

Reachability with Restricted Reactions in Inhibitory Chemical Reaction Networks

Divya Bajaj ✉

University of Texas Rio Grande Valley, TX, USA

Ryan Knobel ✉

University of Texas Rio Grande Valley, TX, USA

Aiden Massie ✉

University of Texas Rio Grande Valley, TX, USA

Ramiro Santos ✉

University of Texas Rio Grande Valley, TX, USA

Evan Tomai ✉

University of Texas Rio Grande Valley, TX, USA

Bin Fu ✉

University of Texas Rio Grande Valley, TX, USA

Austin Luchsinger ✉

University of Texas Rio Grande Valley, TX, USA

Pablo Santos ✉

University of Texas Rio Grande Valley, TX, USA

Robert Schweller ✉

University of Texas Rio Grande Valley, TX, USA

Tim Wylie ✉

University of Texas Rio Grande Valley, TX, USA

Abstract

Chemical Reaction Networks (CRNs) are a well-established model of distributed computing characterized by quantities of molecular species that can transform or change through applications of reactions. A fundamental problem in CRNs is the reachability problem, which asks if an initial configuration of species can transition to a target configuration through an applicable sequence of reactions. It is well-known that the reachability problem in *general* CRNs was recently proven to be Ackermann-complete. However, if the CRN's reactions are restricted in both power, such as only deleting species (deletion-only rules) or consuming and producing an equal number of species (volume-preserving rules), *and* size (unimolecular or bimolecular rules), then reachability falls below Ackermann-completeness, and is even solvable in polynomial time for deletion-only systems.

In this paper, we investigate reachability under this set of restricted unimolecular and bimolecular reactions, but in the Priority-Inhibitory CRN and Inhibitory CRN models. These models extend a traditional CRN by allowing some reactions to be inhibited from firing in a configuration if certain species are present; the exact inhibition behavior varies between the models. We first show that reachability with Priority iCRNs mostly remains in P for deletion-only systems, but becomes NP-complete for one case. We then show that reachability with deletion-only reactions for iCRNs is mostly NP-complete, and PSPACE-complete even for (1, 1)-size (general) reactions. We also provide FPT algorithms for solving most of the reachability problems for the iCRN model. Finally, we show reachability for CRNs with states is already NP-hard for the simplest deletion-only systems, and is PSPACE-complete even for (general) (1, 1)-size reactions.

2012 ACM Subject Classification Theory of computation → Models of computation; Theory of computation → Complexity classes

Keywords and phrases Chemical Reaction Networks, Vector Addition Systems, Petri-nets, Reachability, Inhibitors, Void Reactions

Digital Object Identifier 10.4230/LIPIcs.SWAT.2026.3

Related Version *Full Version:* <https://doi.org/10.48550/arXiv.2604.17510>

Funding This research was supported in part by National Science Foundation Grant CCF-2329918.

1 Introduction

Chemical Reaction Networks (CRNs) [4, 5] are an abstract model of molecular computation in which molecular *species* can evolve through applications of *reactions*. Although initially motivated by modeling natural chemical interactions, CRNs have been recently explored for



© Divya Bajaj, Bin Fu, Ryan Knobel, Austin Luchsinger, Aiden Massie, Pablo Santos, Ramiro Santos, Robert Schweller, Evan Tomai, and Tim Wylie;

licensed under Creative Commons License CC-BY 4.0

20th Scandinavian Symposium on Algorithm Theory (SWAT 2026).

Editor: Pierre Fraigniaud; Article No. 3; pp. 3:1–3:18



Leibniz International Proceedings in Informatics

LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

developing artificial chemical systems [23], computations of semilinear functions [8], digital logic [7, 16, 17], and neural networks [16], along with successful implementations in DNA Strand Displacement (DSD) systems [28].

CRNs are computationally equivalent to two classical models of infinite-state systems: Petri-nets [25] and Vector Addition Systems (VAS) [18, 19]. This equivalence connects molecular computation with decades of complexity-theoretic results concerning reachability [9, 14]. In all three models, the *reachability problem* asks whether a target configuration $\vec{C}_t$ can be reached from a given initial configuration $\vec{C}_s$ via a sequence of rule applications. Although certain characteristics of reachability in general CRNs (and equivalently, Petri-nets and VAS) were well-known for decades, such as EXPSpace-hardness [21] and decidability [22], only recently has the problem been proven to be Ackermann-complete [10, 20].

A crucial limitation of the classical models is the absence of *zero-checking*. To address this, a natural and well-studied extension across all three models is to allow *inhibition*, which allows certain reactions to be disabled by the presence of designated species. Petri-nets with inhibitor arcs [15, 26], Priority Vector Addition Systems [13], and Inhibitory Chemical Reaction Networks [6] all explore this idea. Because inhibition enables zero-checking, these inhibitory extensions gain substantial power, achieving Turing Universality [6, 24]. However, due to the Turing-universal power of zero-checking, the reachability problem becomes undecidable in general for such systems with inhibition. For this reason, and to better understand the boundaries between decidable/tractable and undecidable/intractable, we limit our study to restricted classes of CRNs with inhibition that use only deletion-only or volume-preserving rules.

For CRNs, *deletion-only* (void) reactions never create any new species in the system (e.g., $\lambda_1 + \lambda_2 \rightarrow \emptyset$, or $\lambda_1 + \lambda_2 \rightarrow \lambda_1$), and *volume-preserving* reactions contain an equal number of reactants and products, ensuring that the system’s volume never changes (e.g., $\lambda_1 + \lambda_2 \rightarrow \lambda_3 + \lambda_4$). These restrictions are well-motivated: unlike general CRN systems, which can exhibit complex and error-prone behaviors that are challenging to implement [29], deletion-only and volume-preserving systems are potentially simpler to realize and analyze [1, 12]. Moreover, in the absence of inhibition, such restrictions significantly reduce computational power: deletion-only systems cannot compute even simple functions [2], reachability with bimolecular void reactions is solvable in polynomial time [1, 12], and reachability with (1,1)-size reactions is solvable in polynomial time [1]. These observations raise a central question: to what extent do inhibitory extensions increase the computational power of CRNs when restricted to these small-sized restricted reactions?

Under these structural restrictions, we consider two types of inhibitory CRN systems. The general *Inhibitory* CRN (iCRN) [6] model simply allows reaction execution to be inhibited by the presence of some subset of species in the configuration. The *Priority-Inhibitory* CRN (P-iCRN) model establishes a priority for each reaction, in which a reaction with priority k is inhibited from firing unless the first k species are absent from the system. Here, the order of species is used to determine the inhibitors for each reaction. In this paper, we study the reachability problem under deletion-only and volume-preserving reactions for both of these models.

Our Results

For the restricted reaction types studied in this paper, we use the notation (r, p) to denote reactions with r reactants and p products (e.g., (2,0) is a bimolecular void reaction and (1,1) is a volume-preserving unimolecular reaction). For reference, in a traditional CRN model,

■ **Table 1** Reachability with Inhibition for restricted rule types. Void systems that are NP-complete in regular CRNs, $(k, k - i)$ where $k \geq 3$ and $i \geq 2$, are not listed since they are still NP-hard with inhibitions. Cells highlighted in blue link to interesting open directions (in Section 5) that extend the results in the table.

Reachability under Inhibition						
Rule Size	CRN	Ref.	P-iCRN	Ref.	iCRN	Ref.
Deletion-Only Systems						
(1, 0)	P	-	P	Obs. 7	P	Obs. 23
(2, 0)	P	[1]	P	Thm. 15	NPC FPT	Thm. 24, 30
(2, 1)	P	[12]	P	Thm. 17	NPC FPT	Thm. 31, 33
$(k, k - 1)$	P	[12]	P	Thm. 17	NPC FPT	Thm. 32, 33
$(2, 0), (2, 1)$	P	[12]	NPC	Thm. 20	NPC	Thm. 34
General Systems						
(1, 1)	P, NL-hard	[1]	Open		PSPACE-c	Thm. 35
(2, 2)	PSPACE-c	[11]	PSPACE-c	Obs. 22	PSPACE-c	Obs. 37
General	Ack-c	[10, 20]	Decidable	[13, 26]	Undecidable	[6, 26]

reachability is in P for all of our considered void reactions [1, 12], NL-hard for (1, 1)-size reactions [1], PSPACE-complete for (2, 2)-size reactions [11], and Ackermann-complete for general systems [10, 20].

As for organization, we first provide formal descriptions of all CRN models studied in this paper and some additional preliminaries in Section 2. We then study reachability under the Priority iCRN model in Section 3. Here, we prove that reachability with our considered void reactions is mostly in P, but becomes NP-complete for $(2, 0) + (2, 1)$ -size reactions. Section 4 looks at reachability for iCRNs. We show that reachability becomes NP-complete for all void reactions except for size (1, 0), and PSPACE-complete for (1, 1) and (2, 2)-size reactions. We additionally provide FPT algorithms for solving reachability with void reactions of sizes $(2, 0)$, $(2, 1)$, and $(k, k - 1)$. The complete scope of our reachability results is shown in Table 1. Finally, in Section 5, we conclude with a discussion on some open problems for resolving complexity gaps of Table 1 and future potential work concerning CRNs using states as an unconventional yet powerful form of inhibition.

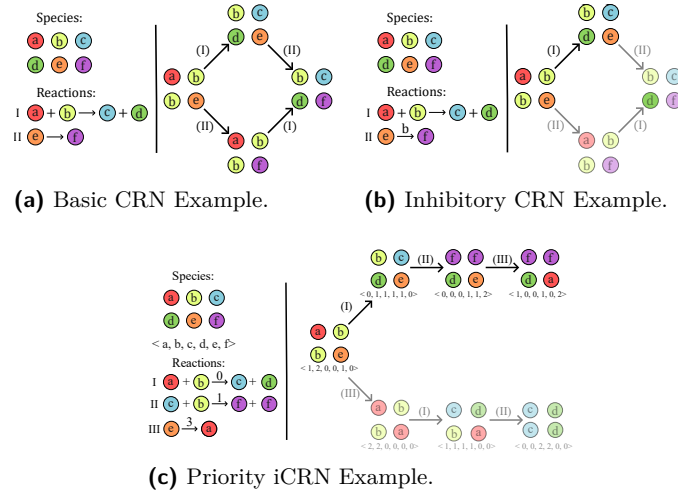
2 Preliminaries

We define the three chemical reaction network models considered in this paper: the basic CRN model (Section 2.1), the Inhibitory CRN model (Section 2.1.1), and the Priority iCRN model (Section 2.2.1).

2.1 Chemical Reaction Networks

A *chemical reaction network* (CRN) $\mathcal{C} = (\Lambda, \Gamma)$ is defined by a finite set of species Λ , and a finite set of reactions Γ where each reaction is a pair $(\vec{R}, \vec{P}) \in \mathbb{N}^\Lambda \times \mathbb{N}^\Lambda$, sometimes written $\vec{R} \longrightarrow \vec{P}$, that denotes the *reactant* species consumed by the reaction and the *product* species generated by the reaction. For instance, given $\Lambda = \{a, b, c\}$, the reaction $((2, 0, 0), (0, 1, 1))$ represents $2a \longrightarrow b + c$; 2 a species are removed, and 1 new b and c species are created. See Figure 1a for an example.

A *configuration* $\vec{C} \in \mathbb{N}^{|\Lambda|}$ of a CRN assigns integer counts to every species $\lambda \in \Lambda$, and we use notation $\vec{C}[\lambda]$ to denote that count. For a species $\lambda \in \Lambda$, we denote the configuration consisting of a single copy of λ and no other species as $\vec{\lambda}$. It is often useful to reference the



■ **Figure 1** Example systems of the CRN models studied in this paper. Each arrow represents an application of a reaction. Transparent arrows represent invalid reaction applications. (a) Basic CRN. Observe how, despite two different reaction sequences being applicable in the initial configuration, both lead to the same terminal configuration. (b) Inhibitory CRN. Because Reaction 2 is inhibited by the species b , it cannot be applied in this system. (c) Priority iCRN. Reaction 3 is inhibited by species a and b , since both species are in the system, it's not an applicable reaction.

set of species whose counts are not zero in a given configuration. In such cases, the notation $\{\vec{C}\}$ is used. Formally, $\{\vec{C}\} = \{\lambda \in \Lambda \mid \vec{C}[\lambda] > 0\}$, and when convenient and clear from the context, we further use $\{\vec{C}\}$ to denote the configuration (vector) representation in which each element has a single copy. Finally, let $|\vec{C}| = \sum_{\lambda \in \Lambda} C[\lambda]$ denote the total number of copies of all species in a configuration, sometimes referred to as the *volume* of $\vec{C}$.

A reaction $(\vec{R}, \vec{P})$ is said to be *applicable* in configuration $\vec{C}$ if $\vec{R} \leq \vec{C}$; in other words, a reaction is applicable if $\vec{C}$ has at least as many copies of each species as $\vec{R}$. If the reaction $(\vec{R}, \vec{P})$ is applicable, it results in configuration $\vec{C}' = \vec{C} - \vec{R} + \vec{P}$ if it occurs, and we write $\vec{C} \xrightarrow{(\Lambda, \Gamma)}_{crn} \vec{C}'$, or simply $\vec{C} \rightarrow \vec{C}'$ when the model and CRN are clear from context.

A *void* reaction is any rule that does not create any new copies of any species type, and can only delete or preserve existing species copies. Thus, a void reaction either has no products or has products that are a subset of its reactants (in which case these products are termed *catalysts*).

► **Definition 1** (Discrete Chemical Reaction Network). 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 .

► **Definition 2** (Basic CRN Dynamics). For a CRN (Λ, Γ) and configurations $\vec{A}$ and $\vec{B}$, we say that $\vec{A} \xrightarrow{(\Lambda, \Gamma)}_{crn} \vec{B}$ if there exists a rule $(\vec{R}, \vec{P}) \in \Gamma$ such that $\vec{R} \leq \vec{A}$, and $\vec{A} - \vec{R} + \vec{P} = \vec{B}$.

If there exists a finite sequence of configurations such that $\vec{C} \rightarrow \vec{C}_1 \rightarrow \dots \rightarrow \vec{C}_n \rightarrow \vec{D}$, then we say that $\vec{D}$ is *reachable* from $\vec{C}$ and we write $\vec{C} \rightsquigarrow \vec{D}$. A configuration is said to be *terminal* if no reactions are applicable. We also define an *initial configuration* for a CRN as its starting configuration. A *CRN System* T is then defined as a pair of a CRN model and its initial configuration.

The following sections define extensions of the basic CRN model by way of defining modified dynamics. For each model, assume the concepts of reachability and terminality are derived from the dynamics in the same manner as for the basic CRN model.

2.1.1 Inhibitory CRNs

A reaction γ is said to be *inhibited* by a species λ when the reaction γ may only be applied if λ is absent in the system. We define an inhibitor mapping $\mathcal{I} : \Gamma \rightarrow \mathbb{P}(\Lambda)$ that maps a reaction to a subset of species that inhibit the reaction. An *Inhibitory CRN* $\mathcal{C}_{\mathcal{IC}} = ((\Lambda, \Gamma), \mathcal{I})$ as defined by [6] is then a basic CRN along with the mapping $\mathcal{I}$. See Figure 1b for an example.

► **Definition 3** (Inhibitory Dynamics). *For a Inhibitory CRN $((\Lambda, \Gamma), \mathcal{I})$ and configurations $\vec{A}$ and $\vec{B}$, we say that $\vec{A} \xrightarrow{\mathcal{C}_{\mathcal{IC}}} \vec{B}$ if there exists a rule $\gamma = (\vec{R}, \vec{P}) \in \Gamma$ such that $\vec{R} \leq \vec{A}$, $\vec{A} - \vec{R} + \vec{P} = \vec{B}$, and $A[\lambda] = 0, \forall \lambda \in \mathcal{I}(\gamma)$.*

2.2 Vector Addition Systems

A d -dimensional *Vector Addition System* (VAS) $\mathcal{V} = (\vec{C}_0, \mathcal{T})$ is defined by [19] as an initial configuration vector $\vec{C}_0 \in \mathbb{Z}_{\geq 0}^d$, and a finite set of transitions $\mathcal{T} \subseteq \mathbb{Z}^d$.

► **Definition 4** (VAS Dynamics). *For a VAS $(\vec{C}_s, \mathcal{T})$ and configurations $\vec{A}$ and $\vec{B}$, we say $\vec{A} \rightarrow_{\mathcal{V}} \vec{B}$ if there exists a transition $\vec{y} \in \mathcal{T}$ such that $\vec{A} + \vec{y} = \vec{B}$.*

The concept of priority on transitions in a Vector Addition System is defined in [13]. Each transition in a d -dimensional *Priority Vector Addition System* (PVAS) is mapped to an integer in $[0, d]$. For a system in configuration $\vec{A}$, a transition $\vec{y}$ with priority $p \leq d$ is applicable iff $\forall i \leq p, \vec{A}[i] = 0$ and $\vec{A} + \vec{y} \in \mathbb{Z}_{\geq 0}^d$.

The correspondence between Vector Addition Systems and Chemical Reaction Networks is direct, especially in the absence of catalysts. A transition vector $\vec{y}$ in a VAS can be written as a reaction $(\vec{R}, \vec{P})$ in CRN where

$$(\vec{R}[\lambda_i], \vec{P}[\lambda_i]) = \begin{cases} (|\vec{y}[i]|, 0) & \text{if } \vec{y}[i] < 0 \\ (0, \vec{y}[i]) & \text{if } \vec{y}[i] > 0 \\ (0, 0) & \text{otherwise} \end{cases}$$

Since PVAS are essentially VAS with prioritized inhibition on transitions, it is natural to compare this model to iCRNs, which have unprioritized inhibition on reactions. While Vector Addition Systems are not capable of *catalytic* transitions, it is known that a catalytic reaction in a CRN can be replaced by two non-catalytic reactions that use a unique intermediate species. Therefore, Vector Addition Systems are equivalent to Chemical Reaction Networks in general. However, this equivalence may break down under restricted settings such as deletion-only or small-size rules. This observation motivates a reformulation of PVAS as Priority iCRN in order to compare these results to inhibitory CRNs without priority.

2.2.1 Priority iCRN

We define a *priority* mapping $\mathcal{P} : \Gamma \rightarrow [0, |\Lambda|]$ that maps a reaction to an integer in $[0, |\Lambda|]$. A *Priority iCRN* $\mathcal{C}_{\mathcal{PI}} = ((\Lambda, \Gamma), \mathcal{P})$ is defined as a basic CRN along with mapping $\mathcal{P}$. Because the priority relies on the ordering of species, we make our set of species Λ in a Priority iCRN an ordered set. See Figure 1c for an example.

► **Definition 5** (Priority Inhibitory Dynamics). *For a Priority iCRN $((\Lambda, \Gamma), \mathcal{P})$ and configurations $\vec{A}$ and $\vec{B}$, we say that $\vec{A} \xrightarrow{\mathcal{C}_{\mathcal{PI}}} \vec{B}$ if there exists a rule $\gamma = (\vec{R}, \vec{P}) \in \Gamma$ such that $\vec{R} \leq \vec{A}$, $\vec{A} - \vec{R} + \vec{P} = \vec{B}$, and $A[\lambda_i] = 0, \forall i \in [1, \mathcal{P}(\gamma)]$.*

2.3 Reachability

► **Definition 6** (Reachability Problem). *Given any CRN System with a species set Λ and a reaction set Γ , an initial configuration $\vec{C}_s$ and a destination (target) configuration $\vec{C}_t$: is $\vec{C}_t$ reachable from $\vec{C}_s$ following the dynamics of the model.*

We also consider a special variant of reachability referred to as the *empty configuration reachability* problem. Here, the destination configuration is always the empty configuration $\vec{0}$, where all species have a count of zero. Thus, the problem can be re-interpreted as asking if there exists a sequence of reactions that can delete all present species in $\vec{C}_s$.

3 Priority Inhibitory CRNs

We start our analysis of reachability in inhibitory systems with Priority iCRNs. We first show that the problem does not leave P with size (2, 0) or (2, 1) reactions. However, when considering systems with both (2, 0) and (2, 1) size rules, we show that the problem becomes NP-complete, even with a maximum priority of 1.

Priority iCRNs with Only Void Rules

We state the following observation for completeness.

► **Observation 7.** *Given a Priority iCRN $\mathcal{C}_{PI} = ((\Lambda, \Gamma), \mathcal{P})$ with only void rules of size (1, 0), the reachability problem is solvable in $\mathcal{O}(|\Lambda| + |\Gamma|)$ time.*

Proof. Given a Priority iCRN, start configuration $\vec{C}_s$ and target configuration $\vec{C}_t$:

1. For any species λ_i , if $\vec{C}_t[\lambda_i] \neq 0$ then remove all reactions γ_j such that $\mathcal{P}(\gamma_j) \geq i$. Since we only consider void rules, such reactions will never be applicable in any configuration along any path from $\vec{C}_s$ through $\vec{C}_t$.
2. For $i = 1$ to $|\Lambda|$, if there exists a reaction of form $\lambda_i \rightarrow \phi$:
 - a. then apply the reaction $\vec{C}_s[\lambda_i] - \vec{C}_t[\lambda_i]$ times.
 - b. else, skip to step (3).
3. If the final configuration is reached, return “yes”, otherwise return “no”. ◀

Void Rules of Size (2, 0). We now show that the reachability problem in Priority iCRNs with (2, 0) reactions is also solvable in polynomial time. To do so, we start by showing that the general reachability problem for a given Priority iCRN with (2, 0) void rules can be reduced to the empty reachability problem in the given CRN in Lemmas 8, 9, and 10. We then reduce the empty configuration reachability problem in the Priority iCRN to the known maximum b -matching problem, which is solvable in polynomial time [3], in Lemma 12.

Given a (2, 0) void Priority iCRN system $((\Lambda, \Gamma), \mathcal{P})$ such that $k = \max_{\gamma \in \Gamma} \mathcal{P}(\gamma)$. We can partition the set of species Λ into set of inhibiting species $\{\lambda_1, \dots, \lambda_k\}$ and set of non-inhibiting species $\{\lambda_{k+1}, \dots, \lambda_{|\Lambda|}\}$.

► **Lemma 8.** *Given two configurations $\vec{C}_s$ and $\vec{C}_t$ in a (2, 0) void Priority iCRN $\mathcal{C}_{PI}$, $\vec{C}_s \rightsquigarrow \vec{C}_t \iff \vec{C}_s - \vec{C}_t \rightsquigarrow \vec{0}$, if $\forall i \in [1, k], \vec{C}_t[\lambda_i] = 0$.*

Proof. Given that $\forall i \leq k, \vec{C}_t[\lambda_i] = 0$, $\vec{C}_s[\lambda_i] - \vec{C}_t[\lambda_i] = \vec{C}_s[\lambda_i]$. Therefore, all inhibited reactions will be applicable at some point.

Since void rules only delete species, and do not add new counts for species, for all non-inhibiting species λ_j with $j > k$, if $\vec{C}_t[\lambda_j]$ is reachable from $\vec{C}_s[\lambda_j]$, then $\vec{0}[\lambda_j]$ is reachable from $\vec{C}_s[\lambda_j] - \vec{C}_t[\lambda_j]$. ◀

► **Lemma 9.** *Given two configurations $\vec{C}_s$ and $\vec{C}_t$ in a $(2, 0)$ void Priority iCRN $\mathcal{C}_{\mathcal{PT}}$, $\vec{C}_s \rightsquigarrow \vec{C}_t \iff \vec{C}_s - \vec{C}_t \rightsquigarrow \vec{0}$, if $\exists i \leq k, \vec{C}_t[\lambda_i] \neq 0$.*

Proof. We modify the given Priority iCRN $\mathcal{C}_{\mathcal{PT}}$ system as follows. We remove all reactions $\gamma \in \Gamma$ such that $\mathcal{P}(\gamma) \geq i, \forall i \leq k$ if $\vec{C}_t[\lambda_i] \neq 0$. This is because such reactions will never be applicable as they will always be inhibited. Due to this modification, the species $\lambda_i, \dots, \lambda_k$ no longer permanently act as inhibitors. The new reaction set Γ' contains the remaining inhibited reactions with priority greater than 0. We now compute k' over Γ' as $k' = \max_{\gamma \in \Gamma'} \mathcal{P}(\gamma)$. We partition the set Λ again into set of inhibiting species $\{\lambda_1, \dots, \lambda_{k'}\}$ and set of non-inhibiting species $\{\lambda_{k'+1}, \dots, \lambda_{|\Lambda|}\}$. From Lemma 8, $\vec{C}_s \rightsquigarrow \vec{C}_t \iff \vec{C}_s - \vec{C}_t \rightsquigarrow \vec{0}$ in a Priority iCRN $\mathcal{C}_{\mathcal{PT}} = ((\Lambda, \Gamma'), \mathcal{P})$. ◀

► **Lemma 10.** *Given two configurations $\vec{C}_s$ and $\vec{C}_t$ in a $(2, 0)$ void Priority iCRN $\mathcal{C}_{\mathcal{PT}}$, $\vec{C}_s \rightsquigarrow \vec{C}_t \iff \vec{C}_s - \vec{C}_t \rightsquigarrow \vec{0}$.*

Proof. Follows from Lemmas 8 and 9. ◀

► **Definition 11** (*b-matching*). *Given a graph $G = (V, E)$ and some edge capacity function $u : E \rightarrow \mathbb{N} \cup \{\infty\}$, a b -value function $b : V \rightarrow \mathbb{N}$, and a $\delta(v)$ function that returns the set of incident edges of vertex v , find a maximum assignment $f : E \rightarrow \mathbb{N}$ s.t. $f(e) \leq u(e)$ for all $e \in E$ and $\sum_{e \in \delta(v)} f(e) \leq b(v)$ for all $v \in V$. We call it a perfect b -matching if $\sum_{e \in \delta(v)} f(e) = b(v)$ for all $v \in V$.*

The runtime of the maximum b -matching problem is strictly polynomial and runs in $O(|V|^2 \log(|V|)(|E| + |V| \log(|V|)))$ [3]. We use this runtime to solve the reachability problem for Priority iCRNs with only $(2, 0)$ -size reactions in polynomial time.

► **Lemma 12.** *Given a Priority iCRN $\mathcal{C}_{\mathcal{PT}} = ((\Lambda, \Gamma), \mathcal{P})$ with only void rules of size $(2, 0)$, the empty reachability problem is decidable in $O(|\Lambda|^2 \log(|\Lambda|)(|\Gamma| + |\Lambda| \log(|\Lambda|)))$ time.*

Proof. To solve the empty-configuration reachability problem in $(2, 0)$ void Priority iCRN, we reduce it to an instance of the maximum b -matching problem.

Given a Priority iCRN $\mathcal{C}_{\mathcal{PT}} = ((\Lambda, \Gamma), \mathcal{P})$ with starting configuration $\vec{C}_s$, we construct a b -matching instance by building a graph $G = (V, E)$ as follows. Here we assume that the given Priority iCRN only includes reactions $\gamma : \lambda_i + \lambda_j \rightarrow \phi$ where $i, j > \mathcal{P}(\gamma)$; any self-inhibiting reactions that exist (i or j is less than the priority) are ignored. We turn every species $\lambda_i \in \Lambda$ into a vertex $v_i \in V$ and set the b -values for each of the vertices $b(v_i) = \vec{C}_s[\lambda_i]$. For each $\gamma : \lambda_i + \lambda_j \rightarrow \phi$ in Γ such that $i \neq j$, add an undirected edge $e = (v_i, v_j)$ in E . For any reaction of the form $\lambda_i + \lambda_i \rightarrow \phi$, add vertices v_i^1, v_i^2 to the set V and set the b -values $b(v_i^1) = b(v_i^2) = \left\lfloor \frac{b(v_i)}{2} \right\rfloor$. We also add undirected edges $(v_i, v_i^1), (v_i, v_i^2), (v_i^1, v_i^2)$ in E .

From the rules above, we have the graph G well defined. We now present the following claims regarding G .

▷ **Claim 13.** If $\vec{C}_s$ can reach the empty configuration, then there exists a perfect b -matching in the graph G .

This is straightforward for the case when applying a reaction of the form $\lambda_i + \lambda_j \rightarrow \phi$ such that $i \neq j$. The reaction application results in the removal of one copy of species λ_i and λ_j . This corresponds to an edge matching of (v_i, v_j) on the graph G according to the construction above. We reduce the values of $b(v_i)$ and $b(v_j)$ by one, and the pair (v_i, v_j) form a perfect matching. For the case when applying a reaction of form $\lambda_i + \lambda_i \rightarrow \phi$, two copies of λ_i are deleted from the configuration. This rule application corresponds to selecting edges (v_i, v_i^1)

and (v_i, v_i^2) . We reduce the counts of $b(v_i^1)$ and $b(v_i^2)$ by one, and that of $b(v_i)$ by two. The remaining b -values of vertices v_i^1 and v_i^2 are reduced by selecting edge (v_i^1, v_i^2) . And the pairs (v_i, v_i^1) , (v_i, v_i^2) , and (v_i^1, v_i^2) form a perfect matching.

▷ **Claim 14.** If there exists a perfect b -matching in the graph G , then $\vec{C}_s$ can reach the empty configuration.

If there exists a perfect b -matching for the graph G , there exists an assignment $f : E \rightarrow \mathbb{N}$ such that $\sum_{e \in \delta(v)} f(e) = b(v)$ for all $v \in V$. From the construction of the graph, all edges $(v_i, v_j) \in E$ such that $i \neq j$ correspond to a reaction $\lambda_i + \lambda_j \rightarrow \phi$ in Γ in the Priority iCRN. And, edges of the form $(v_i, v_i^1), (v_i, v_i^2) \in E$ correspond to a reaction $\lambda_i + \lambda_i \rightarrow \phi$ in Γ . We argue that, given any perfect b -matching, we can apply a corresponding set of reactions in a rearranged order based on their priority to reach the empty configuration.

Let M contain the set of edges to be a perfect b -matching for G . We apply reactions in Priority iCRN corresponding to the edges in M as follows.

1. Apply all reactions $\gamma = \lambda_i + \lambda_j \rightarrow \phi$ corresponding to edges $e_k = (v_i, v_j)$, $f(e_k)$ times, if $e_k \in M$, $i \neq j$ and $\mathcal{P}(\gamma) = 0$. All such reactions are uninhibited and can be applied.
2. Apply all reactions $\gamma = \lambda_i + \lambda_i \rightarrow \phi$ corresponding to edges $e_k = (v_i, v_i^1)$ and $e_\ell = (v_i, v_i^2)$, $f(e_k)$ times, if $e_k, e_\ell \in M$ and $\mathcal{P}(\gamma) = 0$. The vertices v_i^1 and v_i^2 are only connected to each other and to vertex v_i . To achieve perfect b -matching for vertices v_i^1 and v_i^2 , assigned values $f(e_k)$ and $f(e_\ell)$ must be exactly equal, $f(e_k) = f(e_\ell) = b(v_i^1) - f(e_m)$ where $e_m = (v_i^1, v_i^2)$. Applying such a reaction will delete $2 * f(e_k)$ copies of species λ_i .
3. For all p in 1 through $\max_{\gamma \in \Gamma} \mathcal{P}(\gamma)$:
 - a. apply all reactions $\gamma = \lambda_i + \lambda_j \rightarrow \phi$ corresponding to edges $e_k = (v_i, v_j)$, $f(e_k)$ times, if $e_k \in M$, $i \neq j$ and $\mathcal{P}(\gamma) = p$.
 - b. apply all reactions $\gamma = \lambda_i + \lambda_i \rightarrow \phi$ corresponding to edges $e_k = (v_i, v_i^1)$ and $e_\ell = (v_i, v_i^2)$, $f(e_k)$ times, if $e_k, e_\ell \in M$, and $\mathcal{P}(\gamma) = p$.

Furthermore, the reactions applied in each iteration in Step 3 are guaranteed to be applicable (absence of inhibitors and sufficient count of reactants) because (1) all species count go to zero; (2) no reaction that uses a reactant λ_i has a priority $p > i$; and (3) each reaction e_k is only applied $f(e_k)$ times. If multiple reactions use a species λ_i , the total applications of all those reactions $\sum_{e \in \delta(v_i)} f(e)$ will be exactly $b(v_i)$. Therefore, rearranging applications of each of those reactions will not affect the availability of reactants in later applications. ◀

► **Theorem 15.** Given a Priority iCRN $\mathcal{C}_{PI} = ((\Lambda, \Gamma), \mathcal{P})$ with only void rules of size $(2, 0)$, the reachability problem is decidable in $O(|\Lambda|^2 \log(|\Lambda|)(|\Gamma| + |\Lambda| \log(|\Lambda|)))$ time.

Proof. Given a Priority iCRN with starting configurations $\vec{C}_s$ and target configuration $\vec{C}_t$, create the configuration $\vec{C}_s - \vec{C}_t$. From Lemma 12, empty configuration reachability is decidable from $\vec{C}_s - \vec{C}_t$ in polynomial time. Finally, from Lemma 10, empty configuration reachability in $(2, 0)$ Priority iCRN implies reachability. ◀

► **Lemma 16.** Given a CRN $\mathcal{C} = (\Lambda, \Gamma)$ with only void rules of size $(k, k-1)$, the reachability problem is decidable in $\mathcal{O}(|\Lambda|^2 |\Gamma|)$ time (Complete proof in [12]).

Proof (Sketch). The dynamic programming algorithm uses a $|\Lambda| \times (|\Lambda| + 1)$ table D of boolean entries, where each row represents a different species. Since each $(k, k-1)$ catalytic void reaction reduces the count of exactly one species, only one of the reactions need to be applied at most $\vec{C}_s[\lambda_i] - \vec{C}_t[\lambda_i]$ times to reduce the count of a species λ_i . Hence, only $|\Lambda|$ distinct reactions need to be considered to reach the target configurations, if possible. The total number of rule applications, however, will be polynomial in the volume of the system.

A bottom-up approach is used to determine the ordering of the species in which their count needs to be reduced. This is essential since a species λ_i shouldn't go to zero before it is used as a catalyst in other reactions that need to be applicable. The algorithm starts by filling first column of D with 1 for species who have reached their target count. It then progresses column-by-column, applying reactions based on the ordering on species. If the last column has all 1s, then the configuration is reached. ◀

► **Theorem 17.** *Given a Priority iCRN $\mathcal{C}_{\mathcal{PI}} = ((\Lambda, \Gamma), \mathcal{P})$ with only void rules of size $(k, k - 1)$, the reachability problem is decidable in $\mathcal{O}(|\Lambda|^2|\Gamma|)$ time.*

Proof. This follows as a modification of the dynamic programming algorithm for general CRNs with only void rules of size $(k, k - 1)$ from Lemma 16. Let $\vec{C}_f$ denote the target configuration, with $\lambda_1, \dots, \lambda_{|\Lambda|}$ denoting the order assigned to the species. Remove all rules of the form $\lambda_j + (\cdot) \rightarrow^k (\cdot)$, where $j < k$. The rest of the system remains the same.

Start with a counter $l = |\Lambda|$. Construct an $|\Lambda| \times (|\Lambda| + 1)$ table $D(s, j)$ of boolean entries, where each row represents a different species. Reduce the count of each species to $\max(k, s^f)$, where s^f represents the final count of species s , if there exists a rule that can do so. Starting from the first column $j = 0$, place a 1 if the respective species is already in its final count. Then, for each entry $D(s, j)$, place a 1 if $D(s, j - 1)$ is a 1, or if there exists a reaction $\gamma \in \Gamma$ for $s \in \Lambda$ that reduces s to its final count such that 1) $\vec{C}_f[s] > 0$ or 2) $s = \lambda_l$, where all the reactants of γ have either reached their final counts or will not prevent the reaction from occurring once they do. Any time species λ_l has found a reaction to reach a final count 0, set $l = l - 1$. If column $|\Lambda| + 1$ contains all 1's, then reachability is possible. Otherwise, it is not.

The algorithm follows the same bottom-up approach as for general CRNs. The main difference is that now there exists a predetermined ordering that species must go to zero for other rules to be applicable. Thus, any species λ_j such that $j < l$ is not considered until species λ_l has found a suitable reaction and reaches a count of zero. The proof then follows from general CRNs. ◀

► **Lemma 18** (Rearrangement Lemma for Void (Priority) iCRNs). *For any sequence of applicable void rules A in a given iCRN, there exists a sequence B that is a permutation of A such that all applications of a given rule type occur contiguously.*

Proof. Consider a sequence of applicable void rules A for a given (Priority) iCRN that is not contiguous. We construct a contiguous sequence as described in the Rearrangement Lemma in [12]. For the given sequence A , suppose rule γ occurs at positions i and j such that $i < j - 1$, and there is at least one non- γ rule in between i and j . Construct a sequence A' by shifting the rule γ at position i up to position $j - 1$, and shifting all rules in between down one position.

This new sequence must be applicable as (1) the only rule that moved to a higher index in the sequence is of type γ , and we know that γ is applicable at position $j - 1$ since it is applicable at position j , and (2) because, in a system with just void rules, species are only deleted and not produced. Therefore, all rules that occur between positions i and j must not be inhibited by the reactants of γ (since the reactants of γ are present until position j); therefore, such rules can be applied before all occurrences of γ .

Since this swapping preserves the applicability of the sequence while reducing the number of non-contiguous blocks of one rule type in the sequence, we can repeat this process of swapping rule positions until the sequence is contiguous. ◀

► **Lemma 19.** *The reachability problem for (Priority) iCRNs with only void reactions is in NP.*

Proof. Given a sequence of applicable void rules in a given iCRN, we encode this sequence with a sequence of rule types accompanied by a count on the number of applications of each rule type, which must exist by Lemma 18. Although the original sequence is potentially exponential in length, the result of the contiguous sequence can be computed in polynomial time. Therefore, we utilize a contiguous sequence of applicable rules as a certificate for the reachability problem. ◀

► **Theorem 20.** *The reachability problem in Priority iCRN with only $(2, 0)$ and $(2, 1)$ void rules with maximum priority 1 is NP-complete.*

Proof. Due to space constraints, we provide a proof sketch here, while the complete proof can be found in the full version of the paper on arxiv.

From Lemma 19, we have that the reachability problem for Priority iCRNs with purely void rules is in NP. To show hardness, we reduce from Vertex Cover. Given an instance of VC $\langle G, k \rangle$ where $k > 0$, we construct a Priority iCRN $\mathcal{C}_{PI}$. We create one copy of v for all $v \in V$ and one copy of e for all $e \in E$. We also add two new species to the $\mathcal{C}_{PI}$ system. In the initial configuration, we add $|V| - k$ copies of a species R , used to eliminate $|V| - k$ vertices, and k copies of a species X , used to remove the remaining k vertices. Denote this configuration the starting configuration $\vec{C}_s$. We also order our species set as: $\Lambda = \langle R, v_1, \dots, v_{|V|}, e_1, \dots, e_{|E|}, X \rangle$.

For all $v \in V$, we create the *assignment reaction* $v + R \rightarrow \emptyset$ and the *clean-up reaction* $v + X \xrightarrow{1} \emptyset$. For all edges $e \in E$, for all vertices $v \in V$ where v is incident to e , we create the *covering reaction* $v + e \xrightarrow{1} v$. Each of the clean-up and covering reactions have priority 1.

The target configuration $\vec{C}_t$ is the empty configuration $\vec{0}$. A vertex cover of size k can only exist in G iff all the species copies can be completely deleted. ◀

► **Lemma 21.** *The reachability problem in (Priority) iCRNs with only reactions of size $(2, 2)$ is in PSPACE.*

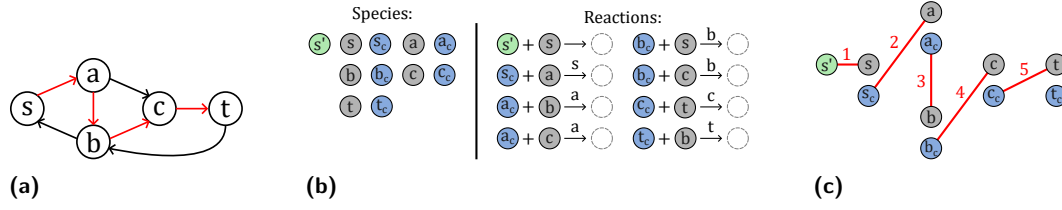
Proof. Given that the system is volume-preserving, the reachability problem is solvable by a non-deterministic Turing machine using only polynomial space, where the tape of the machine stores a configuration (counts of species). Therefore, the problem is in NPSpace, and hence in PSPACE [27]. ◀

► **Observation 22.** *The reachability problem in Priority iCRNs with only reactions of size $(2, 2)$ is PSPACE-complete.*

Proof. The hardness follows from the reachability problem in regular CRNs with only $(2, 2)$ -size rules being PSPACE-complete [11], and membership follows from Lemma 21. ◀

4 Inhibitory CRNs

We now shift our focus to the reachability problem in the iCRN model. Specifically, we prove that reachability of iCRNs with void rules of size $(2, 0)$, $(2, 0) + (2, 1)$, and more generally $(k, k - 1)$ are all NP-complete. Alongside these hardness results, we present FPT algorithms for each of these reachability problems. To conclude, we examine iCRN systems with rules of size $(1, 1)$ and show that the reachability problem is PSPACE-complete.



■ **Figure 2** Transforming an instance of HAMPATH into an instance of iCRN reachability with only $(2,0)$ -size reactions. (a) is a directed graph G with a Hamiltonian path highlighted in red. (b) is the set of species and reactions of $\mathcal{C}_{IC}$ as constructed from G . (c) is the complete reachability reduction, as well as showing how the Hamiltonian path is traversed with $\mathcal{C}_{IC}$. The red line between two species copies represents their deletion by a $(2,0)$ void reaction; the number represents the order the reaction was applied.

4.1 iCRNs with Only Void Rules

For completeness, we include the following observation.

► **Observation 23.** *Given an iCRN $\mathcal{C}_{IC} = ((\Lambda, \Gamma), \mathcal{I})$ with only void rules of size $(1,0)$, the reachability problem is decidable in $\mathcal{O}(|\Lambda||\Gamma|)$ time.*

Proof. Run any applicable rule in the system, reducing the corresponding species to its final count. If the final configuration is reached, then reachability is possible. Otherwise, if no more rules can be applied, reachability is not possible. ◀

► **Theorem 24.** *The reachability problem in iCRNs with only rules of size $(2,0)$ is NP-complete.*

Proof. Due to space constraints, we provide a proof sketch here, while the complete proof can be found in full version of the paper on arxiv.

Lemma 19 shows reachability in void iCRNs is in NP. To show hardness, we reduce from the Hamiltonian path problem. Given a HAMPATH instance $\langle G, s, t \rangle$, we transform it into an instance of reachability with an iCRN $\mathcal{C}_{IC}$. For each vertex $v \in V$, we create the species v and v_c . We also add an extra species s' . For each directed edge $\{u, v\} \in E$, we create the *choosing reaction* $u_c + v \xrightarrow{u} \emptyset$. We also create the *starting reaction* $s' + s \rightarrow \emptyset$. Let the initial configuration $\vec{A}$ be a single copy of each species and the target configuration $\vec{B}$ be a single copy of t_c . A Hamiltonian path from s to t can only exist in G iff the system can delete all copies except for t_c . An example reachability instance is illustrated in Figure 2. ◀

We now explore the reachability problem for a $(2,0)$ void iCRN with a constant number of inhibiting species. We follow the same format as in Priority iCRN section to show that the reachability problem is polynomial time solvable for a constant number of inhibitors. To do so we show that reachability problem in such an iCRN can be reduced to the empty reachability problem in Lemmas 25, 26, and 27. Finally, we reduce the empty reachability problem in the given iCRN to the empty reachability problem in Priority iCRN with $(2,0)$ reactions.

Given a $(2,0)$ void iCRN $\mathcal{C}_{IC} = ((\Lambda, \Gamma), \mathcal{I})$, where the set of species Λ is partitioned into the set of inhibiting species $\Lambda_{\mathcal{I}} = \bigcup_{\gamma \in \Gamma} \mathcal{I}(\gamma)$ and the set of remaining species $\Lambda \setminus \Lambda_{\mathcal{I}}$. We use $\vec{0}$ to denote an empty configuration where all species have a count of zero.

► **Lemma 25.** *Given two configurations $\vec{C}_s$ and $\vec{C}_t$ in a $(2,0)$ void iCRN $\mathcal{C}_{IC}$, $\vec{C}_s \rightsquigarrow \vec{C}_t \iff \vec{C}_s - \vec{C}_t \rightsquigarrow \vec{0}$, if $\forall \lambda_i \in \Lambda_{\mathcal{I}}, \vec{C}_t[\lambda_i] = 0$.*

Proof. Follows from Lemma 8 because the proof does not rely on any ordering of inhibitors. $\blacktriangleleft$

► **Lemma 26.** *Given two configurations $\vec{C}_s$ and $\vec{C}_t$ in a $(2,0)$ void iCRN $\mathcal{C}_{IC}$, $\vec{C}_s \rightsquigarrow \vec{C}_t \iff \vec{C}_s - \vec{C}_t \rightsquigarrow \vec{0}$, if $\exists \lambda_i \in \Lambda_{\mathcal{I}}, \vec{C}_t[\lambda_i] \neq 0$.*

Proof. The proof is similar to that of Lemma 9. Here we remove any reactions inhibited by species $\lambda_i \in \Lambda_{\mathcal{I}}$. Now λ_i is no longer an inhibitor. The remaining set of reactions is Γ' . From Lemma 25, $\vec{C}_s \rightsquigarrow \vec{C}_t \iff \vec{C}_s - \vec{C}_t \rightsquigarrow \vec{0}$ in a iCRN $\mathcal{C}_{IC} = ((\Lambda, \Gamma'), \mathcal{I})$. $\blacktriangleleft$

► **Lemma 27.** *Given two configurations $\vec{C}_s$ and C_t in a $(2,0)$ void iCRN $\mathcal{C}_{IC}$, $\vec{C}_s \rightsquigarrow \vec{C}_t \iff \vec{C}_s - \vec{C}_t \rightsquigarrow \vec{0}$.*

Proof. Follows from Lemmas 25 and 26. $\blacktriangleleft$

► **Lemma 28.** *Given any iCRN system $(\mathcal{C}_{IC}, \vec{C}_s)$ and an ordering on inhibitors, we can construct a Priority iCRN system $(\mathcal{C}_{PI}, \vec{C}_s')$ that contains the same inhibitors in the given ordering. This construction takes $O(|\Lambda| \cdot |\Gamma|)$ where Λ and Γ are the set of species and reactions in the given iCRN.*

Proof (Sketch). We construct an ordered species set Λ' containing all species from the given iCRN. The species are arranged such that all the inhibitors (in order) appear first in the set. The configuration and reactions are also rearranged to match the ordering of species in Λ' . For each reaction, the priority is set to be the maximum index of all inhibitors. Finally, we remove any reactions that are self-inhibiting, i.e., the priority of the reaction is greater or equal to the index of at least one of the reactants. The complete proof can be found in full version of the paper on arxiv. $\blacktriangleleft$

► **Lemma 29.** *Given an iCRN $\mathcal{C}_{IC} = ((\Lambda, \Gamma), \mathcal{I})$ with only void rules of size $(2,0)$, the empty configuration reachability problem is decidable in $O(c! \cdot f(\Lambda, \Gamma))$ time where c is the number of inhibitors and $f(\Lambda, \Gamma) = |\Lambda|^2 \log(|\Lambda|)(|\Gamma| + |\Lambda| \log(|\Lambda|))$.*

Proof. Given an iCRN $\mathcal{C}_{IC} = ((\Lambda, \Gamma), \mathcal{I})$ with $(2,0)$ void rules and some start configuration $\vec{C}_s$. Given that the iCRN has at most c inhibitors, there exist at most $c!$ different orderings in which the inhibitors are removed.

To solve the empty configuration reachability problem in iCRN, we reduce it to $c!$ instances of the empty configuration reachability problem in Priority iCRN. From Lemma 12 we know that this problem is solvable in $O(f(\Lambda, \Gamma))$. For each ordering $\langle i_1, i_2, \dots, i_c \rangle$ where i_j is some species that acts as an inhibitor, we construct a Priority iCRN as described in Lemma 28. We now show that the empty configuration is reachable in the given iCRN iff the empty configuration is reachable in at least one of the $c!$ Priority iCRNs.

▷ **Claim 1.** If the empty configuration is reachable in iCRN, then it is also reachable in at least one of the constructed Priority iCRNs.

This is true simply by construction of the Priority iCRNs. If the iCRN reaches the empty configuration, then all species, including inhibitor species, go to zero. Therefore, there must exist an ordering in which these inhibitors are deleted. Since we consider every possible ordering, at least one of the $c!$ Priority iCRNs orderings will reach the empty configuration.

▷ **Claim 2.** If the empty configuration is reachable in at least one of the constructed Priority iCRNs, then the iCRN reaches the empty configuration.

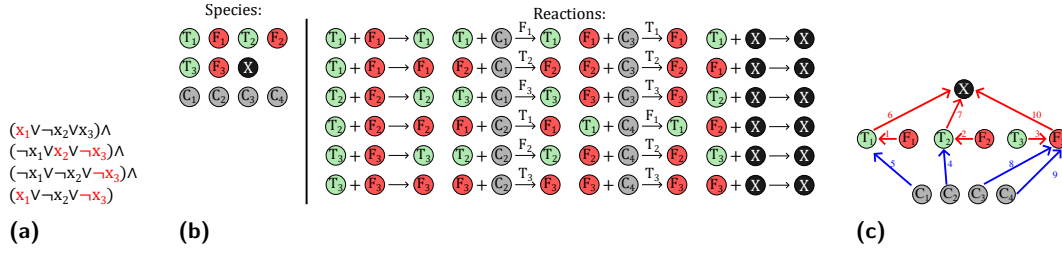


Figure 3 Transforming an instance of 3SAT into an instance of iCRN reachability with only $(2, 1)$ -size reactions. (a) is a 3-CNF formula Φ . A satisfying assignment of Φ is represented by the red letters. (b) is the set of species and reactions of $\mathcal{C}_{IC}$ as constructed from Φ . (c) is the complete reachability reduction, as well as showing how the formula is satisfied with $\mathcal{C}_{IC}$. The red line between two species copies represents their deletion by a $(2, 1)$ void reaction.

Let the ordering corresponding to the Priority iCRN for which the empty configuration is reachable be $\langle i_1, \dots, i_c \rangle$. Modify the reaction set Γ of the given iCRN by removing all reactions of the form $i_j + _ \rightarrow \phi$ or $_ + i_j \rightarrow \phi$ if they are inhibited by species $i_{k>j}$ since i_j is deleted before i_k . For the remaining reactions, apply the reactions in the same order in which their corresponding reactions in $\mathcal{C}_{PI}$ are applied. If none of the $c!$ Priority iCRNs reach the empty configuration, then the given iCRN will also not reach the empty configuration. $\blacktriangleleft$

Theorem 30. *Given an iCRN $\mathcal{C}_{IC} = ((\Lambda, \Gamma), \mathcal{I})$ with only void rules of size $(2, 0)$, the reachability problem is decidable in $O(c! \cdot f(\Lambda, \Gamma))$ time where c is the number of inhibitors and $f(\Lambda, \Gamma) = |\Lambda|^2 \log(|\Lambda|)(|\Gamma| + |\Lambda| \log(|\Lambda|))$.*

Proof. Similar to the case of $(2, 0)$ Priority iCRN, proof follows from Lemmas 27 and 29. $\blacktriangleleft$

Theorem 31. *The reachability problem in $(2, 1)$ void iCRN is NP-complete.*

Proof. Due to space constraints, we provide a proof sketch here, while the complete proof can be found in full version of the paper on arxiv.

From Lemma 19, we have that the reachability problem in void iCRNs is in NP. To show hardness, we reduce from the 3-satisfiability problem. Given an instance of 3SAT $\langle \Phi \rangle$, we create a new iCRN $\mathcal{C}_{IC}$. For each variable $x_i \in X_n$, we create the species T_i and F_i . Additionally, for each clause $c_j \in C_m$, we create the species C_j . Finally, we create the species X . For each variable x_i , we also create the *assignment reactions* $T_i + F_i \rightarrow T_i$ and $T_i + F_i \rightarrow F_i$. For each clause c_j and a variable of c_j x_k , we create the *clause reaction* $C_i + T_k \xrightarrow{F_k} T_k$ if assigning true to x_k satisfies c_j , or $C_i + F_k \xrightarrow{T_k} F_k$ if the false assignment satisfies instead. Finally, for each variable, we create the *clean-up reaction* $T_i + X \rightarrow X$ and $F_i + X \rightarrow X$. The initial configuration $\vec{C}_s$ is a single copy of each species and the target configuration $\vec{C}_t$ is a single copy of X . Φ can only be satisfied iff the system can delete all copies except for X . An example reachability instance is illustrated in Figure 3. $\blacktriangleleft$

Theorem 32. *The reachability problem in $(k, k - 1)$ void iCRN is NP-complete.*

Proof. Follows from Theorem 31. Given an iCRN $\mathcal{C}_{IC}$, we modify it by adding a species d and modifying each reaction to include $k - 2$ copies of d for both reactants and products (i.e., $T_x + F_x + d \rightarrow T_x + d$ for $k = 3$). We include $k - 2$ copies of d for $\vec{C}_s$ and $\vec{C}_t$. Since d acts as a catalyst species for all reactions, its count will never change, and it does not affect the behavior of $\mathcal{C}_{IC}$. Thus, the forward and reverse directions from Lemma 31 follow here. $\blacktriangleleft$

► **Theorem 33.** *Given an iCRN $\mathcal{C}_{\mathcal{I}} = ((\Lambda, \Gamma), \mathcal{I})$ with only void rules of size $(k, k-1)$, the reachability problem is decidable in $\mathcal{O}(c! \cdot |\Lambda|^2 |\Gamma|)$ time, where c is the number of inhibitors.*

Proof. This follows by using the algorithm for Priority iCRN void systems with size $(k, k-1)$ rules from Theorem 17 as a subroutine. Denote this algorithm g . Consider an arbitrary ordering of the c inhibitors $\lambda_1, \dots, \lambda_c$. Run algorithm g with this ordering. If a solution is found, then reachability is possible. Otherwise, consider another ordering until all are considered. The algorithm then runs in $\mathcal{O}(c! \cdot |\Lambda|^2 |\Gamma|)$ time. ◀

We show that reachability is NP-complete with $(2, 0)$ and $(2, 1)$ reactions, even if only one species is used to inhibit reactions.

► **Theorem 34.** *The reachability problem in $(2, 0)$ and $(2, 1)$ void iCRNs is NP-complete, even with a single inhibitor species.*

Proof. From Lemma 19, we have that the reachability for iCRNs with purely void rules is in NP. We now show hardness with a similar reduction as in Theorem 20. We construct our iCRN $\mathcal{C}_{\mathcal{I}C}$ in a similar manner, where our species set is now unordered. Here, the species set $\Lambda = \{R, v_1, \dots, v_{|V|}, e_1, \dots, e_{|E|}, X\}$. Our reaction set contains *assignment reactions* $v + R \rightarrow \phi$, *clean-up reactions* $v + X \xrightarrow{R} \phi$, and *covering reactions* $v + e \xrightarrow{R} v$. The initial and target configurations remain the same. ◀

4.2 iCRNs with General rules

► **Theorem 35.** *The reachability problem in iCRNs with $(1, 1)$ rules is PSPACE-complete.*

Proof. PSPACE membership follows from Lemma 21. We now show hardness. Given a Turing machine $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_s, q_a, q_r)$ with bounded tape length n , we create an iCRN $\mathcal{C}_{\mathcal{I}C} = (\Lambda, \Gamma_C)$. The set Λ consists of the helper species D , H , and E , the tape symbols $\{x_i, 0_i, 1_i\}$ for each tape cell $i \in [1, n]$, and the following sets of species: $\mathcal{Q} = \{Q_y | q_y \in Q\}$, $\mathcal{Q}^0 = \{Q_y 0_i\}_{q_y \in Q, i \in [1, n]}$ and $\mathcal{Q}^1 = \{Q_y 1_i\}_{q_y \in Q, i \in [1, n]}$. Table 2 shows the reactions created.

The initial configuration I of the system consists of one instance of 0_i or 1_i for each tape cell i , according to the initial tape of $\mathcal{M}$. Given the start state q_s , we also add a single copy of Q_s . If the tape head starts at cell i , we include one instance of each species x_j for all $j \neq i$, i.e., $\{x_j\}_{j \neq i}$. Finally, we include one copy each of the helper species D and H . The target configuration T is volume I copies of E .

Forward Direction. Assume we are given a Turing Machine that will reach a halt state by applying a series of transitions from δ , each according to the tape value at head and the state. The corresponding iCRN $\mathcal{C}_{\mathcal{I}C}$ will likewise reach target configuration T by applying a series of reactions representing those transitions, according to the current configuration C . When a single D is present in C , C maps to the Turing Machine. Q_y indicates $\mathcal{M}$ is in state $q_y \in Q$. The Tape head is at tape cell j where x_j is absent. The data for tape cell i is saved in the presence of a copy of 1_i , or one of 0_i . All reactions shown in Table 2 except 1.3, 7.1 and 7.2 handle normal transitions, updating the given configuration to map to the result of the next transition. In this order, the reactions: (1) read the tape, (2) determine the next state, (3) mark the tape head ready to move, (4) change the state, (5) write and move the tape head, and (6) indicate ready for next transition. If the current state is a halt state, reaction 1.3 is applied, then (7) is repeatedly applied until we reach a terminal configuration. The final terminal configuration will only be T if the TM's tape and state input deterministically transitions to q_a according to δ .

■ **Table 2** Reaction set for the $(1, 1)$ iCRN construction.

No.	Notation	Reaction	Intuition
1.1	$\mathcal{Q}_H : Q_a, Q_r$	$D \xrightarrow{\mathcal{Q}_H, \overline{\mathcal{Q}_y}, x_i, 0_i} Q_y 1_i$	Read the tape can combine with state, unless state is halt.
1.2	$\overline{\mathcal{Q}_y} : \mathcal{Q} \setminus Q_y$	$D \xrightarrow{\mathcal{Q}_H, \overline{\mathcal{Q}_y}, x_i, 1_i} Q_y 0_i$	
1.3	$\overline{\mathcal{Q}_H} : \mathcal{Q} \setminus \mathcal{Q}_H$	$D \xrightarrow{\overline{\mathcal{Q}_H}} E$	
2	$\overline{\mathcal{Q}^V}_{\{y,i\}} : \{\mathcal{Q}^0 \cup \mathcal{Q}^1\} \setminus Q_y V_i$ $z : \delta_{state}(y, V)$	$Q_y \xrightarrow{D, E, \overline{\mathcal{Q}^V}_{\{y,i\}}} Q_{zi}$	Prepare to change into the appropriate z according to δ .
3.1		$1_i \xrightarrow{D, E, \mathcal{Q}, x_i} x_i$	Mark read as done by producing x_i .
3.2		$0_i \xrightarrow{D, E, \mathcal{Q}, x_i} x_i$	
4		$Q_{zi} \xrightarrow{1_i, 0_i} Q_z$	Finish changing state after producing x_i .
5	$b : \delta_{dir}(y, V)$ $c : \delta_{write}(y, V)$	$x_{i+b} \xrightarrow{E, 1_i, 0_i, \overline{\mathcal{Q}^V}_{\{y,i\}}, Q_{zi}} c_i$	Move+write based on state and value read V .
6		$Q_y V_i \xrightarrow{x_{i+b}} D$	Once moved, replace D .
7.1	$\mathcal{Q}' : \{\mathcal{Q}^1 \cup \mathcal{Q}^0\}$	$H \xrightarrow{D, \mathcal{Q}'} E$	If $\mathcal{M}$ halts, delete H , and go to term config.
7.2	$S : \Lambda \setminus \{E, Q_r\}$	$S \xrightarrow{H} E$	

Reverse Direction. To reach the target configuration T of iCRN $\mathcal{C}_{IC}$ which was designed to simulate bounded length N TM A , reaction 7.2 must run $2N$ times. This can only happen if Q_a was present, preventing reactions 1.1 and 1.2, and not being Q_r , which would also prevent the reactions, but could not be removed by 7.2. For Q_a to be present, it either was in I , or was produced by a type 4 reaction. To have Q_a produced by a type 4 reaction, the most recent type 2 reaction must have produced Q_{ai} . For any reaction of type 2 to be applicable, D cannot be present, so before this reaction could occur, a reaction of type 1.1 or 1.2 must have been run. It produced a copy of $Q_y V_i$ where Q_y represents the state the system was in before Q_a , and V is the symbol at our tape head. $Q_y V_i$ prevents all type 2 reactions except $\delta_{state}(y, V)$, so we could only have reached Q_a if $\delta_{state}(y, V) = q_a$. For our configuration to have included a 1_i or 0_i and Q_y which would lead to Q_a , they either must have been in I , or were produced by repetitions of the same process. This process also includes one reaction of type 3 and one of type 5, which together update the tape. Thus we could only have reached the target configuration if the initial tape values and State deterministically Halt in an accept state. ◀

► **Theorem 36.** *The reachability problem in iCRN with $(1, 1)$ rules and at most 1 inhibitor per rule is NP-Hard.*

Proof. We show this by a reduction from 3SAT. Consider a 3SAT instance of n variables $X_1, \dots, X_n$ and m clauses $C_1, \dots, C_m$, and assume no clause contains both x_i and $\bar{x}_i$. From this instance, we create the following $(1, 1)$ iCRN system and start and end configurations for the Reachability problem.

For each variable $x_i \in X_n$, we create the species y_i, x_i, T_i , and F_i . For each clause $c_j \in C_m$, we create the species $c_j^1, c_j^2, c_j^3, d_j^1, d_j^2, d_j^3$, and S_j . Additionally, for each variable, we create the reactions $y_i \rightarrow T_i, y_i \rightarrow F_i, x_i \xrightarrow{x_{i-1}} T_i$ (no inhibitor for $i = 1$), and $x_i \xrightarrow{x_{i-1}} F_i$ (no inhibitor for $i = 1$). For each clause, we create the reactions $c_j^k \xrightarrow{x_n} d_j^k$, either $d_j^k \xrightarrow{T_i} S_j$

if $\bar{x}_i$ satisfies the k^{th} literal of clause C_j or $d_j^k \xrightarrow{F_i} S_j$ if x_i satisfies the k^{th} literal of clause C_j , and $d_j^k \rightarrow W$. Finally, let the initial configuration be one copy for each of y_i , x_i , and c_j^1 , c_j^2 , c_j^3 , and the target configuration be one copy for each of T_i , F_i , S_j , and $2m$ copies of W .

Forward Direction. Suppose the 3SAT formula is satisfiable. We can then reach the destination configuration by first transitioning each x_i species into the appropriate true/false species T_i/F_i that matches the satisfying assignment for the 3SAT formula. Next, convert exactly one of the three c_j species from each clause into S_j (based on which literal was satisfied by the satisfying variable assignment), and the remaining two into copies of species W . Finally, convert each y_i into the opposite truth value taken on by x_i (to ensure there is exactly one T_i and F_i for each i).

Reverse Direction. Suppose we can reach the destination configuration, which specifically requires creating each species S_j . Each S_j can only be created by the absence of some F_i where the clause contains literal X_i , or the absence of a T_i where the clause contains $\bar{X}_i$. Since the c_j to d_j transformation must precede the creation of S_j , we know that each x_i must be converted to a T_i or F_i before the S_j 's can be created. Therefore, the only possible way to create each S_j is for each x_i to select a truth assignment that corresponds to a satisfying assignment of the given 3SAT formula. ◀

► **Observation 37.** *The reachability problem for iCRNs with rule size $(2, 2)$ is PSPACE-complete.*

Proof. The proof is analogous to the proof of Observation 22. ◀

5 Conclusion

This paper provides a thorough treatment of reachability in restricted CRNs with inhibition. Our results (Table 1) show a clear boundary between tractable / intractable cases under prioritized and unprioritized inhibition. For Priority iCRNs, reachability remains solvable in polynomial time for most unimolecular, bimolecular, and mostly catalytic void rules, but becomes NP-complete with combined $(2, 0)$ and $(2, 1)$ void rules. For general iCRNs, nearly all cases become NP-complete (except $(1, 0)$), and we give FPT algorithms for the $(2, 0)$, $(2, 1)$, and $(k, k - 1)$ cases. For volume-preserving reactions, $(1, 1)$ reachability – polynomial-time solvable for basic CRNs – drastically increases to PSPACE-complete for general iCRNs.

While many of our complexity classifications are tight, several gaps remain in Table 1. For inhibitory systems with $(2, 0)$ and $(2, 1)$ void rules, we establish NP-completeness with even one inhibitor, but it remains open whether alternative parameters (e.g., the number of inhibited rules) yield FPT algorithms or whether the problem is instead W[1]-hard. It may be possible to improve our existing FPT algorithms with some pruning method that avoids exhaustively trying all inhibitor permutations. Another key question is the complexity of $(1, 1)$ Priority iCRNs: the problem is in P for basic CRNs and PSPACE-complete for general inhibition, but it is unclear where prioritized inhibition falls. Lastly, we consider *CRNs with States*, analogous to VASS, where reactions are conditioned on a global state; this model can be viewed as a form of inhibition and motivates studying its reachability complexity. In the full version of this paper, we formalize CRNs with States and provide preliminary results that show reachability is NP-hard for $(1, 0)$ rules and PSPACE-complete for $(1, 1)$ rules.

References

- 1 Robert M. Alaniz, Bin Fu, Timothy Gomez, Elise Grizzell, Andrew Rodriguez, Marco Rodriguez, Robert Schweller, and Tim Wylie. Reachability in restricted chemical reaction networks. *Theoretical Computer Science*, 1056:115514, 2025. doi:10.1016/j.tcs.2025.115514.
- 2 Rachel Anderson, Alberto Avila, Bin Fu, Timothy Gomez, Elise Grizzell, Aiden Massie, Gourab Mukhopadhyay, Adrian Salinas, Robert Schweller, Evan Tomai, and Tim Wylie. Computing threshold circuits with void reactions in step chemical reaction networks. In *10th conference on Machines, Computations and Universality (MCU 2024)*, 2024.
- 3 Richard P. Anstee. A polynomial algorithm for b-matchings: An alternative approach. *Information Processing Letters*, 24(3):153–157, 1987. doi:10.1016/0020-0190(87)90178-5.
- 4 Rutherford Aris. Prolegomena to the rational analysis of systems of chemical reactions. *Archive for Rational Mechanics and Analysis*, 19(2):81–99, January 1965. doi:10.1007/BF00282276.
- 5 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, January 1968. doi:10.1007/BF00251438.
- 6 Kim Calabrese and David Doty. Rate-independent continuous inhibitory chemical reaction networks are turing-universal. In *International Conference on Unconventional Computation and Natural Computation*, pages 104–118. Springer, 2024. doi:10.1007/978-3-031-63742-1_8.
- 7 Luca Cardelli, Marta Kwiatkowska, and Max Whitby. Chemical reaction network designs for asynchronous logic circuits. *Natural computing*, 17:109–130, 2018. doi:10.1007/S11047-017-9665-7.
- 8 Ho-Lin Chen, David Doty, and David Soloveichik. Deterministic function computation with chemical reaction networks. *Natural computing*, 13:517–534, 2014. doi:10.1007/S11047-013-9393-6.
- 9 Matthew Cook, David Soloveichik, Erik Winfree, and Jehoshua Bruck. Programmability of chemical reaction networks. In *Algorithmic bioprocesses*, pages 543–584. Springer, 2009. doi:10.1007/978-3-540-88869-7_27.
- 10 Wojciech Czerwiński and Łukasz Orlikowski. Reachability in vector addition systems is ackermann-complete. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1229–1240. IEEE, 2022.
- 11 Javier Esparza, Mikhail Raskin, and Chana Weil-Kennedy. Parameterized analysis of immediate observation petri nets. In Susanna Donatelli and Stefan Haar, editors, *Application and Theory of Petri Nets and Concurrency*, pages 365–385, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-21571-2_20.
- 12 Bin Fu, Timothy Gomez, Ryan Knobel, Austin Luchsinger, Aiden Massie, Marco Rodriguez, Adrian Salinas, Robert Schweller, and Tim Wylie. Reachability in deletion-only chemical reaction networks. In *Proc. of the International Conference on DNA Computing and Molecular Programming*, DNA, 2025.
- 13 Roland Guttenberg. Flattability of priority vector addition systems. *arXiv preprint arXiv:2402.09185*, 2024. doi:10.48550/arXiv.2402.09185.
- 14 Michel Henri Théodore Hack. *Decidability questions for Petri Nets*. PhD thesis, Massachusetts Institute of Technology, 1976.
- 15 Michel Henri Théodore Hack. Petri net language. Technical report, Massachusetts Institute of Technology, 1976.
- 16 Allen Hjelmfelt, Edward D Weinberger, and John Ross. Chemical implementation of neural networks and turing machines. *Proc. of the National Academy of Sciences*, 88(24):10983–10987, 1991.
- 17 Hua Jiang, Marc D Riedel, and Keshab K Parhi. Digital logic with molecular reactions. In *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 721–727. IEEE, 2013. doi:10.1109/ICCAD.2013.6691194.

- 18 Richard M Karp and Raymond E Miller. Parallel program schemata: A mathematical model for parallel computation. In *8th Annual Symposium on Switching and Automata Theory (SWAT 1967)*, pages 55–61. IEEE, 1967. doi:10.1109/FOCS.1967.27.
- 19 Richard M Karp and Raymond E Miller. Parallel program schemata. *Journal of Computer and system Sciences*, 3(2):147–195, 1969. doi:10.1016/S0022-0000(69)80011-5.
- 20 Jérôme Leroux. The reachability problem for Petri nets is not primitive recursive. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1241–1252. IEEE, 2022.
- 21 Richard Lipton. The reachability problem requires exponential space. *Research Report 62. Department of Computer Science, Yale University*, 1976.
- 22 Ernst W Mayr. An algorithm for the general petri net reachability problem. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, pages 238–246, 1981. doi:10.1145/800076.802477.
- 23 Rajiv Teja Nagipogu and John H. Reif. Neural crns: A natural implementation of learning in chemical reaction networks. *ACS Synthetic Biology*, 14(10):3899–3912, 2025. doi:10.1021/acssynbio.5c00099.
- 24 James Lyle Peterson. Petri net theory and the modeling of systems, 1981.
- 25 Carl Adam Petri. *Communication with automata*. PhD thesis, Technische Universität Darmstadt, 1966.
- 26 Klaus Reinhardt. Reachability in petri nets with inhibitor arcs. *Electronic Notes in Theoretical Computer Science*, 223:239–264, 2008. Proceedings of the Second Workshop on Reachability Problems in Computational Models (RP 2008). doi:10.1016/j.entcs.2008.12.042.
- 27 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970. doi:10.1016/S0022-0000(70)80006-X.
- 28 David Soloveichik, Georg Seelig, and Erik Winfree. Dna as a universal substrate for chemical kinetics. *Proc. of the National Academy of Sciences*, 107(12):5393–5398, 2010.
- 29 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.