i$ has a time of $\frac{n^k}{i^k}$, so considering only the last reaction the total time is $\Omega(n^k)$ (or $n^{k-1}$ if the reaction was leaderless). In a similar way to the pairing off reaction that is the scenario 3 of the previous section, it can be shown that the rest of time is $O(n^k)$ and the total time is $\Theta(n^k)$ because $\sum \frac{1}{i^k}$ is constant.

### 4.2.2 Output Oblivious CRN

**Affine Function**

Similar to the minimum with multiple species, this function's reactions can be rewritten to only take one instance of a single species at a time.
There were three reactions in the CRN of an affine function. Each reaction was in the form of,

$$L + aX_1 + bX_2 \rightarrow L' + cY.$$

The corresponding two reactant reactions can be written as follows,

$$L \rightarrow L_{(0,0)}$$

$$L_{(i,j)} + X_1 \rightarrow L_{(i+1,j)} \qquad \text{for } i < a$$

$$L_{(i,j)} + X_2 \rightarrow L_{(i,j+1)} \qquad \text{for } j < b$$

$$L_{(a,b)} \rightarrow L' + cY$$

Because more than one reaction is possible at a time, the other reactions should be able to consume absorbed reactants.

**Fissure Function**

Assume that all $Z$-producing reactions happen before all $Z$-consuming and $Y$-producing reactions. This gives an upper bound for time. This is because although $Z$-producing reactions change the leader-state, $Z$-consuming reactions do not depend on the leader state $l_x$. When all $X$ are consumed the largest possible upper bound for amount of time for $Z$ consumption is when no $Z$ is consumed. We will see this upper bound later.
For the $Z$-producing reactions, they are analyzed in the reactions with leader subsection and they are $\Theta(n \lg n)$. Because there are two of these reactions we know that the maximum of the time needed is less than the sum of both times which again has an expectation of $\Theta(n \lg n)$.
For the $Z$-consuming reactions, this is similar to the two previous subsections, but as another example I will write this in more detail.
Assume that there are $z_i$ copies of $Z_i$ when $X_i$ are completely consumed. Let $T_{ij}$ be the time spent by the CRN to consume an instance of $Z_i$ when there is $j$ instance remaining and the reaction is possible. The quantities $T_{ij}$ are independent of each other. The total time can be less than the sum of all $T_{ij}$ when more than one reaction is possible at a time, because always at least one of the $Z$-consuming reactions are possible the sum gives a valid upper bound.

$$E[T_{Z\text{-consuming}}] \leq \sum_{i,j} E[T_{ij}] = (\sum_{i,j} \frac{n}{j}) \approx (\sum_{i} n \log(z_i)) = O(n \lg n).$$

The upper bound is maximum when all $Z$-producing reactions happen before all $Z$-consuming reactions as stated before.
For the final $Y$-producing reaction, it may be a $O(1)$ $Y$-producing reaction or may consume more than one reactant other than the leader. In the second case (that is Equation 3.2 and Equation 3.3) because different reactions consume the same reactants with different amount, the trick in the minimum with multiple species subsection should be modified. Instead of the following reactions,

$$L + i \times A \to L' + i \times A + B$$

$$L + j \times A \to L' + j \times A + B',$$

where $i \times A$ means $i$ instance of $A$ and $j < i$. We can write,

$$L \to L_0$$

$$L_k + A \to L_{k+1} \qquad \text{for } k \leq i$$

$$L_k \to L'_k + B \qquad \text{for } k \geq i$$
$$L_k \to L'_k + B' \qquad \text{for } k \geq j,$$

and for all other reactions involving $L$, similar reactions with $L_k$ should be added which releases $k$ instances of $A$. Applying this for $Y$-producing reactions will not affect the execution. The time analysis when all $Z$-consuming reactions are finished is as follows, the time to consume $r_1$ instance of $Z_1$ or $r_2$ instance of $Z_2$ is $O(n \lg n)$ and final $Y$-producing reaction is $O(1)$. All set of reactions are either $O(n \lg n)$ or $\Theta(n \lg n)$, so the total time of the CRN is $\Theta(n \lg n)$.

**Stitching Lemma**

As discussed before for the minimum operation in the stitching lemma the $O(n \lg n)$ minimum operation constructed in the previous sections can be used. For the rest of reactions we can assume that all the other reactions should happen before any minimum reaction. This gives an upper bound of $O(n \lg n)$ for the time.
All other reactions are analyzed in the reactions with leader subsection. Although a few of the reactions take more than one reactant except the leader similar to the minimum with multiple species subsection the reactions can be rewritten to take only one reactant other than the leader at a time. This is because different reactions don't use the same reactants at the same time. The preparation reactions are $O(n \lg n)$ and the minimum operation is $O(n \lg n)$, so if the individual functions are $O(n \lg n)$ the stitched CRN is $O(n \lg n)$. This bound is tight because many of reactions with leader can take $\Omega(n \log n)$ time alone. (For example the preparation reactions are reactions with leaders which were analyzed before, and are $\Theta(n \lg n)$).

**Final Time Complexity**

All the fissure functions and affine functions can be constructed to work in $O(n \lg n)$. These functions can be stitched in an operation that keeps the time complexity below $O(n \lg n)$. This bound is tight as mentioned in the previous section.
All this analysis is correct as long as the volume is $O(n)$. It is easy to verify that a volume of $O(n)$ is enough for the computation. Both of the CRNs for the construction of fissure functions and the stitching lemma are functional with a volume of $O(n)$ for an input of size $O(n)$.

# Bibliography


[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.

[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 $\lambda$-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, Dec 2014.

[9] Ben Chugg, Hooman Hashemi, and Anne Condon. Output-oblivious stochastic chemical reaction networks. In *Conference on Principles of Distributed Systems*, 2018.

[10] Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui, and Eric Ruppert. Secretive birds: Privacy in population protocols. In E. Tovar, P. Tsigas, and H. Fouchal, editors, *International Conference On Principles Of Distributed Systems*, pages 329–342, 2007. Lecture Notes in Computer Science, vol 4878. Springer.

[11] 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.

[12] David Doty. David doty's ecs 289 lecture notes. pages 25–27. `https://web.cs.ucdavis.edu/~doty/winter2016ecs289/notes.pdf`.

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

[14] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. In *Journal of Physical Chemistry*, 1977.

[15] 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.

[16] 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.

[17] 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.

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

[19] Eric E. Severson, David Haley, and David Doty. Composable computation in discrete chemical reaction networks. *https://arxiv.org/pdf/1903.02637.pdf*, 2019.

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

[21] 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.

[22] Chris Thachuk and Anne Condon. Space and energy efficient computation with DNA strand displacement systems. In D. Stefanovic and A. Turberfield, editors, *International Workshop on DNA-Based Computers*, pages 135–149, 2012. Lecture Notes in Computer Science, vol 7433. Springer.

[23] 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.

# Appendix: Proofs of Lemmas

## A.1 Proof of Lemma 11

*Proof.* Let $\varphi(\mathbf{n}) = \langle \mathbf{a}, n \rangle + a_0$ and $\mathcal{G} = \{(x_1\alpha_1, x_2\alpha_2) + \mathbf{o} : \alpha_i \in \mathbb{N}\}$. Let $\mathbf{n}' \in \mathcal{G} \setminus \mathcal{T}$. Assuming that $\mathbf{n}' \geq \mathbf{m}$, we can find some $\mathbf{n} \in D$ with $\mathbf{n} \leq \mathbf{n}'$ which shares either the $n_1$ or $n_2$ projection of $\mathbf{n}'$. Assume that it is the former; the other case is similar. We can write $\mathbf{n}' = (x_1, 0)\alpha_1 + (0, x_2)(\alpha_2 + \beta) + \mathbf{o}$ and $\mathbf{n} = (x_1, 0)\alpha_1 + (0, x_2)\alpha_2 + \mathbf{y}$ for some $\alpha_1, \alpha_2, \beta \in \mathbb{N}$. Notice that $\varphi(\mathbf{n}') = \varphi(\mathbf{n}) + \beta\langle \mathbf{a}, (0, x_2)\rangle$ and since $\varphi(\mathbf{n}) \in \mathbb{N}$ it remains only to show that $\beta\langle \mathbf{a}, (0, x_2)\rangle \in \mathbb{N}$. Since $D$ is a wedge domain, the threshold sets which define its upper and lower boundary are not parallel, meaning that we can find two points in $D$, $\mathbf{n}_1$ and $\mathbf{n}_2$ such that $\mathbf{n}_1 - \mathbf{n}_2 = (0, x_2)\beta$ (indeed there are infinitely many such points). Thus, $\beta\langle \mathbf{a}, (0, x_2)\rangle = \varphi(\mathbf{n}_1) - \varphi(\mathbf{n}_2) \in \mathbb{N}$, completing the proof. □