

Certifying complete families of irreducible autocatalytic sets is co-W[P]-complete

Anonymous manuscript

September 13, 2026

Abstract

An irreducible reflexively autocatalytic and food-generated set (irrRAF) is an inclusion-minimal self-sustaining reaction subsystem of a catalytic reaction system. Steel, Hordijk and Smith (*J. Theor. Biol.*, 2013) gave a criterion for deciding whether a supplied list of k irrRAFs is complete, that is, whether every irrRAF of the system appears on the list. Their procedure runs in polynomial time for each fixed k , with an exponent that grows with k , and they asked whether the problem is fixed-parameter tractable in k . We show that it is not, unless $\text{FPT} = \text{W[P]}$: exact completeness certification is co-W[P]-complete when parameterized by the number of listed irrRAFs, and coNP-complete without parameterization. Hardness already holds for systems with a single food molecule whose supplied family consists of k pairwise reaction-disjoint simple catalytic cycles. The reduction starts from MINIMUM AXIOM SET. Each listed cycle is a guard whose unique omitted reaction selects one axiom; decoder reactions read the omission, a literal implication network derives the remaining statements, and a single closing reaction supplies the common catalyst that makes everything else self-sustaining. Every unlisted irrRAF is shown to determine a generating axiom set, including when the implication system is cyclic. Membership follows from the deletion criterion together with the bounded-nondeterminism characterization of W[P]. A second, Lean-verified reduction from k -CLIQUE shows that no algorithm runs in time $f(k) N^{o(k)}$ unless the Exponential Time Hypothesis fails, so the exponent of the known algorithm cannot be made sublinear in k ; the problem is fixed-parameter tractable in k plus the size of the largest listed irrRAF. Consequences for the problem of finding one more irrRAF and for bounded-delay enumeration are derived. The finite reductions, the validity of the supplied families, the exact parameter, the one-deletion irreducibility test and the deletion criterion are verified in the Lean 4 proof assistant with warnings promoted to errors; the complexity-class transfers are conventional proofs given in full.

Keywords. autocatalytic sets; RAF theory; irreducible RAFs; parameterized complexity; W[P]-completeness; Minimum Axiom Set; Exponential Time Hypothesis; formal verification.

MSC 2020. 68Q17, 68Q27, 92E20, 68V20.

1 Introduction

Reflexively autocatalytic and food-generated sets (RAFs) are a combinatorial model of collectively self-sustaining reaction networks. The notion goes back to Kauffman’s autocatalytic sets [18], was given its present form by Hordijk and Steel [14], and has developed into a structural theory with polynomial-time algorithms and applications to the origin of metabolism [16, 13, 15, 20, 21, 23]. A catalytic reaction system consists of molecule types, reactions, a food set of freely available molecules, and a catalysis relation. A nonempty set of reactions is a RAF if all of its reactants can be built up from the food set by reactions of the set and each of its reactions is catalysed by a food molecule or by a product of the set.

Because unions of RAFs are RAFs, a system that contains a RAF contains a unique largest one, the maxRAF, and it can be computed in polynomial time [14]. The minimal self-sustaining subsystems point the other way. An *irreducible* RAF (irrRAF) is a RAF containing no proper RAF. Every RAF contains an irrRAF, and a single irrRAF can be extracted in polynomial time by repeated deletion [19, 13]. Random deletion orders produce different irrRAFs, and a system can contain exponentially many of them. This raises a stopping question that every discovery procedure eventually faces:

Having found several irreducible autocatalytic sets, how hard is it to establish that none remain undiscovered?

Steel, Hordijk and Smith [19, Section 6.1] answered the question algorithmically. Their Theorem 2 states that a list $I_1, \dots, I_k$ of irrRAFs is complete if and only if for every choice of one reaction $r_i \in I_i$ the system obtained by deleting $r_1, \dots, r_k$ contains no RAF. Running the RAF algorithm once for each of the at most $\prod_i |I_i|$ choices decides completeness in polynomial time for every fixed k . The degree of that polynomial grows with k , and the authors asked whether completeness testing is fixed-parameter tractable in the number k of listed irrRAFs. That is the question answered here.

Results

We treat completeness certification as a total decision problem on explicitly encoded inputs. An instance is a finite catalytic reaction system $\mathcal{Q}$ together with a finite family $\mathcal{L}$ of reaction subsets; the parameter is $k = |\mathcal{L}|$. The instance is a yes-instance of EXACT if every member of $\mathcal{L}$ is an irrRAF of $\mathcal{Q}$ and every irrRAF of $\mathcal{Q}$ belongs to $\mathcal{L}$. The complementary question on valid families, EXTRA, asks whether some irrRAF is missing from the list.

Theorem 1.1 (Main theorem, informal). *EXACT is coNP-complete and, parameterized by $k = |\mathcal{L}|$, co-W[P]-complete. Equivalently, EXTRA on valid families is NP-complete and W[P]-complete. Hardness holds on instances with one food molecule in which the supplied family consists of exactly k pairwise reaction-disjoint irrRAFs, each a simple catalytic cycle whose reactions consume only food.*

Hardness is an unconditional classification. Its algorithmic reading is conditional in the usual way: a certification algorithm with running time $f(k) N^c$, with c independent of k and N the encoded input length, would imply $\text{FPT} = \text{W[P]}$, and a polynomial-time algorithm would imply $\text{P} = \text{NP}$. Since $\text{W[1]} \subseteq \text{W[P]}$, the more familiar assumption $\text{FPT} \neq \text{W[1]}$ suffices for the first statement. Two further results sharpen the picture.

- *Exponent lower bound.* A separate, Lean-verified reduction from k -CLIQUE preserves the parameter exactly and has polynomial output size with an exponent independent of k . By the lower bound of Chen, Huang, Kanj and Xia [4], EXACT has no algorithm with running time $f(k) N^{o(k)}$ unless the Exponential Time Hypothesis [17] fails (Corollary 7.3). The exponent of the deletion procedure of [19] is therefore linear in k for a reason, not for lack of a better idea.
- *Positive boundary.* The same deletion procedure runs in time $\ell^k N^{O(1)}$ where ℓ is the size of the largest listed irrRAF, so EXACT is fixed-parameter tractable in the combined parameter $k + \ell$ (Proposition 7.5). The hard instances have ℓ unbounded, and no parameterization by the overlap structure of the family alone can help: the hard families are pairwise disjoint.

From the classification we also obtain that any procedure which, given a valid list, either returns a further irrRAF or correctly reports that none exists is NP-hard and not fixed-parameter

tractable in k unless $\text{FPT} = \text{W[P]}$, and that a duplicate-free enumerator of all irrRAFs with polynomially bounded delay would imply $\text{P} = \text{NP}$ (Section 7). The last statement concerns delay, not output-polynomial total time; the question of output-polynomial enumeration, raised in [19], remains open.

Relation to earlier hardness results

Two hardness results about minimal autocatalytic networks precede this one and must be distinguished from it. Steel, Hordijk and Smith proved that finding a RAF of *minimum size* is NP-hard, and asked whether the minimum can be approximated within a constant factor [19]; a companion paper answers this negatively unless $\text{P} = \text{NP}$ by an approximation-preserving reduction from SET COVER [2]. That construction also shows that the irrRAFs of a system can be exactly the minimal transversals of a hypergraph, so listing all irrRAFs is at least as hard as hypergraph dualization [8, 10]. Neither result bears on the present problem. Difficulty of optimizing the size of one RAF says nothing about the difficulty of checking a supplied list of inclusion-minimal RAFs, and the hardness of *producing* a possibly exponential output is different from the hardness of *deciding* whether a given polynomial-size list is exhaustive. The completeness question is a decision problem whose input includes the list, and its natural parameter is the length of that list. This is the setting of the 2013 fixed-parameter question, and it is the setting of Theorem 1.1.

Hordijk has recently described practical algorithms for enumerating and sampling irreducible and closed autocatalytic sets, including a completeness test built on the deletion criterion [12]. Those algorithms are exact and their running times are bounded in terms of the deletion criterion; they do not claim, and by Theorem 1.1 cannot achieve unless $\text{FPT} = \text{W[P]}$, a running time of the form $f(k) N^{O(1)}$.

Proof idea

The upper bound is the deletion criterion read as a witness: an unlisted irrRAF omits at least one reaction from each listed set, so the k omitted reactions, with $O(k \log N)$ bits, together with a polynomial-time maxRAF computation certify incompleteness. This is exactly the bounded-nondeterminism form of W[P] [3, 9].

The lower bound is a reduction from MINIMUM AXIOM SET [1], the problem of deciding whether a set of k statements suffices to derive a whole universe of statements under given implications. The constructed system has k slots. Each slot carries one listed irrRAF, a cycle of *selector* reactions $s_{i,u}$, one per statement u , each consuming only food and catalysed by the product of its cyclic successor. A selector cycle is self-sustaining and irreducible on its own, and any RAF that avoids the closing reaction is forced, catalyst by catalyst, to be a whole cycle. An unlisted irrRAF must therefore contain the closing reaction, and it must omit at least one selector from every slot. *Decoder* reactions $d_{i,u}$ require the products of all selectors of slot i other than $s_{i,u}$; an unlisted irrRAF can carry a decoder only in the slot’s unique omitted position, and that position is the chosen axiom. Implication reactions copy the source rules literally on statement molecules, and the closing reaction t consumes every statement molecule and every slot marker and produces the common catalyst z of all non-selector reactions. A generating axiom set yields a RAF containing t whose minimal sub-RAFs are unlisted; conversely, an induction on the stages of the food closure shows that every statement molecule reached by an unlisted irrRAF is derivable from the omitted axioms. The induction follows the first entry of each molecule into the closure, so cyclic implications, which are permitted by the source problem, cannot manufacture ungrounded derivations.

Formal verification

The finite reduction, the validity and exact cardinality of the supplied family, the deletion criterion, the one-deletion irreducibility test and the k -CLIQUE reduction used for the ETH consequence have been formalized in Lean 4 [5] on top of Mathlib [22] and compiled with warnings promoted to errors; Section 8 gives the concordance. The formalization treats the reduction as a finite combinatorial equivalence between the source predicate and the certification predicate; complexity classes, the external completeness theorem for MINIMUM AXIOM SET, and the construction-time analysis are conventional proofs given in full in this paper. Before formalization the construction was tested exhaustively on all 256 implication systems over two statements for $k \leq 2$; those computations are reported in Section 8 and play no role in the proofs.

Organization

Section 2 fixes the RAF semantics, states the certification problems precisely, and recalls the parameterized-complexity notions used. Section 3 proves the deletion criterion and the membership results. Section 4 presents the reduction from MINIMUM AXIOM SET with a worked example, and Section 5 proves its correctness. Section 6 assembles the size analysis and the main theorem. Section 7 derives the consequences and the ETH lower bound. Section 8 describes the formal verification, and Section 9 discusses scope and open questions.

2 Preliminaries and problem statement

2.1 Catalytic reaction systems and RAFs

A *catalytic reaction system* (CRS) is a tuple $\mathcal{Q} = (X, R, F, \text{in}, \text{out}, C)$ where X and R are finite sets of molecule types and reaction identifiers, $F \subseteq X$ is the *food set*, $\text{in}(r) \subseteq X$ and $\text{out}(r) \subseteq X$ are the reactants and products of reaction r , and $C \subseteq X \times R$ is the *catalysis relation*: $(x, r) \in C$ means that molecule x catalyses reaction r . Distinct reaction identifiers may have equal reactant and product sets. All sets and incidence relations are represented explicitly in the input.

For $S \subseteq R$ the *staged food closure* is defined by $K_0(S) = F$ and

$$K_{t+1}(S) = K_t(S) \cup \bigcup \{\text{out}(r) : r \in S, \text{in}(r) \subseteq K_t(S)\},$$

and $\text{cl}(S) = \bigcup_t K_t(S)$. The stages are increasing and at most $|X|$ of them are strict, so $\text{cl}(S) = K_{|X|}(S)$ is computable in polynomial time. Closure is monotone in S : if $U \subseteq S$ then $K_t(U) \subseteq K_t(S)$ for all t .

Definition 2.1 (RAF, irrRAF). A nonempty set $S \subseteq R$ is a *RAF* of $\mathcal{Q}$ if every $r \in S$ satisfies $\text{in}(r) \subseteq \text{cl}(S)$ (*food-generated*) and there is an $x \in \text{cl}(S)$ with $(x, r) \in C$ (*reflexively autocatalytic*). A RAF is *irreducible* (an *irrRAF*) if no proper nonempty subset of it is a RAF.

Remark 2.2 (Semantic convention). Catalysts are checked in the final closure $\text{cl}(S)$; they are not prerequisites for a reaction to contribute its products to the closure. A catalyst may therefore be produced at a later stage than the reactions it catalyses, and a reaction may be catalysed by its own product. This is the standard RAF condition of [14, 19], sometimes called “non-temporal” catalysis; it is the convention under which the maxRAF algorithm and the 2013 deletion criterion are stated, and it is the convention formalized in Section 8.

The union of two RAFs is a RAF, so every $S \subseteq R$ that contains a RAF contains a unique maximal one. It is computed by the RAF algorithm of Hordijk and Steel [14], which we recall in the form used below.

Lemma 2.3 (maxRAF oracle). *For $S \subseteq R$ let $T_0 = S$ and let T_{j+1} be the set of $r \in T_j$ with $\text{in}(r) \subseteq \text{cl}(T_j)$ and $(x, r) \in C$ for some $x \in \text{cl}(T_j)$. The sequence stabilizes after at most $|S|$ rounds at a set $\text{maxRAF}(S)$, computable in polynomial time, which contains every RAF $U \subseteq S$ and is itself a RAF whenever it is nonempty. In particular S contains a RAF if and only if $\text{maxRAF}(S) \neq \emptyset$.*

Proof. Each non-final round removes at least one reaction, so there are at most $|S|$ rounds, each polynomial. If $U \subseteq T_j$ is a RAF then $\text{cl}(U) \subseteq \text{cl}(T_j)$ by monotonicity, so every $r \in U$ has its reactants and a catalyst in $\text{cl}(T_j)$ and survives into T_{j+1} ; by induction $U \subseteq \text{maxRAF}(S)$. At stabilization every surviving reaction has its reactants and a catalyst in the closure of the surviving set, which is the RAF condition when the set is nonempty. $\square$

Lemma 2.4 (one-deletion test). *A set $S \subseteq R$ is an irrRAF if and only if S is a RAF and $\text{maxRAF}(S \setminus \{r\}) = \emptyset$ for every $r \in S$. Consequently irreducibility is decidable in polynomial time, and every nonempty RAF contains an irrRAF.*

Proof. A nonempty $\text{maxRAF}(S \setminus \{r\})$ is a proper RAF subset of S . Conversely a proper RAF subset $T \subsetneq S$ omits some $r \in S$ and survives inside $S \setminus \{r\}$ by Lemma 2.3. The last claim follows by choosing an inclusion-minimal RAF subset. $\square$

Note that it is not enough to check that $S \setminus \{r\}$ itself fails to be a RAF: that set may still contain a smaller RAF. The maxRAF-based form is the correct polynomial test, both in proofs and in implementations.

2.2 The certification problems

Let $\mathcal{L}$ be a finite set of reaction subsets of $\mathcal{Q}$ and $k = |\mathcal{L}|$; representing the family as a set means that repeated entries do not count as separate supplied solutions. Define

$$\begin{aligned} \text{VALID}(\mathcal{Q}, \mathcal{L}) &: \iff \text{every member of } \mathcal{L} \text{ is an irrRAF of } \mathcal{Q}, \\ \text{COMPLETE}(\mathcal{Q}, \mathcal{L}) &: \iff \text{every irrRAF of } \mathcal{Q} \text{ belongs to } \mathcal{L}, \\ \text{EXACT}(\mathcal{Q}, \mathcal{L}) &: \iff \text{VALID}(\mathcal{Q}, \mathcal{L}) \wedge \text{COMPLETE}(\mathcal{Q}, \mathcal{L}), \\ \text{EXTRA}(\mathcal{Q}, \mathcal{L}) &: \iff \text{there is an irrRAF } J \text{ of } \mathcal{Q} \text{ with } J \notin \mathcal{L}. \end{aligned}$$

Problem 2.5 (EXACT). *Input: a CRS $\mathcal{Q}$ and a family $\mathcal{L}$ of reaction subsets, both explicitly encoded. Parameter: $k = |\mathcal{L}|$. Question: does EXACT($\mathcal{Q}, \mathcal{L}$) hold?*

EXACT is a total language: an invalid entry makes the answer no. On inputs satisfying the promise $\text{VALID}(\mathcal{Q}, \mathcal{L})$, the complement of EXACT is EXTRA. Two edge cases are handled by the definitions rather than excluded. If $k = 0$ then $\text{EXACT}(\mathcal{Q}, \emptyset)$ holds exactly when $\mathcal{Q}$ has no RAF at all, which is decidable in polynomial time by Lemma 2.3. If $\mathcal{Q}$ has no RAF then the only exact family is the empty one. The encoded input length N includes the family $\mathcal{L}$; in particular $k \leq N$ and $|R| \leq N$.

By Lemma 2.4, VALID is decidable in polynomial time. Completeness is where the difficulty lies, and it must be distinguished from weaker properties. A family may cover every reaction of the maxRAF and still be incomplete.

Example 2.6 (coverage is not completeness). Let $F = \{f\}$ and consider three reactions,

$$r_a: f \rightarrow a \text{ (catalysts } b, c), \quad r_b: f \rightarrow b \text{ (catalysts } a, c), \quad r_c: f \rightarrow c \text{ (catalysts } a, b).$$

No singleton is a RAF, since a lone reaction has no catalyst in its closure, while every pair is a RAF, and each pair is irreducible. The complete family therefore has three members. The family $\{\{r_a, r_b\}, \{r_a, r_c\}\}$ covers every reaction of the maxRAF $\{r_a, r_b, r_c\}$ but omits $\{r_b, r_c\}$. Deleting the shared reaction r_a from both listed sets, as the deletion criterion of Section 3 prescribes, leaves exactly the missing irrRAF. (These statements were confirmed by enumerating all 7 nonempty reaction subsets under Definition 2.1.)

2.3 Parameterized complexity

We use the standard framework [6, 9, 7]. A parameterized problem is a language of pairs (x, k) . It is in **FPT** if it is decidable in time $f(k) |x|^{O(1)}$ for some computable f , and in **XP** if it is decidable in time $|x|^{g(k)}$ for some computable g . An *FPT reduction* from (A, κ) to (B, κ') is a map $x \mapsto x'$ computable in time $f(\kappa(x)) |x|^{O(1)}$ with $x \in A \iff x' \in B$ and $\kappa'(x') \leq g(\kappa(x))$ for computable f, g . The class **W[P]** is the class of parameterized problems FPT-reducible to **WEIGHTED CIRCUIT SATISFIABILITY** (does a Boolean circuit have a satisfying assignment of Hamming weight exactly k ?), and $\mathbf{W}[1] \subseteq \mathbf{W}[P]$. A problem is **W[P]-hard** if every problem in **W[P]** FPT-reduces to it, and **co-W[P]** denotes the class of complements of **W[P]** problems. We use the following machine characterization.

Theorem 2.7 (bounded nondeterminism; [3], see [9, Chapter 3]). *A parameterized problem is in **W[P]** if and only if it is decided by a nondeterministic algorithm running in time $f(k) |x|^{O(1)}$ that makes at most $h(k) \log |x|$ nondeterministic binary choices, for computable f and h .*

Our source problem is the following [11].

Problem 2.8 (MINIMUM AXIOM SET). *Input:* a finite set U of statements, a finite list of rules (P_j, u_j) with $P_j \subseteq U$ and $u_j \in U$, and an integer $k \geq 0$. *Parameter:* k . *Question:* is there a set $B \subseteq U$ with $|B| \leq k$ that *generates* U , that is, such that every statement of U is obtained from B by repeated application of the rules?

Formally, the set of statements derivable from B is the least set $D \supseteq B$ such that $u_j \in D$ whenever $P_j \subseteq D$; equivalently, u is derivable if there is a finite derivation tree with leaves in B and internal nodes labelled by rules. Empty premise sets $P_j = \emptyset$ and cyclic rule systems are allowed. MINIMUM AXIOM SET is NP-complete (it appears in the logic section of the catalogue of [11]) and **W[P]-complete** parameterized by k [1, Theorem 3.3].

Finally, the Exponential Time Hypothesis (ETH) of Impagliazzo, Paturi and Zane [17] asserts that 3-SAT on n variables has no $2^{o(n)}$ -time algorithm. Chen, Huang, Kanj and Xia [4] proved that, unless ETH fails, k -CLIQUE on n -vertex graphs has no algorithm running in time $f(k) n^{o(k)}$ for any computable f .

3 The deletion criterion and the upper bounds

The following criterion is Theorem 2 of Steel, Hordijk and Smith [19]. We include the short proof because the proof of the main theorem reuses its two directions, and because its formal statement records the witness as a function from listed sets to reactions, whose image may have fewer than k elements.

Lemma 3.1 (preserving transversal; [19, Theorem 2]). *Let $\mathcal{L} = \{I_1, \dots, I_k\}$ satisfy $\text{VALID}(\mathcal{Q}, \mathcal{L})$. Then $\text{EXTRA}(\mathcal{Q}, \mathcal{L})$ holds if and only if there are reactions $r_i \in I_i$ ($1 \leq i \leq k$) such that $\text{maxRAF}(R \setminus \{r_1, \dots, r_k\}) \neq \emptyset$.*

Proof. Let J be an unlisted irrRAF. For each i , $I_i \not\subseteq J$, for otherwise irreducibility of J would give $J = I_i \in \mathcal{L}$. Choose $r_i \in I_i \setminus J$. Then $J \subseteq R \setminus \{r_1, \dots, r_k\}$, so the residual contains a RAF. Conversely, if the residual contains a RAF, let J be an irrRAF inside it (Lemma 2.4). For every i , $r_i \in I_i \setminus J$, so $J \neq I_i$, and J is unlisted. The argument is valid for $k = 0$: the empty tuple preserves a RAF exactly when $\mathcal{Q}$ has one. $\square$

Different coordinates of the tuple may select the same reaction, so the deleted set can have fewer than k elements; no disjointness of the listed sets is assumed. Lemma 3.1 yields the algorithm of [19]: test validity by Lemma 2.4, then run the maxRAF oracle on the residual of each of the $\prod_i |I_i|$ tuples. With $\ell = \max_i |I_i|$ this takes time $\ell^k N^{O(1)} \leq N^{k+O(1)}$, which places

EXACT in XP and, as recorded in Proposition 7.5, in FPT for the combined parameter $k + \ell$. The exponent depends on k ; the question is whether it must.

Proposition 3.2 (membership). *On inputs satisfying VALID, EXTRA is in NP and in W[P]. On all inputs, EXACT is in coNP and in co-W[P].*

Proof. For EXTRA on valid inputs, guess one reaction from each listed set, check the k memberships, and run the maxRAF oracle on the residual; accept if it is nonempty. Correctness is Lemma 3.1. The guess uses at most $k \lceil \log_2(|R| + 1) \rceil \leq (k + 1) \log_2 N + k$ nondeterministic bits and the rest is deterministic polynomial time, so Theorem 2.7 gives membership in W[P]; the same guess is an ordinary polynomial-size NP witness. Alternatively, an unlisted irrRAF J itself is an NP witness, verified by Lemma 2.4 and a comparison with the list.

For the total problem, the complement of EXACT is $\neg \text{VALID} \vee \text{EXTRA}$. A verifier first tests validity deterministically in polynomial time and accepts if it fails; otherwise it runs the EXTRA verifier. This preserves both the polynomial running time and the bound on nondeterministic bits, so the complement of EXACT is in $\text{NP} \cap \text{W[P]}$ and EXACT is in $\text{coNP} \cap \text{co-W[P]}$. $\square$

None of the ingredients of this section is new; they are the standard upper bound. The contribution of the paper is the matching lower bound.

4 The reduction from Minimum Axiom Set

4.1 Source normalization

Let $\mathcal{A} = (U, (P_j, u_j)_{j < q}, k)$ be an instance of MINIMUM AXIOM SET with $m = |U|$ statements and q rules. We may assume $m \geq 2$ and identify U with $\{0, \dots, m - 1\}$: if $m < 2$, add fresh statements together with empty-premise rules deriving them, which changes neither the derivability of the original statements nor the minimum number of axioms. For the classical reduction we may further assume $k \leq m$, since $k > m$ makes $B = U$ a solution and the instance can be mapped to a fixed yes-instance. The parameterized construction below is defined for every $k \geq 0$; the classical normalization is used only in the running-time analysis of Section 6.

The construction uses k slots. A slot assignment is a function $a: \{1, \dots, k\} \rightarrow U$; it represents the axiom set $B(a) = \{a(1), \dots, a(k)\}$, which has at most k elements and may have fewer when slots repeat. Every set B with $|B| \leq k$ is contained in some $B(a)$ because $U \neq \emptyset$, and derivability is monotone in the initial set. Hence:

Lemma 4.1 (slots represent small sets). *$\mathcal{A}$ has a generating set of size at most k if and only if some slot assignment a has $B(a)$ generating U . For $k = 0$ both sides refer to the empty set.*

4.2 The constructed system

Construction 4.2. Given $\mathcal{A}$ as above, the CRS $\mathcal{Q}(\mathcal{A})$ has the food set $F = \{f\}$ and the molecules and reactions of Table 1. Subscripts $u + 1$ are taken cyclically modulo m . The catalysis relation is: z catalyses every reaction; $c_{i,u+1}$ catalyses $s_{i,u}$; there are no other catalytic pairs. The supplied family is

$$\mathcal{L}(\mathcal{A}) = \{I_1, \dots, I_k\}, \quad I_i = \{s_{i,u} : u \in U\}.$$

Each reaction family has one role. Selectors form the listed cycles and emit the signals $x_{i,u}$. Decoders read a slot: $d_{i,u}$ can fire only if every selector of slot i other than $s_{i,u}$ is present, and it announces statement u as an axiom. Implication reactions copy the rules of $\mathcal{A}$ on statement molecules; the food reactant f in p_j makes empty-premise rules well defined. The closing reaction t verifies that every slot has been decoded and every statement has been derived, and

Molecule	Index range	Count	Role
f		1	the only food molecule
$x_{i,u}, c_{i,u}$	$i \in [k], u \in U$	$2km$	signal and cyclic catalyst produced by selector $s_{i,u}$
b_i	$i \in [k]$	k	slot marker: some decoder of slot i fired
A_u	$u \in U$	m	statement u has been derived
z		1	common catalyst of all non-selector reactions

Reaction	Reactants $\rightarrow$ products	Count	Catalysts
$s_{i,u}$ (selector)	$\{f\} \rightarrow \{x_{i,u}, c_{i,u}\}$	km	$c_{i,u+1}$ and z
$d_{i,u}$ (decoder)	$\{x_{i,v} : v \in U, v \neq u\} \rightarrow \{b_i, A_u\}$	km	z
p_j (implication)	$\{f\} \cup \{A_v : v \in P_j\} \rightarrow \{A_{u_j}\}$	q	z
t (closing)	$\{b_i : i \in [k]\} \cup \{A_u : u \in U\} \rightarrow \{z\}$	1	z

Table 1: The system $\mathcal{Q}(\mathcal{A})$ of Construction 4.2: $M = 2 + 2km + k + m$ molecules and $H = 2km + q + 1$ reactions.

pays for that verification by producing z , the only catalyst available to anything but a selector. Three provenance facts are immediate from Table 1 and are used repeatedly: $x_{i,u}$ and $c_{i,u}$ are produced only by $s_{i,u}$; b_i is produced only by decoders of slot i ; z is produced only by t .

Example 4.3 (a worked instance). Take $U = \{0, 1, 2\}$ with the cyclic rules $\{0\} \Rightarrow 1$, $\{1\} \Rightarrow 2$, $\{2\} \Rightarrow 0$ and $k = 1$. Any single statement generates U , so the instance is a yes-instance. The system $\mathcal{Q}(\mathcal{A})$ has 12 molecules and 10 reactions: three selectors, three decoders, three implication reactions and t . Enumerating all $2^{10} - 1$ nonempty reaction subsets under Definition 2.1 finds 33 RAFs and exactly four irrRAF: the listed cycle $I_1 = \{s_{1,0}, s_{1,1}, s_{1,2}\}$ and the three unlisted sets

$$\{s_{1,1}, s_{1,2}, d_{1,0}, p_0, p_1, t\}, \quad \{s_{1,0}, s_{1,2}, d_{1,1}, p_1, p_2, t\}, \quad \{s_{1,0}, s_{1,1}, d_{1,2}, p_2, p_0, t\},$$

one for each choice of the omitted selector, that is, of the axiom. In each unlisted irrRAF the implication reaction that would close the cycle of rules is absent, as irreducibility demands. With $k = 0$ the same rules give a system whose only candidate for the closing reaction is $t: \{A_0, A_1, A_2\} \rightarrow \{z\}$ with no producer of any A_u , and indeed the system has no RAF at all, matching the fact that the empty set generates nothing. With the single rule $\{0\} \Rightarrow 1$ instead, the instance needs two axioms: for $k = 1$ the only irrRAF is the listed cycle, while for $k = 2$ the two unlisted irrRAF correspond to the slot assignments $(0, 2)$ and $(2, 0)$.

5 Correctness of the reduction

Throughout this section $\mathcal{A}$ is a normalized instance, $\mathcal{Q} = \mathcal{Q}(\mathcal{A})$, $\mathcal{L} = \mathcal{L}(\mathcal{A})$, and closures are taken in $\mathcal{Q}$.

Lemma 5.1 (guards). *Each I_i is an irrRAF of $\mathcal{Q}$, the family $\mathcal{L}$ has exactly k members, and every irrRAF of $\mathcal{Q}$ that does not contain t is one of the I_i .*

Proof. Every selector consumes only food, so all products of I_i lie in $K_1(I_i)$; in particular $c_{i,u+1} \in \text{cl}(I_i)$ catalyses $s_{i,u}$, and I_i is a RAF. Let $H \subseteq I_i$ be a nonempty RAF. Since $t \notin H$ and t is the only producer of z , we have $z \notin \text{cl}(H)$, so each $s_{i,u} \in H$ must be catalysed by $c_{i,u+1} \in \text{cl}(H)$, whose only producer is $s_{i,u+1}$; hence $s_{i,u+1} \in H$. Following the cyclic successor around the m -cycle from any member of H gives $H = I_i$. Thus I_i is irreducible. The sets I_i are pairwise disjoint and nonempty, so $|\mathcal{L}| = k$.

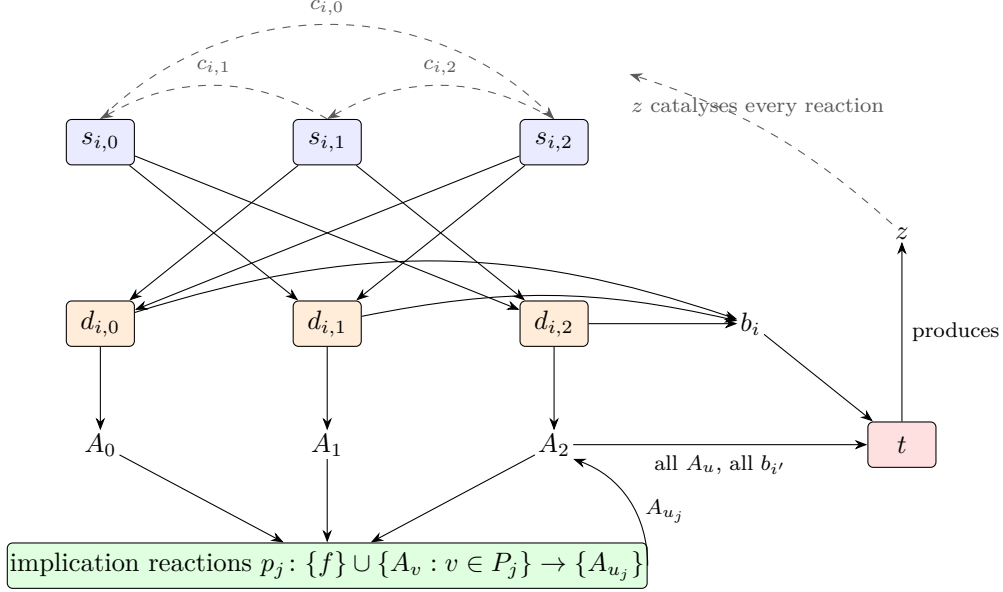


Figure 1: One slot of Construction 4.2 for $m = 3$. Solid arrows are reactant/product incidences (every selector also consumes the food molecule f , not drawn), dashed arrows are catalysis. The selector cycle $I_i = \{s_{i,0}, s_{i,1}, s_{i,2}\}$ is the listed irrRAF of the slot; the other slots $i' \neq i$ carry further cycles feeding the same closing reaction t . Decoder $d_{i,u}$ needs the signals $x_{i,v}$ of all selectors except $s_{i,u}$, so in an unlisted irrRAF, which omits exactly one selector per slot, the decoder that fires names the omitted statement as an axiom.

Now let J be an irrRAF with $t \notin J$. Then $z \notin \text{cl}(J)$, and since every non-selector reaction has z as its only catalyst, J consists of selectors. Take $s_{i,u} \in J$; the successor argument above, applied to J , shows $I_i \subseteq J$, and irreducibility of J together with I_i being a RAF gives $J = I_i$. $\square$

Lemma 5.2 (omission pattern). *Let J be an irrRAF of $\mathcal{Q}$ with $J \notin \mathcal{L}$. Then $t \in J$; for every slot i there is a unique $a(i) \in U$ such that*

$$s_{i,u} \in J \iff u \neq a(i) \quad (u \in U);$$

and every decoder $d_{i,u} \in J$ has $u = a(i)$.

Proof. By Lemma 5.1, $t \in J$. Fix a slot i . Since I_i is a RAF and J is irreducible with $J \neq I_i$, we have $I_i \not\subseteq J$: some selector of slot i is missing from J . On the other hand $t \in J$ is food-generated, so $b_i \in \text{cl}(J)$, and b_i is produced only by decoders of slot i ; hence some $d_{i,u} \in J$. This decoder is food-generated, so $x_{i,v} \in \text{cl}(J)$ for every $v \neq u$, and $x_{i,v}$ is produced only by $s_{i,v}$; hence $s_{i,v} \in J$ for all $v \neq u$. Thus at most one selector of slot i is missing, and combined with the first observation exactly one is: call it $s_{i,a(i)}$, so that $u = a(i)$ for the decoder just considered. Finally, any decoder $d_{i,u'} \in J$ with $u' \neq a(i)$ would need $x_{i,a(i)} \in \text{cl}(J)$, whose only producer $s_{i,a(i)}$ is absent from J ; this contradicts food generation. For $k = 0$ the slot statements are vacuous and only $t \in J$ is asserted. $\square$

All three arguments track the *producer* of a molecule that appears in a staged closure, not merely the set of reactions that list it as a product. This is what prevents a signal produced by an unrelated reaction from being misread as a chosen axiom, and it is why no acyclicity assumption on the implication system is needed.

Theorem 5.3 (reduction correctness). *$\mathcal{A}$ has a generating set of size at most k if and only if $\text{EXTRA}(\mathcal{Q}(\mathcal{A}), \mathcal{L}(\mathcal{A}))$ holds. In words: the source instance is a yes-instance if and only if the supplied family is incomplete.*

Proof. Forward direction. By Lemma 4.1 choose a slot assignment a with $B(a)$ generating U . Let

$$S(a) = \{s_{i,u} : u \neq a(i)\} \cup \{d_{i,a(i)} : i \in [k]\} \cup \{p_j : j < q\} \cup \{t\}.$$

We show that $S(a)$ is a RAF. Food generation: selectors consume only f . Every reactant $x_{i,v}$, $v \neq a(i)$, of the decoder $d_{i,a(i)}$ is produced at stage 1 by $s_{i,v} \in S(a)$, so the decoders fire at stage 2 and produce every b_i and every $A_{a(i)}$. We claim that every derivable statement u has $A_u \in \text{cl}(S(a))$; by induction on a derivation of u from $B(a)$: if $u \in B(a)$ then $u = a(i)$ for some i and $A_u \in K_2(S(a))$; if $u = u_j$ is obtained by rule j from premises in P_j , then by induction each A_v , $v \in P_j$, lies in some stage, and since stages increase they all lie in a common stage $K_s(S(a))$, together with f , so p_j fires and $A_{u_j} \in K_{s+1}(S(a))$. As $B(a)$ generates U , every A_u lies in $\text{cl}(S(a))$. Hence every p_j is food-generated, all reactants of t are in $\text{cl}(S(a))$, t fires, and $z \in \text{cl}(S(a))$. Reflexive autocatalysis: z catalyses every reaction. So $S(a)$ is a nonempty RAF. Let $J \subseteq S(a)$ be an inclusion-minimal RAF (Lemma 2.4). For each i , $s_{i,a(i)} \notin S(a) \supseteq J$, so $J \neq I_i$; thus J is an unlisted irrRAF.

Reverse direction. Let J be an unlisted irrRAF and let a be the assignment of Lemma 5.2. We prove by induction on the stage h that

$$A_u \in K_h(J) \implies u \text{ is derivable from } B(a).$$

For $h = 0$, $K_0(J) = \{f\}$ contains no statement molecule. For the step, suppose $A_u \in K_{h+1}(J) \setminus K_h(J)$. Then A_u is a product of some reaction $r \in J$ whose reactants lie in $K_h(J)$. Selectors and t produce no statement molecules. If $r = d_{i,u}$ then $u = a(i) \in B(a)$ by Lemma 5.2. If $r = p_j$ then $u = u_j$ and $A_v \in K_h(J)$ for every $v \in P_j$; by the induction hypothesis each such v is derivable from $B(a)$, and rule j derives u_j . This exhausts the reaction types. Since $t \in J$ is food-generated, every A_u lies in some $K_h(J)$, so every statement is derivable from $B(a)$, a set of size at most k . $\square$

Cyclic rules do not create unsupported derivations in the reverse direction: a statement molecule enters the closure at a definite first stage through a decoder, which names an axiom, or through an implication whose premises entered earlier. Nor does the order of catalyst appearance matter: z appears only after t fires, which is after all statement molecules are present, and Remark 2.2 is exactly what permits this.

6 Size, construction time and the classification

Proposition 6.1 (size). *$\mathcal{Q}(\mathcal{A})$ has exactly $M = 2 + 2km + k + m$ molecules and $H = 2km + q + 1$ reactions. A dense Boolean encoding consisting of a food-membership vector, three molecule-by-reaction incidence matrices (reactants, products, catalysis) and k reaction-membership rows for the listed family has exactly $E = M + (3M + k)H$ cells. The supplied family has exactly k distinct members, all irrRAFs.*

Proof. Count Table 1; the last sentence is Lemma 5.1. The three identities are also Lean theorems (Section 8). $\square$

Proposition 6.2 (construction time). *The map $\mathcal{A} \mapsto (\mathcal{Q}(\mathcal{A}), \mathcal{L}(\mathcal{A}))$ is computable in time polynomial in $k + m + q + L_{\mathcal{A}}$, where $L_{\mathcal{A}}$ is the encoded length of the rule lists. On classically normalized inputs ($k \leq m$) this is polynomial in the input length; on arbitrary inputs it is bounded by a computable function of k times a polynomial in the input length, and the output parameter equals k .*

Proof. Enumerate the index sets of Table 1 and decide each of the $E = O((km + m)(km + q))$ cells of the dense encoding directly from the defining formulas: membership of x in $\text{in}(r)$, $\text{out}(r)$ or as a catalyst of r is a constant number of index comparisons, one cyclic successor computation, or a scan of one premise list P_j of length at most L_A . Headers and index names add polynomial overhead. No closure computation, no search for a generating set, and no RAF computation is performed. The parameter claim is Proposition 6.1. Dimension headers may be written in unary so that k contributes at most polynomially to the output length. $\square$

Theorem 6.3 (classification).

- (i) EXTRA restricted to valid inputs is NP-complete, and EXACT is coNP-complete.
- (ii) Parameterized by $k = |\mathcal{L}|$, EXTRA restricted to valid inputs is W[P]-complete and EXACT is co-W[P]-complete.
- (iii) Hardness in (i) and (ii) holds on inputs $(\mathcal{Q}, \mathcal{L})$ in which $\mathcal{Q}$ has one food molecule, $\mathcal{L}$ consists of exactly k pairwise reaction-disjoint irrRAFs each of which is a simple catalytic cycle of reactions with the single reactant f and two catalysts, and every other reaction has a single catalyst z .

Proof. Upper bounds are Proposition 3.2. For the lower bounds compose the normalization of Section 4.1 with Construction 4.2. By Theorem 5.3 the map sends yes-instances of MINIMUM AXIOM SET to instances satisfying EXTRA and no-instances to instances satisfying COMPLETE; all outputs satisfy VALID by Lemma 5.1, so EXTRA and $\neg$ EXACT coincide on them. By Proposition 6.2 the map is a polynomial-time many-one reduction on classically normalized inputs, giving NP-hardness of EXTRA on valid inputs from the NP-completeness of MINIMUM AXIOM SET; and it is an FPT reduction with output parameter exactly k , giving W[P]-hardness from [1, Theorem 3.3]. Taking complements, EXACT is coNP-hard and co-W[P]-hard, already on valid inputs. The restrictions in (iii) are read off Table 1: the listed sets are the pairwise disjoint cycles I_i , each selector has reactant set $\{f\}$ and catalysts $c_{i,u+1}$ and z , and every other reaction is catalysed only by z . $\square$

Remark 6.4 (what is and is not restricted). Reactant arity is unbounded in the hard instances: decoders have $m - 1$ reactants and the closing reaction has $k + m$. The implication subsystem may contain cycles. We make no bounded-arity or acyclic-verifier hardness claim; see Section 9. The listed irrRAFs all have size m , which is unbounded, as Proposition 7.5 requires.

7 Consequences and boundaries

7.1 Reading the classification

Table 2 states what each established claim licenses. Here N is the full encoded input length and $\text{FPT} \neq \text{W[P]}$, $\text{P} \neq \text{NP}$ and ETH are the respective standard assumptions. The distinction between XP and FPT is the whole content of the 2013 question: fixing the number of listed sets leaves a polynomial whose degree grows with that number, whereas fixed-parameter tractability demands a degree independent of it.

We do not infer a counting classification (no counting reduction is given), average-case hardness (the results are worst-case), or any statement about kinetic persistence or thermodynamic feasibility (the results concern the combinatorial RAF condition only).

7.2 An exponent lower bound under ETH

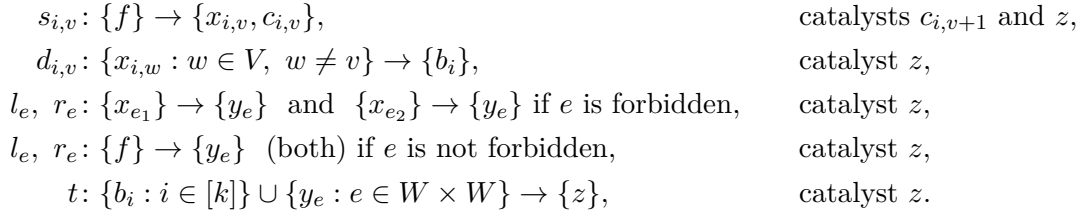
W[P]-hardness by itself yields no quantitative lower bound on the exponent. A second reduction, from k -CLIQUE, does. It predates the MINIMUM AXIOM SET reduction in our development,

Established claim	Appropriate interpretation
EXACT is coNP-complete (Thm. 6.3)	A polynomial-time completeness test for general systems would imply $P = NP$.
EXACT is co-W[P]-complete in k (Thm. 6.3)	A test running in time $f(k) N^c$ with c independent of k would imply $FPT = W[P]$; this answers the question of [19] negatively under that assumption.
No $f(k) N^{o(k)}$ algorithm under ETH (Cor. 7.3)	The linear dependence of the exponent on k in the deletion procedure cannot be improved to sublinear.
FPT in $k + \ell$ (Prop. 7.5)	When the listed irrRAFs are small, completeness is tractable for moderately many of them.
Polynomial verification of an unlisted irrRAF (Lemma 2.4)	A discovered missing member can be checked efficiently; this is not an efficient certificate that none exists.
Exact test in XP (Lemma 3.1)	Exhaustiveness is decidable, and practical for small k , even though the worst-case running time is large.

Table 2: Interpretation of the results at the strength proved.

uses the same selector cycles and closing reaction, and is included in the Lean development (Section 8); we describe it and give the proof in outline, since its two decisive lemmas are the same as Lemmas 5.1 and 5.2.

Construction 7.1 (clique guards). Let G be a simple graph on the vertex set $V = \{0, \dots, n-1\}$, $n \geq 2$, and let $k \geq 0$. Put $W = [k] \times V$ and call an ordered pair $e = ((i, u), (j, v)) \in W \times W$ *forbidden* if $i \neq j$ and $uv \notin E(G)$; in particular two copies of the same vertex in different slots form a forbidden pair. The CRS $\mathcal{Q}(G, k)$ has food $\{f\}$; molecules f, x_w, c_w ($w \in W$), b_i ($i \in [k]$), y_e ($e \in W \times W$) and z ; and the reactions below, where $v + 1$ is taken modulo n :



The supplied family is $\{I_1, \dots, I_k\}$ with $I_i = \{s_{i,v} : v \in V\}$.

Proposition 7.2. G has a clique of size k if and only if $\text{EXTRA}(\mathcal{Q}(G, k), \{I_1, \dots, I_k\})$ holds. The family is valid with exactly k members, and $\mathcal{Q}(G, k)$ has $2 + 2kn + k + (kn)^2$ molecules and $2kn + 2(kn)^2 + 1$ reactions.

Proof outline. Lemmas 5.1 and 5.2 hold verbatim, with the pair gates in the role of the implication reactions: they produce no signal, no cyclic catalyst, no marker and no z . Let J be an unlisted irrRAF with omission assignment $a : [k] \rightarrow V$. Since $t \in J$ is food-generated, every y_e is in $\text{cl}(J)$. For a forbidden pair $e = ((i, u), (j, v))$ this requires $x_{i,u}$ or $x_{j,v}$ in $\text{cl}(J)$, i.e. $u \neq a(i)$ or $v \neq a(j)$. Hence $a(i)a(j) \in E(G)$ for all $i \neq j$, and a is injective because equal vertices in distinct slots are forbidden; so $a([k])$ is a k -clique. Conversely, given a k -clique $\{a(1), \dots, a(k)\}$, the set consisting of all selectors $s_{i,v}$ with $v \neq a(i)$, the decoders $d_{i,a(i)}$, for each pair e the gates among l_e, r_e whose reactant is available, and t , is a RAF, and an inclusion-minimal RAF inside it is unlisted. The counts are read off the construction. $\square$

Corollary 7.3 (ETH lower bound). *Unless ETH fails, there is no algorithm deciding EXACT in time $f(k) N^{o(k)}$ for any computable f , even on inputs of the restricted form of Theorem 6.3(iii).*

Proof. Given (G, k) with $n \geq 2$ vertices, Construction 7.1 is computable in time polynomial in k and n , its output parameter is k , and by Proposition 7.2 its dense encoding has length

$N \leq c_0 k^4 n^4$ for a constant c_0 . Suppose EXACT were decidable in time $f(k) N^{h(k)}$ with $h(k) = o(k)$. Composing, k -CLIQUE would be decidable in time $(kn)^{O(1)} + f(k) (c_0 k^4 n^4)^{h(k)} = f_1(k) n^{4h(k)+O(1)} = f_1(k) n^{o(k)}$, contradicting [4]. The restricted form holds because $\mathcal{Q}(G, k)$ has the same guard structure as $\mathcal{Q}(\mathcal{A})$. $\square$

7.3 Finding one more irrRAF, and enumeration with bounded delay

Consider the search problem ANOTHER-IRRRAF: given $\mathcal{Q}$ and a valid family $\mathcal{L}$, return an irrRAF not in $\mathcal{L}$ or report correctly that none exists. Any algorithm for it decides EXTRA on valid inputs, so by Theorem 6.3 a polynomial-time algorithm would imply $P = NP$, and an $f(k) N^{O(1)}$ -time algorithm would imply $FPT = W[P]$. Moreover, the reverse direction of Theorem 5.3 is constructive: from any returned irrRAF the omitted selectors read off a generating axiom set of the source instance.

Corollary 7.4 (bounded-delay enumeration). *Suppose an algorithm enumerates, for every CRS $\mathcal{Q}$, all irrRAFs of $\mathcal{Q}$ without repetition, and that the delay before its first output, between consecutive outputs, and between its last output and its termination announcement is bounded by a polynomial $p(N)$. Then $P = NP$. If instead the delay after j outputs (including $j = 0$ and the termination announcement) is bounded by $f(j) N^c$ for a computable f and constant c , then $FPT = W[P]$.*

Proof. Given $(\mathcal{Q}, \mathcal{L})$, test validity in polynomial time and reject if it fails. Otherwise run the enumerator, comparing each output with $\mathcal{L}$, and stop at the first unlisted output (answer: incomplete) or at termination (answer: complete). On a valid list at most k listed outputs can precede the stopping event, so at most $k+1$ delays elapse, for a total time of $(k+1) p(N) N^{O(1)}$ in the first case, which is polynomial since $k \leq N$, and $(\sum_{j \leq k} f(j)) N^{O(1)}$ in the second. Theorem 6.3 gives the conclusions. $\square$

The corollary concerns delay bounds only. It does not exclude enumeration in total time polynomial in the input plus the entire output (output-polynomial enumeration), because the output may be exponentially larger than $\mathcal{L}$; that question remains open, and by [2] it is at least as hard as hypergraph dualization.

7.4 The positive boundary

Proposition 7.5 (FPT in $k + \ell$). *Let $\ell = \max_{I \in \mathcal{L}} |I|$ (with $\ell = 1$ if $\mathcal{L} = \emptyset$). EXACT is decidable in time $\ell^k N^{O(1)}$, hence fixed-parameter tractable in the combined parameter $k + \ell$.*

Proof. Lemma 3.1 with the maxRAF oracle, after a polynomial validity test. $\square$

The hard instances of Theorem 6.3 have $\ell = m$ unbounded, so the two results are consistent, and ℓ is the parameter that separates them. By contrast, structural parameters of the family alone do not help. In the hard instances the listed irrRAFs are pairwise disjoint: the overlap graph of $\mathcal{L}$ is edgeless, its treewidth and feedback vertex number are zero, and every reaction lies in at most one listed set. Restricting EXACT to any constant value of these parameters therefore leaves the problem coNP-hard and, in k , co-W[P]-hard. The number of connected components of the overlap graph is k itself in these instances, so hardness at a fixed number of components is not established by them.

8 Formal verification and computation

8.1 Lean formalization

The finite combinatorial content of Sections 3–6 and of Construction 7.1 has been formalized in Lean 4 (toolchain `leanprover/lean4:v4.30.0`) over a frozen Mathlib manifest. The

RAF semantics is the staged closure and the RAF predicate of Definition 2.1, shared with the other formal developments of this programme. The publication-facing module is `proofs/AllIrrRAFCert/Publication.lean`; its two exact-certification theorems, quoted here verbatim, quantify over an arbitrary finite implication system $A : \text{System } n \ q$ on the statement universe $\text{Fin } (n+2)$ with q rules and over an arbitrary decidable simple graph on $\text{Fin } (n+2)$, for every number k of slots including $k = 0$.

```
theorem axiom_iff_not_exactCertification {k n q : Nat}
  (A : WPHardness.System n q) :
  WPHardness.SmallGenerating A k <=>
    not ExactIrrRAFCertification (WPHardness.crs A) WPHardness.cat
    (WPHardness.listedFamily k n q)

theorem clique_iff_not_exactCertification {k n : Nat}
  (G : SimpleGraph (Fin (n+2))) [DecidableRel G.Adj] :
  (exists s : Finset (Fin (n+2)), G.IsNClique k s) <=>
    not ExactIrrRAFCertification (crs (graphBad G)) cat
    (listedFamily k n)
```

(The Unicode connectives of the source are transliterated here.)

Here `ExactIrrRAFCertification Q C L` is the conjunction of `ValidListedFamily Q C L` (every member of L is an irrRAF) and `AllCertified Q C L` (every irrRAF is in L), exactly as in Section 2.2; `SmallGenerating A k` asserts the existence of a finite set B with $B.\text{card} \leq k$ from which every statement is derivable under the inductively defined derivability relation; and `crs`, `cat` and `listedFamily` are the literal finite data of Constructions 4.2 and 7.1. Table 3 maps the statements of this paper to the proof-bearing declarations. The bundled theorem `axiom_reduction` in `Main.lean` additionally records that every listed member is an irrRAF and that the family has exactly k members.

The strict verification bridge compiles the publication module and its 36 imported dependencies with warnings treated as errors and authenticates the four exported declarations `axiom_iff_not_exactCertification`, `axiom_reduction`, `extra_iff_preservingTransversal` and `clique_iff_not_exactCertification`. The axiom audit of each reports only the standard axioms `propext`, `Classical.choice` and `Quot.sound`; no `sorry` and no custom axiom is admitted. The final successful run took 70 seconds. The receipt records the SHA-256 digests of the frozen Mathlib manifest (`a8df0b60...bb9fac`), of `Publication.lean` (`15fb4b80...ff3bcdcf`) and of every dependency; the full digests are in the reproduction record accompanying the source bundle.

Boundary of the formal claim. The Lean theorems establish the finite equivalences: for every source instance, a small generating set exists if and only if exact certification fails for the constructed instance, with the listed family valid and of the stated cardinality, and with the stated molecule, reaction and cell counts. They do not formalize Turing machines, complexity classes, the $W[P]$ -completeness of MINIMUM AXIOM SET, the ETH lower bound for CLIQUE, or the polynomial running time of the maxRAF oracle and of the construction. Those are the conventional arguments of Proposition 3.2, Proposition 6.2, Theorem 6.3 and Corollary 7.3, given in full above. No complexity-theoretic statement is hidden as a Lean axiom.

8.2 Exhaustive finite checks

Before formalization, Construction 4.2 was tested against the source predicate on every implication system over two statements: all 256 subsets of the 8 possible rules (including empty premises and cycles) for $k \in \{0, 1, 2\}$, that is 768 cases, of which 624 are yes-instances. For each case the listed cycles were confirmed to be irrRAFs and the existence of an unlisted

Statement in this paper	Lean declaration
Lemma 2.4 (one-deletion test)	<code>AllIrrRAFCert.irrRAF_single_deletion</code>
Lemma 3.1 (preserving transversal)	<code>AllIrrRAFCert.extra_iff_preservingTransversal;</code> <code>image bound preservingTransversal_card_le</code>
Lemma 4.1 (slots represent small sets)	<code>WPHardness.smallGenerating_iff_slots</code>
Lemma 5.1 (guards)	<code>WPHardness.guard_isIrrRAF, guards_injective,</code> <code>irrRAF_without_close_eq_guard</code>
Lemma 5.2 (omission pattern)	<code>WPHardness.extra_contains_close,</code> <code>extra_omission_pattern, decoder_choice</code>
Theorem 5.3, forward	<code>WPHardness.derivation_reaches, witness_isRAF,</code> <code>generating_implies_extra</code>
Theorem 5.3, reverse (stage induction)	<code>WPHardness.statement_sound,</code> <code>extra_implies_generating</code>
Theorem 5.3, equivalence	<code>WPHardness.smallGenerating_iff_extra; public</code> <code>form axiom_iff_not_exactCertification</code>
Proposition 6.1 (counts, cells, parameter)	<code>WPHardness.molecule_count, reaction_count,</code> <code>encoding_card, parameter_eq</code>
Dense encoding rows (Prop. 6.2)	<code>WPHardness.encode</code> with <code>encode_food,</code> <code>encode_inputs, encode_outputs, encode_catalysis,</code> <code>encode_list</code>
Proposition 7.2 (clique guards)	<code>Hardness.graph_clique_iff_extra,</code> <code>listed_promise, parameter_eq,</code> <code>encoding_polynomial; public form</code> <code>clique_iff_not_exactCertification</code>

Table 3: Concordance between the paper and the Lean development. Unqualified names in the middle rows live in the namespace `AllIrrRAFCert.WPHardness`; the clique row in `AllIrrRAFCert.Hardness`.

irrRAF, found by a maxRAF-based search, was compared with the existence of a generating slot assignment. No discrepancy was found; the replay takes under a second. An independent script that enumerates all reaction subsets literally under Definition 2.1, with no maxRAF pruning, repeated the 512 cases with $k \leq 1$ (finding 368 yes-instances, which together with the 256 trivially positive cases for $k = 2$ reproduces the count 624), a random sample of 12 systems for $k = 2$, a random sample of 40 systems over three statements with $k = 1$, and the instances of Examples 2.6 and 4.3, again without discrepancy. Construction 7.1 was tested on 4,176 small graph instances and 56 dense-source cases in the same way. These computations are diagnostics; the universal statements rest on the proofs and the Lean development.

9 Discussion

The 2013 deletion criterion gave an exact completeness test whose exponent grows with the number of listed irrRAFs, and asked whether that growth is necessary. Theorem 6.3 says that it is, unless $\text{FPT} = \text{W[P]}$, and Corollary 7.3 says that it cannot even be made sublinear unless ETH fails. The certification problem is thereby placed at the top of the W hierarchy, alongside MINIMUM AXIOM SET and WEIGHTED CIRCUIT SATISFIABILITY, rather than at the $\text{W}[1]$ level of CLIQUE. The mechanism is transparent: the listed cycles are rigid, so any additional irrRAF must sacrifice exactly one reaction in each of them, and a system of k such sacrifices is exactly a choice of k axioms. What makes the problem hard is not that there may be exponentially many irrRAFs, but that the omitted reactions of an unknown irrRAF can encode an arbitrary k -element choice, and that the closing reaction can check an arbitrary monotone derivation against it.

The results are structural statements about the combinatorial RAF condition. They say nothing about kinetics, thermodynamics, or whether a given irrRAF can actually establish itself from finite copy numbers; those questions require different models. Within the combinatorial setting, several questions remain.

1. *Bounded arity.* The decoders and the closing reaction of Construction 4.2 have unbounded reactant sets. Is EXACT still co-W[P] -hard, or even coNP -hard, when every reaction has at most two reactants? Chaining a decoder through $m - 2$ binary intermediate reactions preserves the provenance arguments but introduces new reactions that an unlisted irrRAF might use in unintended ways; we have not verified a bounded-arity variant.
2. *Counting.* Is counting the irrRAFs of a CRS $\#P$ -complete? Theorem 5.3 is not parsimonious: distinct slot assignments with the same axiom set give distinct irrRAFs, and an irrRAF uses only the implication reactions needed for one derivation.
3. *Output-polynomial enumeration.* Corollary 7.4 rules out polynomial delay unless $P = NP$. Whether all irrRAFs can be listed in time polynomial in the input and output lengths is open; by [2] it is at least as hard as hypergraph dualization, which is not known to be in polynomial time [10].
4. *Structural parameters of the system.* Proposition 7.5 identifies the size of the largest listed irrRAF as a parameter that restores tractability. Are there parameters of $\mathcal{Q}$ itself, such as the treewidth of its reaction graph or the number of catalysts per reaction, under which EXACT is fixed-parameter tractable in k plus that parameter? The hard instances have a single catalyst on most reactions and two on the rest, so the catalyst count alone does not suffice.

Data and code availability

The Lean sources, the verification receipt with all digests, the exhaustive-check scripts and the editable source of this paper are part of the accompanying research workspace and will be made available with the published version.

References

- [1] Karl A. Abrahamson, Rodney G. Downey, and Michael R. Fellows. Fixed-parameter tractability and completeness IV: On completeness for $W[P]$ and PSPACE analogues. *Annals of Pure and Applied Logic*, 73(3):235–276, 1995. doi:10.1016/0168-0072(94)00034-Z.
- [2] Anonymous. Minimum autocatalytic networks cannot be approximated within any constant factor. Companion manuscript, September 2026.
- [3] Liming Cai, Jianer Chen, Rodney G. Downey, and Michael R. Fellows. On the structure of parameterized problems in NP. *Information and Computation*, 123(1):38–49, 1995. doi:10.1006/inco.1995.1156.
- [4] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8):1346–1367, 2006. doi:10.1016/j.jcss.2006.04.007.
- [5] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, volume 12699 of *Lecture Notes in Computer Science*, pages 625–635. Springer, 2021. doi:10.1007/978-3-030-79876-5_37.

- [6] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, New York, 1999. doi:10.1007/978-1-4612-0515-9.
- [7] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, London, 2013. doi:10.1007/978-1-4471-5559-1.
- [8] Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995. doi:10.1137/S0097539793250299.
- [9] Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin, 2006. doi:10.1007/3-540-29953-X.
- [10] Michael L. Fredman and Leonid Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21(3):618–628, 1996. doi:10.1006/jagm.1996.0062.
- [11] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.
- [12] Wim Hordijk. Algorithms for enumerating irreducible and closed autocatalytic sets. Preprints.org, manuscript 202604.1787, 2026. preprints.org/manuscript/202604.1787.
- [13] Wim Hordijk, Joshua I. Smith, and Mike Steel. Algorithms for detecting and analysing autocatalytic sets. *Algorithms for Molecular Biology*, 10:15, 2015. doi:10.1186/s13015-015-0042-8.
- [14] Wim Hordijk and Mike Steel. Detecting autocatalytic, self-sustaining sets in chemical reaction systems. *Journal of Theoretical Biology*, 227(4):451–461, 2004. doi:10.1016/j.jtbi.2003.11.020.
- [15] Wim Hordijk and Mike Steel. Chasing the tail: The emergence of autocatalytic networks. *Biosystems*, 152:1–10, 2017. doi:10.1016/j.biosystems.2016.12.002.
- [16] Wim Hordijk, Mike Steel, and Stuart Kauffman. The structure of autocatalytic sets: Evolvability, enablement, and emergence. *Acta Biotheoretica*, 60(4):379–392, 2012. doi:10.1007/s10441-012-9165-1.
- [17] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.
- [18] Stuart A. Kauffman. Autocatalytic sets of proteins. *Journal of Theoretical Biology*, 119(1):1–24, 1986. doi:10.1016/S0022-5193(86)80047-9.
- [19] Mike Steel, Wim Hordijk, and Joshua Smith. Minimal autocatalytic networks. *Journal of Theoretical Biology*, 332:96–107, 2013. doi:10.1016/j.jtbi.2013.04.032.
- [20] Mike Steel, Wim Hordijk, and Joana C. Xavier. Autocatalytic networks in biology: structural theory and algorithms. *Journal of the Royal Society Interface*, 16(151):20180808, 2019. doi:10.1098/rsif.2018.0808.
- [21] Mike Steel, Joana C. Xavier, and Daniel H. Huson. The structure of autocatalytic networks, with application to early biochemistry. *Journal of the Royal Society Interface*, 17(171):20200488, 2020. doi:10.1098/rsif.2020.0488.

- [22] The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020)*, pages 367–381. ACM, 2020. doi:[10.1145/3372885.3373824](https://doi.org/10.1145/3372885.3373824).
- [23] Joana C. Xavier, Wim Hordijk, Stuart Kauffman, Mike Steel, and William F. Martin. Autocatalytic chemical networks at the origin of metabolism. *Proceedings of the Royal Society B*, 287(1922):20192377, 2020. doi:[10.1098/rspb.2019.2377](https://doi.org/10.1098/rspb.2019.2377).