11-2022

Reachability in Restricted Chemical Reaction Networks

Robert M. Alaniz

Bin Fu

Timothy Gomez

Elise Grizzell

Andrew Rodriguez

See next page for additional authors

Authors

Robert M. Alaniz, Bin Fu, Timothy Gomez, Elise Grizzell, Andrew Rodriguez, Robert Schweller, and Tim Wylie

Reachability in Restricted Chemical Reaction Networks *

Robert M. Alaniz[1], Bin Fu[1], Timothy Gomez[2], Elise Grizzell[1], Andrew Rodriguez[1], Robert Schweller[1], and Tim Wylie[1]

[1]University of Texas Rio Grande Valley, Edinburg, TX, USA.,
{robert.alaniz01,bin.fu,elise.grizzell01,andrew.rodriguez09,robert.schweller,timothy.wylie}@utrgv.edu
[2]Massachusetts Institute of Technology, Cambridge, MA, USA., tagomez7@mit.edu

Abstract

The popularity of molecular computation has given rise to several models of abstraction, one of the more recent ones being Chemical Reaction Networks (CRNs). These are equivalent to other popular computational models, such as Vector Addition Systems and Petri-Nets, and restricted versions are equivalent to Population Protocols. This paper continues the work on core *reachability* questions related to Chemical Reaction Networks; given two configurations, can one reach the other according to the system's rules? With no restrictions, reachability was recently shown to be Ackermann-complete, which resolved a decades-old problem.

In this work, we fully characterize monotone reachability problems based on various restrictions such as the allowed rule size, the number of rules that may create a species (k-source), the number of rules that may consume a species (k-consuming), the volume, and whether the rules have an acyclic production order (*feed-forward*). We show PSPACE-completeness of reachability with only bimolecular reactions in two-source and two-consuming rules. This proves hardness of reachability in Population Protocols, which was unknown. Further, this shows reachability in CRNs is PSPACE-complete with rules of size two, which was previously only known with input rules of size five. This is accomplished using new techniques within the motion planning framework.

We give several important results for feed-forward CRNs, where rules are single-source or single-consuming. We show that reachability is solvable in polynomial time as long as the system does not contain special *void* or *autogenesis* rules. We then fully characterize all systems of this type and show that if you allow void/autogenesis rules, or have more than one source and one consuming, the problems become NP-complete. Finally, we show several interesting special cases of CRNs based on these restrictions or slight relaxations and note future significant open questions related to this taxonomy.

*This research was supported in part by National Science Foundation Grant CCF-1817602.

Contents

1 Introduction

The popularity of molecular computation and the need to model distributed reactions has given rise to several models of abstraction and multiple areas of research. Many of these models arose naturally in different fields decades apart, yet mathematically are nearly equivalent. The focus of this paper is on Chemical Reaction Networks (CRNs) [5,6], a model equivalent [12] to Vector Addition Systems (VASs) [22] and Petri-Nets [27]. Further, the Population Protocols model [2] is just a restricted version of these models, limited by the number of input and output elements in each operation.

Although these models may be substantively equivalent, we focus on CRNs for two reasons, first, due to the simplicity and convenience of the system definition. Specifically, expressing the production operations through reaction rules gives a straightforward rubric to measure and characterize the power of the system. The second is because CRNs are also the oldest formulation of these types of distributed systems, although its definition evolved considerably early on. Even though the reaction rule description is more intuitive in many cases, in our formal definitions, we rely on a matrix interpretation for precision similar to Vector Addition Systems.

From the initial inceptions of each of these models, two of the most fundamental questions have always been *production* and *reachability*. Production asks if some species/element can ever be produced. Reachability simply asks if, given a system and initial configuration, whether it will even reach another specific configuration. As the expectation of behaviors and the attainability of end goals are necessary in the design of any of these systems, production and reachability are at the heart of system design. Despite the fundamental nature of reachability questions, most remain open or have only recently been solved, while production is better understood overall. Since many results are applicable between these models, we briefly survey the reachability question in relation to each of the models.

1.1 An Overview of Related Models

Petri-Nets. Petri-Nets were first formally introduced in [27] in 1962 to visually describe chemical processes. Petri-Nets provide a graphical representation of reactions by having *places* with edges to *transitions*, which have edges to other places. Weights accompanying these edges correspond to consumption within reactions between different elements. The reachability problem was shown to be EXPSPACE-hard in 1976 [25] and was shown to be decidable five years later [26]. Reachability in Petri-Nets has continued to be an active area of research with numerous extensions and restrictions on the graph structure. The unrestricted version of the reachability problem was not proven until 2021, when it was shown to be Ackermann-complete [14, 24].

Vector Addition Systems. Vector Addition Systems were formulated shortly after Petri-Nets in 1969 [22]. A VAS has an initial matrix of nonnegative integers and a set of operational integer vectors. The system is allowed to add these vectors however desired as long as none of the values in the initial/counter matrix ever become negative. Possibly due to the matrix interpretation and convenience of calculations, or simply because it originated in Mathematics, VAS has often been the more popular model among these equivalent models.

In fact, the unrestricted reachability results from the other models were typically proven for VASs. Thus, reachability was proven EXPSPACE-hard for VASs in [25] and decidable in [26]. Proven in [20], reachability is decidable with dimension ≤ 5 since the set of reachable configurations is a semilinear set. It was also shown that there exists 6-dimensional VASs whose reachable configurations can form non-semilinear functions. This was proven by introducing VAS with states (VASS), another well-studied generalization, but is equivalent to standard VASs with an extra three dimensions. In 2-Dimensional Unary VASS, reachability is PSPACE-complete if the values are encoded in binary and NL-complete if encoded in unary [8]. A survey of VAS reachability was presented in 2016 [30]. Again, only in 2021 was completeness proven [14, 24]. Similar problems, coverage and boundness, were shown to be in EXPSPACE [29]. The first asks if there exists a reachable configuration with at least the given number of each species. The second is if the set of reachable configurations is finite.

Population Protocols. Population Protocols can be viewed as a restricted form of CRNs, VASs, and Petri-Nets. It was introduced in 2004 [1,2], motivated by a desire for distributed decentralized computing by agents that were highly resource-limited. This has many applications in areas such as swarm robotics, sensor networks, and bimolecular reactions in chemistry. Due to the limited nature of the agents, all reactions are volume-preserving with two inputs and two outputs. This can be thought of as two agents communicating and choosing to change state or not based on this communication.

Due to this limited communication protocol, this is only a subset of the other systems where the input and output rules must be two. For reachability, the model does not suffer from problems that might occur from exponential growth and is therefore easier. Shortly after its introduction, [3] showed that all predicates stably computable in this model are semilinear. There has been little work on the complexity of reachability in the model, with most of the focus being on efficiently computing certain tasks, such as leader election. However, a fairly straightforward reduction could show NP-hardness of reachability, even though we could not find this in the literature. Recent work investigated the complexity of verifying that a given Population Protocol computes a specific function, and shows that it is at least as hard as reachability in Petri-Nets [18].

Chemical Reaction Networks. The desire and attempt to bring a mathematical abstraction to model chemical reactions has been well-documented for over a hundred years, but began in its modern incarnation during the 1960s as chemical reaction network theory [5, 6]. Given that CRNs are motivated by actual chemical processes, there is a reasonably distinct split between the purely mathematical and applied experimental work in the research literature. We address that here regarding implementations and the reachability problem by separating the results for applied work into the stochastic section.

For standard CRNs, as with the other models, EXPSPACE-hardness [25] and decidability [26] were solved based on VASs first. In 2012, it was shown that, with size five or more rules, reachability is PSPACE-hard [33]. Specifically, this was shown for *Proper* CRNs, where reactions do not change the system's total volume. The reduction presented there extends to the production problem, which asks if a copy of a specific target species can be produced. The eventual complexity, again, was resolved in 2021 by showing that it was Ackermann-complete [14, 24].

Stochastic Chemical Reaction Networks. The work to reproduce results from CRNs and to adapt the model for application has resulted in a flurry of experimental work in the past few decades. The model itself can be interpreted and implemented in many different ways. One recent innovation is the implementation in DNA through DNA Strand Displacement (DSD) systems as described in [32]. Many techniques have been developed to improve further strategies for constructing these systems [21, 36]. There are even several CRN to DSD compilers that have been designed and implemented [7, 23, 34, 38].

Deterministic Chemical Reaction Networks can compute precisely the set of semilinear functions [3, 10]. However, when allowing a small error, CRNs are capable of Turing universal computation [31]. These stochastic CRNs were even shown to implement universal computation with probability one, in the sense that the computation eventually stops with the correct answer [13].

Rate-Independent Continuous Chemical Reaction Networks is a continuous version of the model introduced in [11]. The systems studied had the *feed-forward* property, which requires that there exists an ordering on the rules such that species produced in earlier rules, "feed-forward" to be used in later rules. Later work, [35], introduced the concept of *non-competitive* rules, a stricter constraint than single-consuming, and show a connection with neural networks. They show that with both the feed-forward and non-competitive property, programming a rate-independent CRN is equivalent to training a rectified linear unit (ReLU) neural network.

For these stochastic versions, determining if a CRN computation terminates is decidable, but becomes undecidable when asking if it terminates with any probability [37]. In the continuous case, reachability was shown to be in P, while deciding if the target configuration is reachable with a size-k subset of the reactions is NP-complete [9].

General CRNS					
Cons.	*Source*	*Rule Size*	*Volume*	*Membership*	*Theorem*
2	2	$(2,2)$	U/B	PSPACE-complete	Thm. 3.3
2	2	$(1,2)$ and $(2,1)$	U/B	PSPACE-hard	Thm. 3.5
j	k	$(1,1)$	U/B	NL-hard	Thm. 3.7
Feed-Forward					
Cons.	*Source*	*Rule Size*	*Volume*	*Membership*	*Theorem*
2	2	$(1,2)$ or $(2,2)$ or $(2,1)$	U/B	NP-complete	Thm. 4.6
j	1	No Void	U/B	P	Thm. 4.18
1	k	No Autogenesis	U/B	P	Thm. 4.21
1	1	Any	U/B	P	Cor. 4.22
Feed-Forward with Void/Autogenesis Rules					
Cons.	*Source*	*Rule Size*	*Volume*	*Membership*	*Theorem*
3	0	$(3,0)$	U/B	NP-complete	Thm. 5.1
j	0	$(2,0)$	B	P (Bipartite)	Thm. 5.5
j	0	$(2,0)$	U	P	Thm. 5.3

Table 1: Table of reachability results. The *Volume* column indicates the input encoding of the volume, and thus U is for unary and B is for binary. Rule sizes (a,b) indicates a elements interact as input to produce b elements as output.

1.2 Our Contributions

In this paper, we improve on nearly all known results related to monotone volume restricted CRNs based on several aspects of the rules. An overview of the major results is listed in Table 1. The results in the three sub-tables roughly correspond to the major results in Sections 3, 4, and 5, respectively.

Proper/General CRNs. The general reachability problem in CRNs is Ackermann-complete [14,24], and thus we first focus on the restriction of *proper* CRNs, in which each reaction/rule preserves or decreases system volume, putting the problem in PSPACE. Previous work has shown that reachability with quintuple-molecular reactions (up to 5 products and reactants per rule) is PSPACE-complete [33]. However, practical considerations motivate the study of smaller rules, such as bimolecular reactions in which up to 2 molecules are used as products (such as in Population Protocols). We show that reachability is PSPACE-complete (Theorem 3.3) in this case, and thus, reachability in population protocols is PSPACE-complete. Moreover, we show this hardness result holds even when each species within the system is produced by at most two rules in the system and consumed by at most two rules in the system (2-source and 2-consuming). We show in Corollaries 3.4 and 3.5 this reduction extends to the universal reachability problem, and reachability for non-monotone volume. For production, our PSPACE-Complete result is tight in regards to rule size as we show NL-Completeness for $(1,1)$ rules in Theorem 3.7. We also provide several related smaller results.

Feed-Forward CRNs. We fully characterize, based on rule size, void/autogenesis rules, and the number of source/consuming rules, all CRN systems that are feed-forward, which is where reactions of the CRN permit an ordering such that products of later reactions do not occur as reactants in earlier reactions. We show that, without void/autogenesis rules, the feed-forward property alone moves the reachability problem into the class NP (Theorem 4.4). We show that reachability is polynomial-time solvable for a feed-forward system if it is *either* 1-consuming *or* 1-source and uses no *void* rules (rules that produce no species) or *autogenesis* rules (rules that consume no species). We further show that relaxing these restrictions makes reachability provably hard in that reachability in 2-consuming and 2-source feed-forward systems in NP-complete (Theorem 4.6), as are feed-forward systems that allow void or autogenesis rules.

Void and Autogenesis Rules. Our final consideration is the case of systems that utilize a particularly powerful class of rules: *void* rules which do not produce any species types, and *autogenesis* rules which do not consume any species types. We show that reachability is NP-complete even if system reactions are all void rules or all autogenesis rules of size $(3, 0)$ or $(0, 3)$, respectively (Theorem 5.1). We then explore the case of $(2, 0)$ void rule systems and show that reachability is polynomial-time solvable if the volume of the input configuration is polynomial bounded (Theorem 5.3), *or* if the CRN is *bipartite* (Theorem 5.5). For other $(2, 0)$ void rule systems, we leave the complexity of reachability as an open problem.

Organization. Due to the numerous definitions, parameters, and subsequent results included, we have attempted to organize the paper based on the three main areas discussed in Table 1 as a coarse guide of the major results. There are many smaller results, and we have included the Table of Contents for the initial review to show the entire paper structure in detail. We cover the model and restrictions definitions in Section 2. The three main result areas are general/proper CRNs (Section 3), feed-forward CRNs without void/autogenesis rules (Section 4), and feed-forward CRNs with void/autogenesis rules (Section 5). We then conclude the paper with several important open problems in Section 6.

2 Preliminaries

2.1 Basics

Let $\Lambda = s_1, s_2, \ldots, s_{|\Lambda|}$ denote some ordered alphabet of *species*. A configuration over Λ is a length-$|\Lambda|$ vector of non-negative integers, denoting the number of copies of each present species. A *rule* or *reaction* has two multisets, the first containing one or more *reactant* (species) used to create resulting *product* (species), the second multiset. We represent each rule as an ordered pair of configuration vectors $R = (R_r, R_p)$. R_r contains the minimum counts of each reactant species necessary for reaction R to occur, where reactant species are either *consumed* by the rule in some count or leveraged as *catalysts* (not consumed); in some cases a combination of the two. The product vector R_p has the count of each species *produced* by the *application* of rule R, effectively replacing vector R_r. The species corresponding to the non-zero elements of R_r and R_p are termed *reactants* and *products* of R, respectively.

The *application* vector of R is $R_a = R_p - R_r$, which shows the net change in species counts after applying rule R once. For a configuration C and rule R, we say R is applicable to C if $C[i] \geq R_r[i]$ for all $1 \leq i \leq |\Lambda|$, and we define the *application* of R to C as the configuration $C' = C + R_a$. For a set of rules Γ, a configuration C, and rule $R \in \Gamma$ applicable to C that produces $C' = C + R_a$, we say $C \rightarrow^1_\Gamma C'$, a relation denoting that C can transition to C' by way of a single rule application from Γ. We further use notation $C \rightsquigarrow_\Gamma C'$ to signify the transitive closure of $\rightarrow^1_\Gamma$ and say C' is *reachable* from C under Γ, i.e., C' can be reached by applying a sequence of applicable rules from Γ to initial configuration C. We use the following notation to depict a rule $R = (R_r, R_p)$:

$$\sum_{i=1}^{|\Lambda|} R_r[i]s_i \rightarrow \sum_{i=1}^{|\Lambda|} R_p[i]s_i$$

For example, a rule turning two copies of species H and one copy of species O into one copy of species W would be written as $2H + O \rightarrow W$.

Definition 2.1 (Discrete Chemical Reaction Networks)**.** A discrete chemical reaction network (CRN) is an ordered pair (Λ, Γ) where Λ is an ordered alphabet of species, and Γ is a set of rules over Λ.

The primary computational problem we consider in this paper is the *reachability* problem. We consider additional problems in the paper, such as determining if it is possible to produce a given amount of a particular species from an initial configuration of a CRN, as well as universal reachability, which asks if the target configuration is reachable for all reaction application sequences.

$a + b \to c$	$a + b \to c$	$a + b \to c$	$a + b \to c + b$	$a + b \to a + c$
$a + c \to b$	$a + c \to d$	$a + c \to d$	$b + c \to b + a$	$a + c \to a + d$
$d + e \to b$	$b + c \to c$	$b + c \to d + c$	$d \to d + b$	$c + d \to e + f$
(a)	(b)	(c)	(d)	(e)

Figure 1: Example CRN rules to demonstrate the primary restrictions. (a) A 2-source/2-consuming CRN system that is not feed-forward without any void/autogenesis rules. This is not feed-forward since the first two rules cannot appear before or after each other in a feed-forward ordering since each rule contains products that occur as reactants in the other rule. (b) A feed-forward CRN system that is 1-source/2-consuming with a void rule. A valid feed-forward ordering consists of the first rule, followed by the third rule, followed by the second rule. (c) A feed-forward CRN system that is 2-source/2-consuming with a catalyst rule. (d) A non feed-forward CRN system that is 1-source/1-consuming with an autogenesis rule. (e) A non feed-forward CRN that is 1-source/2-consuming with no void/autogenesis rules.

Definition 2.2 (Reachability Problem.)**.** Given a CRN (Λ, Γ), an initial configuration I, and a destination configuration D, the *Reachability Problem* is to compute whether or not D is reachable from I with respect to Γ.

Definition 2.3 (Production Problem)**.** Given a CRN (Λ, Γ), an initial configuration A, a species $s_i \in \Lambda$, and a positive integer k, decide if there exists a reachable configuration B such that $B[i] \geq k$.

Definition 2.4 (Universal Reachability Problem.)**.** Given a CRN (Λ, Γ), an initial configuration I, and a destination configuration D, the *Universal Reachability Problem* is to compute whether or not D is reachable from all configurations M that are reachable from C with respect to Γ.

2.2 Primary Restrictions

We consider the reachability problem under several different restrictions defined below.

Definition 2.5 (Feed-Forward)**.** A CRN (Λ, Γ) is *feed-forward* if Γ permits an ordering on the rules such that the products of any given rule never occur as reactants for earlier rules of the ordering.

Figure 1 (b) and (c) provides examples of a simple set of 3 rules that permit a feed-forward ordering, while (a), (d), and (e) provide examples that do not permit a feed-forward ordering.

Each rule in a system produces some species and consumes others. The following metric places a maximum bound on the number of rules that either produce a given species (j-source) or consume a given species (j-consuming). Figure 1 provides examples of 1 and 2 consuming/source systems, along with example catalyst reactions.

Definition 2.6 (j-source, j-consuming)**.** A species s_i is *consumed* in rule $R = (R_r, R_p)$ if $R_r[i] > R_p[i]$, *produced* if $R_r[i] < R_p[i]$, and is a *catalyst* in rule R if $R_r[i] = R_p[i] > 0$. A CRN (Λ, Γ) is j-source if for all species $s \in \Lambda$, s is produced in at most j distinct rules in Γ. A CRN (Λ, Γ) is j-consuming if for all species $s \in \Lambda$, s is consumed in at most j distinct rules in Γ. We use the terms *single-source* and *single-consuming* for the special cases of 1-source and 1-consuming CRNs, respectively.

The next concept is a special class of rules that either produce nothing (void rules) or consume nothing (autogenesis rules). This could potentially be motivated by evaporation, the ability to pass through a membrane or the spontaneous appearance of ions in a vacuum. Figure 1 provides several examples with void and autogenesis rules.

Definition 2.7 (Void and Autogenesis rules)**.** A rule $R = (R_r, R_p)$ is a *void* rule if $R_a = R_p - R_r$ has no positive entries. A rule is an *autogenesis* rule if R_a has no negative values.

2.3 Additional Restrictions

We also consider the complexity of reachability and production with respect to the *size* of rules.

Definition 2.8. The *size/volume* of a configuration vector C is $\texttt{volume}(C) = \sum C[i]$.

Definition 2.9 (size-(i,j) rules)**.** A rule $R = (R_r, R_p)$ is said to be a size-(i,j) rule if $(i,j) = (\texttt{volume}(R_r), \texttt{volume}(R_p))$. A reaction is bimolecular if $i = 2$ and unimolecular if $i = 1$.

Definition 2.10 (Volume Decreasing, Increasing, Preserving)**.** A rule $R = (R_r, R_p)$ of size-(i,j) is said to be volume decreasing if $i > j$, volume increasing if $i < j$, and volume neutral if $i = j$. A CRN (Λ, Γ) is said to be volume decreasing (respectively increasing, preserving) if all rules in Γ are volume decreasing (respectively increasing, preserving). Note: In previous work, volume preserving has been called *Proper*.

A special subset of CRN systems studied in the literature is Population Protocols, in which agents bump into each other and adjust their state according to rules. This model is equivalent to a CRN that is limited to exactly volume 2 for both the reactants (the two agents that bump into each other) and the products (representing the two new states of the agents after the collision).

Definition 2.11 (Population Protocols)**.** A CRN (Λ, Γ) in which all rules in Γ are size-$(2,2)$ is called a population protocol.

3 Reachability in General CRNs

The main result of this section is PSPACE-completeness of the reachability problem with 2-consuming, 2-source, size $(2,2)$ reactions in Theorem 3.3. En route, we prove that production is PSPACE-complete with Theorem 3.2. We extend this reduction in two ways. First, in Corollary 3.4, we prove our reduction holds for the universal reachability problem, and second, we show the reduction holds for non-monotonic volume with rule sizes of $(1,2)$ and $(2,1)$ in Theorem 3.5. We follow with several interesting, yet minor results.

3.1 Gadget Reconfiguration Framework

Our main result is based on the motion planning problem through Toggle-Lock and Rotate gadgets [16]. Motion planning is PSPACE-hard with only a rotate gadget and any reversible gadget with interacting tunnels, a class that includes the Toggle-Lock [15]. The motion planning problem considers two input configurations of gadgets, a start location for the agent, and a target location, and asks if the agent can reach the target location. [1]

A gadget consists of a set of labeled ports and states describing the gadget's legal traversals, which may change the state of the gadget. The Toggle-Lock Gadget has an unlocked and locked state, shown in Figure 2a and 2b, respectively. The top path is directed, and the agent must follow the direction. Traversing this path changes the state of the gadget. Traversal of the bottom tunnel can only occur in the unlocked state; this can be done in either direction and does not change the state of the gadget. The rotate gadget only has one state that sends the agent to the next port going clockwise. A motion planning system consists of a set of gadgets, a set of wires denoting port connections, and an initial signal location.

We are simulating a 0-player gadget framework where the agent makes no choices. The agent is directed down the wire and turns around if the gadget is not traversable in the current state, corresponding to a deterministic model of computation. An early version of this result appeared in the short abstract [19], which reduced from a different gadget and used the 1-player framework studied in [4, 15, 17]. However, the reduction was not constant source or constant consuming.

Directed Wire. Two ports are connected by a wire. We label each wire with a symbol such as a and include two species $\overrightarrow{a}, \overleftarrow{a}$ to denote the direction moved along the wire. We refer to these as the agent species.

[1] In [16] this problem is called the reachability problem. To avoid confusion, we refer to this as the motion planning problem.

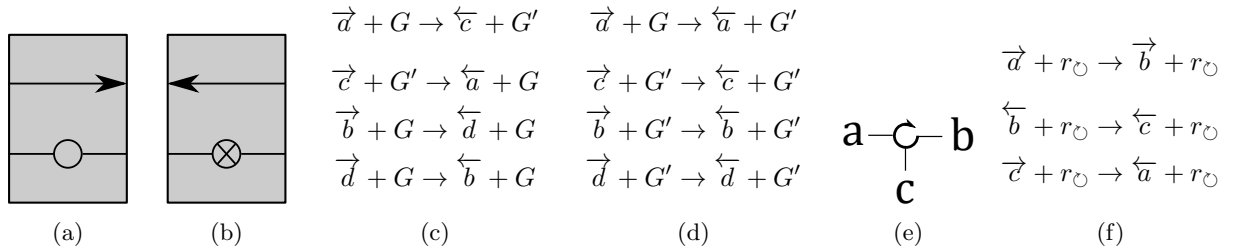


Figure 2: (a) Unlocked state of a Toggle-Lock gadget. (b) Locked state of a Toggle-Lock gadget. (c-d) Reactions which implement a single gadget. (c) represents a successful traversal and (d) represents the 'bound-back' reactions. The arrow is incoming or outgoing from port. (e) The rotate gadget. (f) Rules for the rotate gadget.

Toggle-Lock and Rotate Gadgets. We have two species for each Toggle-Lock that we call the *gate catalysts*. We represent the unlocked gate using the species G, and the locked state with species G'. Traversing the toggle tunnel requires the gadget to be in the correct state for the toggle. The reaction changes the state of the gate catalyst and the agent species. The bottom tunnel can only be traversed if the gate is in the unlocked state. However, this does not change the state of the gate, so the G species acts a catalyst that must be present to traverse the gadget. Figure 2c shows the rules used to implement a toggle lock.

We implement rotate gadgets with a single rotate clockwise species $r_{\circlearrowleft}$. The rotate gadget diagram is shown in Figure 2e. The signal state changes to the outgoing direction of the next wire in the clockwise ordering of ports using the rules shown in Figure 2f. Note that each reaction consumes and creates exactly one agent species yielding the following observation:

Observation 3.1. Any reachable configuration in the reduction only contains a single agent species.

3.2 Production

We prove production is PSPACE-complete using the framework described above. The target species is the agent species representing the target wire.

Theorem 3.2. Production in 2-source, 2-consuming preserving CRNs is PSPACE-complete with only bimolecular reactions.

Proof. Membership in PSPACE for Proper CRNs was shown in [33]. Given an instance of the motion planning problem with toggle-locks and rotate gadgets, create a CRN ruleset and configuration as described above. Our starting configuration of the CRN encodes the start location and starting states of the gadgets. The target species we wish to produce is the agent species for the target location in the motion planning problem.

The agent may only traverse a toggle-lock gadget if the gate catalyst is in the correct state. The rotate gadget sends the signal to the correct next state. Once the agent reaches the target wire, the agent species representing it is produced. If the agent reaches the target location, then reactions apply representing the path of the agent to produce the target species.

From Observation 3.1, there only exists one agent species and the reactions encode only valid traversals through the gadgets. If the target species is produced, then the sequence of reactions to produce it represents the path of the agent though the system of gadgets. Each gate catalyst is only produced and consumed in the reaction that implements the toggle tunnel. Each agent species is directed, so it also is only produced and consumed in one rule.

Note that if an agent ever attempts to cross a gadget in a state that does not allow that traversal, the configuration will no longer have any valid reactions. This is the equivalent to an illegal move in the motion planning problem and the agent cannot progress, so the game is over. □

3.3 Reachability

We extend the reduction above to work for reachability by taking advantage of the fact that the toggle-lock is reversible. Once we reach the target species we flip the rotate catalyst to allow the agent to move counterclockwise through a rotate gadget, allowing us to undo all the changes to the gadget states. When the agent reaches the starting location again, the system is in the initial configuration except with the opposite rotate catalyst. The reconfiguration problem for the 1-player version of the motion planning framework was shown to be PSPACE-complete using a similar technique where the agent changes the state of the final gadget then undoes all of its previous movements [4].

Theorem 3.3. Reachability in 2-source, 2-consuming preserving CRNs is PSPACE-complete with only bimolecular reactions.

Proof. The reduction from Theorem 3.2 can be extended to show the reachability problem is PSPACE-hard as well. We add an additional species $r_{\circlearrowleft}$ which is the rotate counterclockwise catalyst. The target configuration is the same as the initial configuration except with the counterclockwise catalyst $r_{\circlearrowleft}$. If the target wire is a, we add the rule $\overrightarrow{a} + r_{\circlearrowleft} \to \overleftarrow{a} + r_{\circlearrowleft}$, which changes the direction of the rotate catalyst and turns the agent around on the wire. Since the toggle lock gadget is reversible, the agent will undo all of its moves and return to the start configuration. Since the gadget system has the property of being reversible, this backwards traversal is possible. □

This reduction extends to the universal reachability problem since there is only a single reaction at each step that can be performed.

Corollary 3.4. Universal reachability in 2-source, 2-consuming preserving CRNs is PSPACE-complete with only bimolecular reactions.

Proof. From Observation 3.1, we know there only exists one agent state in the system at a time. Since the system is single-consuming, the agent state is only consumed in a single rule. These two points mean there only exists a single move sequence, so if the target configuration is reachable, it is universally reachable. □

3.4 Volume Related Results

Here, we look at several restrictions related to rule size, and consequently, volume.

3.4.1 Non-Monotone Volume.

In this section we extend the reduction to utilize smaller rules. We show PSPACE-hardness of production and reachability when allowing both $(1, 2)$ and $(2, 1)$ rules. The CRN has both volume increasing and decreasing rules, which means it is non-monotone, and thus, these problems are not known to be in PSPACE. To prove this, we add an intermediate species for each reaction. We replace the reaction $\overrightarrow{a} + G \to \overrightarrow{c} + G'$ with the two reactions $\overrightarrow{a} + G \to \overrightarrow{aGc}$ and $\overrightarrow{aGc} \to \overrightarrow{c} + G'$.

Corollary 3.5. Production and reachability in 2-source, 2-consuming CRNs is PSPACE-hard with rules of size $(2, 1)$ and $(1, 2)$.

Proof. The reduction behaves the same way as in Theorems 3.2 and 3.3, however, here, we either have a single agent species, or a single intermediate species. □

3.4.2 Unary Encoded Volume

When a system is volume-increasing or volume-decreasing, the reaction sequence length is polynomial in the volume of the system, so we achieve the following theorem.

Theorem 3.6. Reachability is in NP for Volume Increasing or Volume decreasing CRNs, with the volumes encoded in unary.

Proof. When the volume of a system is strictly increasing or decreasing, any reaction sequence between I and D is bounded by of size $\leq |I - D|$. The sequence can then be given as a 'yes' certificate for reachability. □

3.4.3 Unimolecular Reactions

Unimolecular reactions are of the form $A \to B$, i.e. preserving rules of size 1. If we are limited to only this type of reaction production is NL-complete. The NL-hardness result works for reachablility as well.

Theorem 3.7. Reachability and production in CRNs is NL-hard with rules of size $(1, 1)$.

Proof. Given a directed graph G we create a set of species and reactions as follows. For each node $v \in G$ we create a species. For each edge $(a, b) \in E$ we create a reaction $a \to b$.

We reduce from the directed path problem: given two nodes $s, t \in G$, does there exist a path between s and t? Let our initial configuration be a single copy of the species representing s and our target configuration be a single copy of t. At each step the species will represent the current node in the path to reach t if and only if there exists a path. □

Theorem 3.8. Production with rules of size $(1, 1)$ is NL-complete

Proof. Non-deterministically select a species i with a positive count in the initial configuration, check if the target species is reachable from i, if yes then the target species is producible. Checking reachability is in NL since the reactions can be viewed as directed edges. NL-hardness comes from Theorem 3.7 □

3.5 Results Related to Single-Consuming or Single-Source

Although the majority of the results in this section relate to rule size, we also look at rule restrictions based on the number of source and consuming rules. Our main result shows that reachability in a 2-consuming, 2-source general CRN system is PSPACE-complete. In Sections 4 and 5, we fully characterize these systems if the rules are feed-forward. Here, we address a few interesting results for CRNs that are not feed-forward.

3.5.1 Species-based Restrictions

Our definition of consuming and source is based on all species in the system. If this rule is relaxed to mean j-consuming/k-source per species, then we prove reachability is NP-hard in Corollary 4.11 even when all species are 3-consuming/0-source except for 1 species being 0-consuming/k-source. The reduction from 3-Dimensional Mapping creates a CRN that is feed-forward with no void/autogenesis rules, and thus it was placed with those results as being NP-complete. However, without feed-forward, the problem here is not necessarily in NP.

3.5.2 Production is NP-Hard

Here, we show that production is NP-hard in single-consuming CRNs.

Theorem 3.9. Production in single-consuming CRNs is NP-hard.

Proof. Consider an instance of 3SAT where X is the set of variables and C is the set of clauses. For each $x_i \in X$ we include $x_i, \overline{x_i}, x_i^T, x_i^F$ in our species set. The starting configuration includes a copy of each of the species $x_i, \overline{x_i}$. For each $c_j \in C$ with $c_j = (x_a, x_b, x_c)$, we include $c_j^a, c_j^b, c_j^c, c_j^{SAT}, SAT_j$. The starting configuration includes a copy of each species c_j^a, c_j^b, c_j^c and SAT_0. The target species we wish to produce is $SAT_{|C|}$.

Two assignments species T, F are included. One copy of each of these species is included in the starting configuration that acts as a catalyst. There are two rules for each x_i: $T + x_i + \overline{x_i} \to T + \overline{x_i} + x_i^T$ for assigning true, and $F + x_i + \overline{x_i} \to F + x_i + x_i^F$ for assigning false. We add a rule $c_j^a + x_a^T \to c_j^{SAT} + x_a^T$ for positive literals and $c_j^a + x_a^F \to c_j^{SAT} + x_a^F$ for negated literals. To verify each clause we include the rules, $SAT_j + c_j^{SAT} \to SAT_{j+1}$

The target $SAT_{|C|}$ is only producible by starting with SAT_0 and applying the rule $SAT_j + c_j^{SAT} \to SAT_{j+1}$ $|C|$ times to verify the satisfaction of each clause. □

4 Reachability in Feed-Forward CRNs

Having established PSPACE-completeness for general CRNs, we consider the feed-forward restriction in which rule sets do not have cycles. Feed-forward CRNs are motivated in that they allow functional composition of CRNs [35]. We characterize the complexity of reachability for feed-forward CRNs under the assumption that the system does not contain either void or autogenesis rules, leaving a focused consideration of void and autogenesis rules for Section 5.

We first show NP-completeness of feed-forward systems in Section 4.1, even in the case of size-$(2,2)$ rules (i.e., bimolecular rules / Population Protocols), while at the same time being only 2-source and 2-consuming. We then show a polynomial-time solution to reachability in Section 4.2 for any feed-forward system that is either 1-source or 1-consuming, thus giving a complete characterization of feed-forward reachability.

4.1 NP-completeness for Bimolecular Reactions

In this section, we show that reachability (and production) in feed-forward systems is NP-complete for bimolecular reactions (rules of size at most $(2,2)$), even when the CRN is 2-source and 2-consuming. This hardness result is tight because a decrease to either 1-source or 1-consuming, as shown in Section 4.2, implies a polynomial time solution to reachability. We start with proof of membership in NP, followed by NP-hardness from a reduction from the Hamiltonian Path problem.

4.1.1 NP Membership

A key property of feed-forward systems is that any sequence of rule applications can be reordered such that all system rules are applied consecutively. This new ordering is still a valid sequence of applicable rules that reaches the same final configuration.

Definition 4.1 (Ordered Application)**.** A sequence of reactions is an ordered application if all the applications of any given rule take place right after each other in a contiguous sequence. An example of this is $R_1, R_1, \ldots R_1, R_2 \ldots R_2, R_3$.

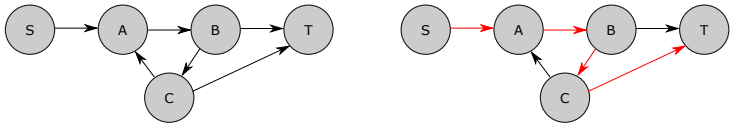
Lemma 4.2. Let $C = (\Lambda, \Gamma)$ be a feed-forward CRN with a feed-forward ordering $F = \{R_0, R_1, \ldots R_{|R|-1}\}$ over Γ. Given configurations c, c', and a sequence of reactions $S = \{\ldots R_j, R_i \ldots\}$ in Γ that converts c to c' where $i < j$, then the sequence $S' = \{\ldots R_i, R_j \ldots\}$, i.e., the sequence obtained by swapping the two rules R_i and R_j, also transforms $c \to c'$.

Proof. Let X denote the configuration obtained by applying the rules of S up to just before the application of rule R_j. Let $R_j = (R_r^j, R_p^j)$ and $R_i = (R_r^i, R_p^i)$. As S is a valid sequence of rule applications, $X - R_r^j$ is a non-negative vector. Due to the feed-forward ordering F, the reactants R_r^i do not occur as products in R_p^j, and thus $X - R_r^j - R_r^i$ is also non-negative, implying that rule R_i and R_j can be applied in either order. □

Corollary 4.3. A configuration D is reachable from a configuration I with a feed-forward CRN if and only if D is reachable from I by an ordered application of rules.

Lemma 4.4. The reachability problem is in NP for feed-forward CRNs that do not use autogenesis rules.

Proof. We utilize the ordered application as a polynomial-sized certificate from Corollary 4.3. While the sequence length (number of total rule applications) could be exponential, the sequence can be stored as a sequence of ordered pairs denoting each rule type in the sequence, along with a binary encoding of the number of times the rule is to be applied. The number of times a given rule type is applied is limited by the volume of the configuration right before such applications since there are no autogenesis rules. Since



(a) Graph (b) Hamiltonian Path

$$\begin{array}{ccccc} \{S_0^* & A & B & C & T\} \\ \{S^v & A_1^* & B & C & T\} \\ \{S^v & A^v & B_2^* & C & T\} \\ \{S^v & A^v & B^v & C_3^* & T\} \\ \{S^v & A^v & B^v & C^v & T^4\} \end{array}$$

(c) Configurations

Figure 3: Our starting configuration $c = \{S_0^*, A, B, C, T\}$. Our goal configuration is $c' = \{S^v, A^v, B^v, C^v, T^4\}$. Each vertex must be changed to the visited state to reach the target, and the T must be the last vertex.

the volume of the initial configuration is small enough to be encoded in a polynomial number of bits, this property is preserved through the application of each rule type, implying the number of applications of each rule type is encoded in a polynomial bounded number of bits. Further, the result of applying a given number of iterations of a particular rule can be computed in poly-logarithmic time in the number of applications, which means the certificate can be checked in polynomial time. □

4.1.2 NP-Hardness

We now show the reachability problem is NP-complete for feed-forward CRNs even for size $(2, 2)$-rules (bimolecular reactions, Population Protocols) and for 2-source, 2-consuming systems. We show this by a reduction from the Directed Hamiltonian Path problem with vertices of in-degree and out-degree of at most 2 [28]. For each vertex X in the graph $G = (V, E)$, we include $2 + |V|$ states: an initial state X, a visited state X^v, and $|V|$ signal states X_i^*. We encode the edges of the graph in the rules as follows,

$$\text{Rules } R = \left\{ \begin{array}{c|c|c} S_i^* + A \to S^v + A_{i+1}^* & B_i^* + C \to B^v + C_{i+1}^* & B_i^* + T \to B^v + T_{i+1}^* \\ A_i^* + B \to A^v + B_{i+1}^* & C_i^* + A \to C^v + A_{i+1}^* & C_i^* + T \to C^v + T_{i+1}^* \end{array} \right\}$$

Definition 4.5 (HAMPATH)**.** Given a graph $G = \{V, E\}$ and two nodes $s, t \in V$, does there exist a path from s to t that visits each node precisely once?

An example reduction is shown in Figure 3. Given this reduction, any Hamiltonian path of graph G has a corresponding sequence of rules that end with every vertex, other than T, represented with the *visited* state, and T represented with the signal state matching the count of the vertices. Conversely, the only way to reach such a configuration corresponds directly to a Hamiltonian path of G from S to T, yielding the following result:

Theorem 4.6. Reachability is NP-complete for feed-forward CRNs with size $(2, 2)$ rules that are 2-source and 2-consuming.

Proof. If there exists a Hamiltonian path P in G, there exists a sequence of rules r_P, each moving the * to a subsequent agent matching the sequence of vertices in P and further setting the previous agent to the visited state. Such a rule sequence ends with a single visited state for each vertex in the graph other than T and a $T_|V|$ state.

Conversely, if a sequence of rules r that results in the target configuration exists, then a Hamilton path exists as each rule changes the agent location to the visited state, which can no longer change state, and moves the * to the next agent. The sequence of * species in the configuration represents the Hamilton path.

Finally, note that this system is feed-forward since an agent starts in the initial state, changes to a signal state, then to the visited state. We can ensure the ordering since the signal states are numbered. □

This reduction, with minor modification, can be adapted to achieve the following additional results.

Corollary 4.7. Reachability is NP-complete for feed-forward CRNs that are 2-source and 2-consuming with all rules of size $(2, 1)$.

Proof. This is obtained by simply removing the visited states in the previous reduction. □

Corollary 4.8. Reachability is NP-complete for feed-forward CRNs that are 2-source and 2-consuming and with all rules of size $(1, 2)$.

Proof. This follows from the above corollary combined with Lemma 4.20 (defined in Section 4.2). □

Corollary 4.9. Production is NP-complete for feed-forward CRNs with either size $(2, 2)$ rules, size $(2, 1)$ rules, or size $(1, 2)$ rules, that are 2-source and 2-consuming.

Proof. In the previous reduction, the species $T_{|V|}$ is only producible if it is reached after reaching all other species, implying the previous reduction above holds for the production problem. □

Species-based Restrictions. Our definition of consuming and source is based on all species in the system. If this rule is relaxed to mean j-consuming/k-source per species, then we show the problem is NP-hard by a reduction from the 3-Dimensional Matching (3DM) problem.

Definition 4.10 (Three Dimensional Matching Problem (3DM))**.** The 3DM problem takes as input a hypergraph $H = (X, Y, Z, T)$ where X, Y, Z are three disjoint sets, and $T \subseteq X \times Y \times Z$ is a set of *hyperedges*. The output is whether or not there exists a subset of T that covers all vertices in H without any overlap.

Corollary 4.11. Reachability in CRNs with each species being k-consuming/1-source or 1-consuming/k-source is NP-complete even with only one species being different than the others and the system being feed-forward without void/autogenesis rules.

Proof. We reduce from the 3DM problem, which is hard even when each vertex is covered by at most 3 hyper edges. Let $H = (X, Y, Z, T)$ be an input to the 3DM problem. From this, create an input to the reachability problem as follows. Let $\Lambda = \{S_v | v \in X \bigcup Y \bigcup Z\} \bigcup \{a\}$, and let $\Gamma = \{S_x + S_y + S_z \to a | (x, y, z) \in T\}$. The initial configuration I is the configuration in which each species has count 1 except species a with count 0. Let D be the configuration in which each species has count 0 except a, which has count $|X| = |Y| = |Z|$. Then D is reachable from I under (Λ, Γ) if and only if H has a three-dimensional mapping.

Species a is never consumed, but is produced by all $k = |X| = |Y| = |Z|$ of the rules. Thus, it is 0-consuming/k-source. All other species are never produced, and are consumed in 3 different rules. Thus, they are 3-consuming/0-source. Finally, since species A is never consumed, any ordering of the rules is feed-forward, and there are no void or autogenesis rules used. By Lemma 4.4, the problem is in NP. □

4.2 Feed-Forward, Single-Consuming/Single-Source

In this section, we establish Theorem 4.18 that shows the reachability problem is polynomial-time solvable for feed-forward, single-source rule sets that do not use void rules. We then extend this into Theorem 4.21 to show that reachability in feed-forward, single-consuming systems without autogenesis rules is also polynomial time solvable. Lastly, we give Corollary 4.22 that states the reachability problem for feed-forward, 1-source, and 1-consuming rule sets is polynomial-time solvable.

We now introduce some machinery for constructing an efficient algorithm for solving reachability in the special case of feed-forward, single-source rule sets with no void rules.

Definition 4.12 (Leaf Rule.)**.** A rule $R \in \Gamma$ is a *leaf rule* for Γ if the products of R do not occur as reactants within any *other* rule of the system Γ (however, the products of a leaf rule may occur as reactants within the same leaf rule). Note that a leaf rule could also be a void rule.

Lemma 4.13. A feed-forward, non-empty rule set has at least one leaf rule.

Proof. The final rule in the feed-forward ordering must be a leaf rule, as its product may not occur as a reactant for any previous rule of the system, which includes all rules other than itself. □

Lemma 4.14. If Γ is a feed-forward, single-source rule set without void rules, then so is any subset of Γ.

Proof. This follows from the definitions of feed-forward, single-source, and void rules. □

Definition 4.15 (Pruned Configuration.)**.** Consider a configuration I, a target configuration D, and feed-forward rule set Γ with non-void leaf rule $R = (R_r, R_p)$. If there exists a non-negative integer x such that $D(i) \setminus xR_a(i) = I(i)$, for all i where $R_p(i) \neq 0$, and R is applicable to $D \setminus xR_a$, we say that D is prunable towards I with respect to rule R, and define the *pruning* of D towards I with respect to R to be the configuration $\texttt{Prune}(D, I, R) = D \setminus xR_a$. If no such non-negative integer x exists, then we say that D is inconsistent with I for rule R. Note that the integer x, and thus the configuration $\texttt{Prune}(D, I, R) = D \setminus xR_a$, are unique as long as R is not a void leaf rule.

Lemma 4.16. If a configuration D is inconsistent with a configuration I for any leaf rule $R \in \Gamma$, then D is not reachable from I with a rule set Γ.

Proof. As R is a leaf rule, the counts of the product species in rule R are only affected by rule R among the rules of Γ. Therefore, if D is reachable from I, it must be possible to generate the counts of these species specified by D by some number of applications x of rule R. If no such integer exists, which is the definition of inconsistent, then the species counts for R's products cannot equal the counts specified by D, making D unreachable. □

Lemma 4.17. For a rule set Γ and configuration D that is consistent with I for non-void leaf rule $R \in \Gamma$, then D is reachable from I with a rule set Γ if and only if $D' = \texttt{Prune}(D, I, R)$ is reachable from configuration I with a rule set $\Gamma \setminus R$.

Proof. We first show that if $D' = \texttt{Prune}(D, I, R)$ is reachable with a rule set $\Gamma \setminus R$, then so is D with a rule set Γ. This is because once D' is reached, we know from the definition of $D' = \texttt{Prune}(D, I, R)$ that there exists a non-negative integer x such that x applications of rule R to configuration D' yields D, implying that D is reachable.

For the other direction, suppose we can reach D. From the sequence of rule applications that reaches D, there must be exactly some non-negative integer x applications of rule R. Create a modified sequence of configurations by omitting these x operations, and you get configuration $D' = \texttt{Prune}(D, I, R)$ by application of rules from $\Gamma \setminus R$, meaning D' is reachable from $\Gamma \setminus R$. □

Theorem 4.18. The reachability problem is solvable in polynomial time for a rule set Γ that is feed-forward, single-source, and without void rules.

Proof. Let I denote the starting configuration, D denote the destination configuration, and (Λ, Γ) be a feed-forward CRN without void rules for a given reachability instance. The following recursive algorithm solves reachability for a feed-forward, single-source rule set with no void rules.

As a base case, if the input system has 0 rules, then D is reachable if and only if $I = D$. Otherwise, identify a leaf rule R from Γ, which must exist by Lemma 4.13. Check if D is consistent with I for rule R. If not, return false, which is the correct answer by Lemma 4.16. If it is, then let $D' = \texttt{Prune}(D, I, R)$ denote the pruning of D towards I for rule R and return the result of recursively solving reachability with initial configuration I, ruleset $\Gamma \setminus R$, and destination configuration D', which is a valid input to this algorithm as $\Gamma \setminus R$ is assured to be a feed-forward, single-source rule set without void rules by Lemma 4.14, and is assured to yield the correct result by Lemma 4.17. In total, this algorithm executes $|\Gamma|$ prune operations, where each prune operation can be computed by a constant number of arithmetic operations on the volume of species in I and D. □

We now consider the case of reachability in feed-forward, single-consuming systems without autogenesis rules. Our approach is to *reverse* the given rule set Γ and apply our algorithm for theorem 4.18.

Definition 4.19. For a rule set Γ, let $\overleftarrow{\Gamma}$ be the *reverse* of Γ where $\overleftarrow{\Gamma} = \{(a, b) | (b, a) \in \Gamma\}$.

Lemma 4.20. For any two configurations A, B and rule set Γ, B is reachable from A in Γ if and only if A is reachable from B in $\overleftarrow{\Gamma}$.

Theorem 4.21. The reachability problem is solvable in polynomial time for a ruleset Γ that is feed-forward, single-consuming, and without autogenesis rules.

Proof. Given a CRN (Λ, Γ) where Γ is feed-forward, single-consuming, and without autogenesis rules, along with initial configuration I, and destination D, generate rule set $\overleftarrow{\Gamma}$. Note that $\overleftarrow{\Gamma}$ must be single-source as Γ is single-consuming, and must have no void rules since Γ has no autogenesis rules, and must be feed-forward since Γ is feed-forward. We can therefore determine if I is reachable from D under $\overleftarrow{\Gamma}$ in polynomial time by Theorem 4.18, which gives the answer to our original reachability problem by Lemma 4.20. □

Here we consider the case of reachability in feed-forward, 1-source, and 1-consuming systems with no further restrictions on the rule set. A single-consuming rule set will contain at most one void rule, allowing void rules to be considered when pruning a configuration.

Corollary 4.22. The reachability problem is solvable in polynomial time for a rule set Γ that is feed-forward, 1-source, and 1-consuming with no further restrictions on the rule set.

Proof. Let I denote the initial configuration, let D denote the target configuration, and let (Λ, Γ) be a feed-forward, 1-source, and 1-consuming CRN for a given reachability instance. With a single-consuming rule set Γ, any rule R, void or otherwise, may be used to prune D if there exists some non-negative integer x such that $D(i) \setminus xR_a(i) = I(i)$. By the definition of single-consuming, the integer x and the configuration $\texttt{Prune}(D, I, R) = D(i) \setminus xR_a(i)$ remain unique.

It follows that the recursive algorithm used in Theorem 4.18 solves reachability for a feed-forward, single-source, and single-consuming rule set by allowing void rules when pruning. □

5 Void and Autogenesis Rules

In our consideration of feed-forward CRNs, we omitted two classes of rules: *void* rules that consume reactants without creating any products and *autogenesis* rules that create products without consuming any reactants. One reason for separating these rules is that their lack of conservation of mass might mean they are not feasible in some experimental settings. Another important reason is that their inclusion alone substantially impacts the complexity of problems such as reachability.

In this section, we explore the reachability question in the scenario where *all* rules are void rules or, conversely, all rules are autogenesis rules. We show that void rules (or autogenesis rules) alone imply the NP-completeness of reachability, even if such systems are both feed-forward and 0-source. We specifically show NP-completeness for size $(3, 0)$ void rules. We then explore the complexity of reachability with size $(2, 0)$ void rules and provide a polynomial time solution when the system's volume is encoded in unary. We further show a polynomial time solution for binary encoded volume for a restricted class of *bipartite* $(2, 0)$ CRNs. We leave the remaining general case of reachability with $(2, 0)$ void rules as an open question.

We note by Lemma 4.19, that we may prove results for void only rules, and they are equivalent for autogenesis rules.

5.1 $(3, 0)$ void rules / $(0, 3)$ autogenesis rules

We show that reachability is NP-complete by a reduction from the 3-Dimensional Matching (3DM) problem (Definition 4.10).

Theorem 5.1. Reachability in CRNs is NP-complete with only rules of size $(3, 0)$.

Proof. We reduce from the 3DM problem. Let $H = (X, Y, Z, T)$ be an input to the 3DM problem. From this, create an input to the reachability problem as follows. Let $\Lambda = \{S_v | v \in X \bigcup Y \bigcup Z\}$, and let $\Gamma = \{S_x + S_y + S_z \to \varnothing | (x, y, z) \in T\}$. Let configuration I be the configuration in which each species has count 1 and let D be the configuration in which each species has count 0. Then D is reachable from I under (Λ, Γ) if and only if H has a three-dimensional mapping. □

Corollary 5.2. Reachability for CRNs with only rules of size $(0, 3)$ is NP-complete.

Proof. We show this by reduction from the reachability problem with size $(3, 0)$ rules. Consider an instance of the reachability problem with an input of a CRN (Λ, Γ), an initial configuration I, and a destination configuration D in which rules in Γ are exclusively sized $(3, 0)$ void rules. Let $\Lambda' = \Lambda$, $\Gamma' = \overleftarrow{\Gamma}$, initial configuration $I' = D$, and target configuration $D' = I$. Observe that Γ' consists of rules only of size $(0, 3)$. By Lemma 4.20, D' is reachable from I' under ruleset Γ' if and only if D is reachable from I under ruleset Γ. □

5.2 $(2, 0)$ rules with Unary Encoding

Just as we have connected three-dimensional mapping with size $(3, 0)$ void rules to show NP-completeness, we can reduce in the other direction for size $(2, 0)$ rules to provide an efficient solution for unary encoded volumes.

Theorem 5.3. Reachability in CRNs is in P with rules of size $(2, 0)$ if configuration counts are encoded in unary.

Proof. We show this by reducing reachability in this scenario to the two-dimensional matching problem, which has an established polynomial time solution. We first consider the configuration $X = I - D$, creating a graph from this configuration. For each non-zero count species $X(i) > 0$, $X(i)$ vertices are added to the graph of type i. For each rule $i + j \to \emptyset$, we add edges to the graph connecting all vertices of type i to all vertices of type j. Then we have that X can reach the empty configuration if and only if the created graph has a perfect two-dimensional mapping, and thus I reaches D if and only if such a matching exists. □

5.3 $(2, 0)$ rules with Binary Encoding

We now consider $(2, 0)$ with binary encoded species counts, which permits a potentially exponential configuration volume, making the algorithm of Theorem 5.3 no longer polynomial time. In this scenario, we consider a new restriction in which the CRN rules are bipartite:

Definition 5.4. Bipartite CRN. A bipartite CRN (Λ, Γ) is one in which the species Λ can be partitioned into two disjoint sets Λ_1 and Λ_2 such that for each rule $R \in \Gamma$, there are at most 2 reactants of R and they do not occur within the same partition of Λ.

Determining if a CRN is bipartite can be solved in polynomial time by a bipartite graph detection algorithm. If the CRN is bipartite, we reduce the problem to the maximum flow problem.

Theorem 5.5. Reachability is polynomial-time solvable for bipartite CRNs with $(2, 0)$ rules.

Proof. We show this by reducing reachability for a $(2, 0)$ rule bipartite CRN to the maximum network flow problem. Consider an input $(2, 0)$-rule bipartite CRN (Λ, Γ) with partitions Λ_1 and Λ_2, input configuration I, and output configuration D. From this, generate a max-flow instance as follows: for each $s \in \Lambda$, let the network contain a corresponding vertex v_s. For each rule, $a + b \to \emptyset \in \Gamma$, add an infinite capacity edge between vertices v_a and v_b. For each $x_i \in \Lambda_1$, add an edge from the source vertex to vertex v_{x_i} of capacity $I[i] - D[i]$, and for each $y_i \in \Lambda_2$, add an edge from v_{y_i} to the sink vertex of capacity $I[i] - D[i]$. The maximum-flow of this network is equal to the configuration volume $I - D$ if and only if D is reachable from I under Γ, and therefore reachability can be computed in polynomial time using the Edmonds-Karp maximum-flow algorithm. □

Although this algorithm only works with bipartite CRNs, we conjecture that the problem is in P and leave it as an important open question related to general matching in weighted graphs.

Conjecture 5.6. Reachability is polynomial-time solvable for CRNs with $(2, 0)$ rules.

6 Conclusion

With the complexity of the general reachability problem solved recently, this paper resolves several restricted cases of the problem. We prove hardness for several open problems or improve the known results. These include answering reachability in Population Protocols, showing that non-increasing volume CRNs are PSPACE-complete with rules of size two (improved from 5), proving that feed-forward systems are NP-complete and giving a polynomial algorithm if it is 1-consuming or 1-source without void/autogenesis rules, and showing how void and autogenesis rules affect the complexity of feed-forward systems. Additionally, we provide several other results related to these restrictions.

6.1 Open Problems

Here, we provide an overview of many of the interesting questions and directions this work brings to light, and open questions related to this paper.

Related Problems. While we give many results, this is by no means the end of investigation into the computational complexity of restricted Chemical Reaction Networks. This work can be extended in multiple ways. While this work focuses on reachability, we give results for the production and universal reachability problem. Further, classification of the complexity of these problems will help provide a better understanding of how these restrictions affect the behavior of these systems. Do the reachability and production problems ever have different complexities for the same version of the model? When is universal reachability a harder problem than reachability? What about the version of production where we want to produce k copies of a given species rather than just a single copy?

Reachability. For a complete characterization of reachability for all parameters, here we note the open problems and future directions of investigations.

- Reachability for 1-consuming *and* 1-source CRNs with the feed-forward property has membership in P. Is the feed-forward restriction required for this result or does it hold for a 1-consuming and 1-source proper CRN as well?
- We show the problem of reachability is PSPACE-complete with rules of size $(2, 2)$. Is reachability also hard with smaller rule sizes? Can we achieve the same result with $(2, 1)$ rules and a binary encoded volume? Or is our result tight?
- Production with $(1, 1)$ rules is NL-complete. However, we do not know if reachability is easy as well. Is there a polynomial time algorithm to decide unimolecular reactions?
- With a non-monotone volume we no longer have membership in PSPACE. In fact, we only have an Ackermann upper bound. With constant rule size, is reachability still Ackermann-hard?
- Reachability for $(2, 0)$ rules with the volume encoded in binary is an open problem. This is a generalized matching problem on a weighted graph. We know the problem is in P for bipartite CRNs or with unary volume. Is the problem still easy in the general case?

References

[1] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. In *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing*, PODC '04, page 290–299, New York, NY, USA, 2004. Association for Computing Machinery.

[2] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distribed Computing*, 18(4):235–253, mar 2006.

[3] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 2007.

[4] Joshua Ani, Erik D Demaine, Yevhenii Diomidov, Dylan Hendrickson, and Jayson Lynch. Traversability, reconfiguration, and reachability in the gadget framework. In *Intl. Conf. and Work. on Algorithms and Comp.*, 2022.

[5] Rutherford Aris. Prolegomena to the rational analysis of systems of chemical reactions. *Archive for Rational Mechanics and Analysis*, 19(2):81–99, jan 1965.

[6] Rutherford Aris. Prolegomena to the rational analysis of systems of chemical reactions ii. some addenda. *Archive for Rational Mechanics and Analysis*, 27(5):356–364, jan 1968.

[7] 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 *International conference on DNA-based computers*, pages 232–248. Springer, 2017.

[8] Michael Blondin, Matthias Englert, Alain Finkel, Stefan GÖller, Christoph Haase, Ranko Lazić, Pierre Mckenzie, and Patrick Totzke. The reachability problem for two-dimensional vector addition systems with states. *Journal of the ACM (JACM)*, 68(5):1–43, 2021.

[9] Adam Case, Jack H Lutz, and Donald M Stull. Reachability problems for continuous chemical reaction networks. *Natural Computing*, 17(2):223–230, 2018.

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

[11] Ho-Lin Chen, David Doty, and David Soloveichik. Rate-independent computation in continuous chemical reaction networks. In *The 5th conference on Innovations in theoretical computer science*, 2014.

[12] Matthew Cook, David Soloveichik, Erik Winfree, and Jehoshua Bruck. *Programmability of Chemical Reaction Networks*, pages 543–584. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[13] Rachel Cummings, David Doty, and David Soloveichik. Probability 1 computation with chemical reaction networks. *Natural Computing*, 15(2):245–261, 2016.

[14] Wojciech Czerwiński and Łukasz Orlikowski. Reachability in vector addition systems is ackermann-complete. In *62nd Annual Symposium on Foundations of Computer Science*, FOCS'21, pages 1229–1240. IEEE, 2021.

[15] Erik D Demaine, Isaac Grosof, Jayson Lynch, and Mikhail Rudoy. Computational complexity of motion planning of a robot through simple gadgets. In *9th Intl. Conf. on Fun with Algorithms (FUN 2018)*, 2018.

[16] Erik D Demaine, Robert A Hearn, Dylan Hendrickson, and Jayson Lynch. Pspace-completeness of reversible deterministic systems. *arXiv preprint arXiv:2207.07229*, 2022.

[17] Erik D Demaine, Dylan H Hendrickson, and Jayson Lynch. Toward a general complexity theory of motion planning: Characterizing which gadgets make games hard. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, 2020.

[18] Javier Esparza, Stefan Jaax, Mikhail Raskin, and Chana Weil-Kennedy. The complexity of verifying population protocols. *Distributed Computing*, 34(2):133–177, 2021.

[19] Bin Fu, Timothy Gomez, Elise Grizzell, Andrew Rodriguez, Robert Schweller, and Tim Wylie. Reachability in population protocols. In *The Japan Conference on Discrete and Computational Geometry, Graphs, Games (JCDCG^3)*, 2022.

[20] John Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135–159, 1979.

[21] Robert F Johnson and Lulu Qian. Simplifying chemical reaction network implementations with two-stranded dna building blocks. In *26th International Conference on DNA Computing and Molecular Programming (DNA 26)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

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

[23] Matthew R Lakin, David Parker, Luca Cardelli, Marta Kwiatkowska, and Andrew Phillips. Design and analysis of dna strand displacement devices using probabilistic model checking. *Journal of the Royal Society Interface*, 9(72):1470–1485, 2012.

[24] Jérôme Leroux. The reachability problem for petri nets is not primitive recursive. In *62nd Annual Symposium on Foundations of Computer Science*, FOCS'21. IEEE, 2021.

[25] Richard J. Lipton. The reachability problem requires exponential space. Technical Report 62, Yale University, 1976.

[26] Ernst W. Mayr. An algorithm for the general petri net reachability problem. In *Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*, STOC '81, page 238–246, New York, NY, USA, 1981. Association for Computing Machinery.

[27] Carl Adam Petri. *Kommunikation mit Automaten*. PhD thesis, Rheinisch-Westfälischen Institutes für Instrumentelle Mathematik an der Universität Bonn, 1962.

[28] J Plesník. The np-completeness of the hamiltonian cycle problem in planar diagraphs with degree bound two. *Information Processing Letters*, 1979.

[29] Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978.

[30] Sylvain Schmitz. The complexity of reachability in vector addition systems. *ACM SigLog News*, 2016.

[31] David Soloveichik, Matthew Cook, Erik Winfree, and Jehoshua Bruck. Computation with finite stochastic chemical reaction networks. *natural computing*, 7(4):615–633, 2008.

[32] 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.

[33] Chris Thachuk and Anne Condon. Space and energy efficient computation with dna strand displacement systems. In *International Workshop on DNA-Based Computers*, 2012.

[34] Anupama J Thubagere, Chris Thachuk, Joseph Berleant, Robert F Johnson, Diana A Ardelean, Kevin M Cherry, and Lulu Qian. Compiler-aided systematic construction of large-scale dna strand displacement circuits using unpurified components. *Nature Communications*, 8(1):1–12, 2017.

[35] Marko Vasić, Cameron Chalk, Austin Luchsinger, Sarfraz Khurshid, and David Soloveichik. Programming and training rate-independent chemical reaction networks. *Proceedings of the National Academy of Sciences*, 119(24):e2111552119, 2022.

[36] Boya Wang, Chris Thachuk, Andrew D Ellington, Erik Winfree, and David Soloveichik. Effective design principles for leakless strand displacement systems. *Proceedings of the National Academy of Sciences*, 115(52):E12182–E12191, 2018.

[37] Gianluigi Zavattaro and Luca Cardelli. Termination problems in chemical kinetics. In *International Conference on Concurrency Theory*, pages 477–491. Springer, 2008.

[38] David Yu Zhang and Georg Seelig. Dynamic dna nanotechnology using strand-displacement reactions. *Nature chemistry*, 3(2):103–113, 2011.