The complexity of complete irreducible autocatalytic families: output-polynomial enumeration of irreducible RAFs if and only if $\mathsf{P} = \mathsf{NP}$

Anonymous manuscript

14 September 2026

Abstract

An irreducible reflexively autocatalytic and food-generated set (irreducible RAF, or irrRAF) is an inclusion-minimal nonempty RAF of a finite catalytic reaction system. Detecting whether a system contains a RAF is polynomial, and Steel, Hordijk and Smith (*J. Theor. Biol.*, 2013) gave a deletion-based criterion for certifying that a supplied list of irreducible RAFs is complete, whose cost grows exponentially in the list length. We settle the complexity of listing the complete family. For finite catalytic reaction systems with an explicit incidence encoding, a fixed deterministic multitape Turing machine that writes every irreducible RAF exactly once in time polynomial in the input length plus the output length exists if and only if $\mathsf{P} = \mathsf{NP}$. The lower bound is a polynomial-size SAT source whose *entire* RAF family has a prescribed normal form: a mandatory catalytic cycle forces all auxiliary reactions into every RAF, and a reset reaction cannot enable itself, so the irreducible RAFs are exactly n conflict pairs together with the satisfying assignments of the formula. Unsatisfiable formulas therefore have a known polynomial-size complete family, and an output-sensitive time bound becomes a polynomial deadline that decides satisfiability. The converse builds, under $\mathsf{P} = \mathsf{NP}$, a uniform enumerator from a polynomial-size supported-trace SAT encoding of the query "is there a RAF avoiding every known output?" and a filtered deletion minimizer. The same source gives an exact count identity $|\mathcal{I}(Q_\varphi)| = n + \#\textsc{Sat}(\varphi)$, coNP-completeness of exact certification, $\#\mathsf{P}$-hardness of counting, and an equivalence between polynomial-delay streaming enumeration and $\mathsf{P} = \mathsf{NP}$. The classification is machine-checked in Lean 4 for concrete multitape machines, including original-input initialization, arbitrary finite identifiers, and a count-prefixed fixed-width output, with no hypotheses beyond the standard axioms.

Keywords. autocatalytic sets; RAF theory; catalytic reaction systems; enumeration complexity; output-polynomial time; irreducible RAFs; satisfiability; formal verification.

MSC 2020. 68Q17, 68Q25, 92E20, 05C65, 68V20.

1 Introduction

Reflexively autocatalytic and food-generated sets (RAFs) are a finite combinatorial model of collectively self-sustaining reaction networks. The concept goes back to Kauffman's autocatalytic sets of proteins [20, 21], was given its present form by Hordijk and Steel [12], and has since developed into a structural theory with efficient algorithms [15, 16, 30, 33, 34] and applications to early metabolism and the origin of life [13, 14, 29, 38]. A catalytic reaction system consists of molecule types, reactions with reactants and products, 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 by reactions of the set, and each of its reactions is catalysed by a molecule that is food or is produced by the set.

Because unions of RAFs are RAFs, a system that contains a RAF contains a unique largest one, the *maxRAF*, and the RAF algorithm of Hordijk and Steel computes it in polynomial time [12]. The *minimal* autocatalytic subsystems point in the opposite direction. Steel, Hordijk and Smith [32] studied the irreducible RAFs (irrRAFs), which contain no proper RAF, and the smallest RAF overall. They proved that finding a RAF of minimum size is NP-hard, showed that a single irreducible RAF can be found in polynomial time by a deletion sweep, and gave a criterion for certifying that a supplied list of k irreducible RAFs is complete: the list is complete if and only if deleting one reaction from each listed set, in every possible way, always leaves a system with no RAF. That test needs up to $\prod_i |I_i|$ maxRAF computations, and they asked whether this dependence on k can be improved [32, Section 6.1]. Hordijk's recent preprint develops exhaustive and randomized methods for enumerating irreducible and closed autocatalytic sets [11]. The complete irreducible family is the natural object behind all of these questions: by an elementary argument (Proposition 7.4) it determines exactly which reaction deletions leave a RAF, and its members are the minimal structural cores that any downstream chemical or biological analysis begins from.

Finding one irreducible RAF and listing the complete family are different computational tasks. A complete family can be exponentially large, but that observation alone does not rule out an algorithm whose running time is polynomial in the input *and the output it must write*; this is the usual output-polynomial (or *output-sensitive*) standard for enumeration problems [18, 35]. The question this paper answers is whether such a guarantee can hold uniformly over all finite catalytic reaction systems.

1.1 Main result

Throughout, $Q = (M, R, F, \rho, \pi, C)$ is a finite catalytic reaction system with molecule set M, reaction identifier set R, food set $F \subseteq M$, reactant and product maps $\rho, \pi \colon R \to 2^M$, and catalysis relation $C \subseteq M \times R$. We write $\mathcal{I}(Q)$ for the family of its irreducible RAFs, and we fix the explicit encoding of Definition 3.1: the input is a dense bit string of length $N = 2d + r + 2 + 3dr$ for $d = |M|$ and $r = |R|$, and the required output for a family of K irreducible RAFs is a unary count followed by K reaction masks of width r, of total length $B = (r+1)K + 1$.

Theorem 1.1 (Classification)**.** *The following are equivalent.*

(i) *There exist a deterministic multitape Turing machine E with a fixed finite number of work tapes and a polynomial p such that, for every finite catalytic reaction system Q and every explicit finite labelling of its molecules and reactions, E started on the encoded input halts within $p(N + B)$ transitions with output exactly the encoding of $\mathcal{I}(Q)$, each member written exactly once.*

(ii) $\mathsf{P} = \mathsf{NP}$.

The quantifier order in (i) is $\exists E\, \exists p\, \forall Q$: one machine and one polynomial for every system. The machine receives only the encoded input; neither the family size nor a prepared formula is supplied as advice. Theorem 1.1 neither decides whether $\mathsf{P} = \mathsf{NP}$ nor asserts an unconditional efficient algorithm. It says that, under the standard assumption $\mathsf{P} \neq \mathsf{NP}$, no uniform output-polynomial enumerator of irreducible RAFs exists, and that this is exactly the obstruction: any polynomial-time SAT decider immediately yields such an enumerator.

The direction (i)⇒(ii) is stronger than an example with exponentially many outputs. It uses a polynomial *deadline* that is guaranteed to suffice on a hard class of *small-output* instances, so the difficulty is not that the family is large but that certifying its completeness requires solving SAT. The direction (ii)⇒(i) chooses one polynomial-time SAT machine and composes it with a uniform enumeration procedure whose every step, including reading the input, building the

formulas, storing the discovered family, and writing the final count-first output, is charged to the bound.

1.2 Proof overview

The proof has three mathematical components and one formal component.

A completion query. For a known family $\mathcal{G} \subseteq \mathcal{I}(Q)$ and a container $U \subseteq R$, ask whether some RAF inside U contains none of the known sets. This query is exact: it is satisfiable for $U = R$ if and only if an unknown irreducible RAF exists (Lemma 4.1), and it is monotone in the container. A polynomial-size *supported-trace* CNF expresses it (Lemma 4.2). The encoding tracks a witness of the least fixed point of food closure by one-way support implications, which makes it both simple and exact.

A SAT source with a known small complete family. Given a CNF φ with $n \geq 2$ variables, we build a polynomial-size system Q_φ (Construction 5.2) whose *entire* RAF family is $\{S(T) : H_\varphi(T)\}$, where T ranges over sets of literal choices and H_φ is a monotone predicate accepting a choice set exactly when it contains a conflicting pair or is a covering set that hits every clause. A mandatory catalytic cycle forces every auxiliary reaction into every RAF, and a first-enablement argument shows that a reset reaction cannot manufacture its own prerequisite. The minimal accepted choice sets are the n conflict pairs and the satisfying assignments (Lemma 5.1), so

$$\mathcal{I}(Q_\varphi) = \{S(P_i) : 1 \leq i \leq n\} \dot\cup \{S(T_\alpha) : \alpha \models \varphi\}, \qquad |\mathcal{I}(Q_\varphi)| = n + \#\textsc{Sat}(\varphi).$$

Unsatisfiable formulas therefore have a known complete family of polynomial encoded length B_0, and simulating any hypothetical enumerator for $p(N_\varphi + B_0)$ steps decides satisfiability (Theorem 5.7).

Filtered deletion. Under $\mathsf{P} = \mathsf{NP}$, each completion query is decided in polynomial time. Starting from the empty family, a single fixed-order deletion sweep that keeps the avoidance condition in force at every step produces a *new* irreducible RAF (Lemma 6.1); each successful round enlarges the family by one, and a final negative query certifies completeness. The total number of queries is at most $(r+1)K+1$ and each has polynomial size in $N+B$ (Theorem 6.2).

Formal realization. Every statement above is realized for concrete multitape Turing machines and checked in Lean 4 [7, 36]. The complexity classes P and NP and the Cook–Levin theorem [5, 24] are those of the `complexitylib` development [28]; the RAF semantics, the source construction, the supported-trace compiler, the count-based deadline, the filtered minimizer, an initializer that starts from the actual input tape, and a byte-preserving transport to arbitrary finite identifiers are new modules. The exported root theorem has no hypotheses and depends only on the axioms `propext`, `Classical.choice` and `Quot.sound` (Section 8).

1.3 Consequences

The exact family correspondence gives more than the classification. Section 7 derives, by conventional arguments from the verified finite semantics: exact certification of a supplied family is coNP-complete, and existence of an unlisted irreducible RAF is NP-complete on valid families (Proposition 7.1); counting irreducible RAFs is #P-complete under polynomial-time Turing reductions (Proposition 7.2); polynomial-delay polynomial-space streaming enumeration exists if and only if $\mathsf{P} = \mathsf{NP}$ (Proposition 7.3); the complete family determines survival under every reaction deletion set, and the reactions common to all irreducible RAFs are computable with r maxRAF calls without any enumeration (Propositions 7.4 and 7.5); and neither pairwise overlap of outputs nor low-order deletion response can serve as a structural indicator of hardness (Proposition 7.6). Section 9 describes a model-assisted enumerator that a SAT solver can run, proves that a completed run from g known outputs makes exactly $K - g + 1$ solver calls, and

reports a small bounded pilot. Section 10 discusses what the result does and does not say for chemical and biological applications and lists open problems.

1.4 Related work

The pattern "output-polynomial enumeration of the minimal solutions of an NP predicate would imply $\mathsf{P} = \mathsf{NP}$" is classical for maximal independent sets with side constraints [18, 23] and for maximal models of a Boolean formula [22]. The contribution here is not the pattern but its realization inside ordinary RAF semantics: RAF membership is a polynomial-time monotone-closure test, so hardness has to be manufactured from the interaction between catalysis and food generation, and the source must preserve the *whole* family, not merely existence. This is also what separates the enumeration question from the minimum-size question of [32], whose approximation hardness is treated in a companion manuscript [4].

Because the family $\mathcal{I}(Q)$ is exactly the set of minimal true points of the monotone viability predicate "U contains a RAF" (Section 2), the problem is an instance of identifying a monotone Boolean function from a membership oracle [2]. The classical dualization problem for explicitly given hypergraphs is quasi-polynomial [8, 10] and its exact status is open [26]; the present hardness does not bear on it, because a catalytic reaction system is a compact implicit description whose negative frontier can be exponentially larger than the positive one (Section 7.5). Blocking already-found minimal sets by a clause and re-solving is a standard technique for enumerating minimal unsatisfiable subsets [25]; we claim no novelty for generic blocking, only for the RAF-specific supported-trace correspondence and the exact exclusion of chemical witnesses.

The parameterized version of the certification question of [32], with the list length as parameter, is treated in a companion manuscript that proves co-W[P]-completeness by a different source [3]; the present result is the unparameterized classification of total enumeration. Compositional verification of concrete multitape machines has precedents in Coq [9]; our development is independent of that infrastructure and is described in Section 8.

1.5 Organization

Section 2 fixes RAF semantics and records the polynomial-time facts used throughout. Section 3 states the enumeration contract precisely. Section 4 gives the completion query and its CNF. Section 5 proves the lower bound and Section 6 the conditional upper bound. Section 7 collects consequences, Section 8 the formal verification, Section 9 the practical enumerator and pilot, and Section 10 the discussion. Appendix A gives exact formula sizes and a worked example; Appendix B gives reproduction information.

2 Catalytic reaction systems and irreducible RAFs

2.1 Closure and the RAF condition

Let $Q = (M, R, F, \rho, \pi, C)$ be as above. Reactions are directed and separately selectable; distinct identifiers remain distinct even when all their incidence data agree. The sets $\rho(a)$ and $\pi(a)$ carry no multiplicities or concentrations. For $S \subseteq R$ define the staged food closure

$$L_0(S) = F, \qquad L_{i+1}(S) = L_i(S) \cup \bigcup_{a \in S:\ \rho(a) \subseteq L_i(S)} \pi(a), \qquad \mathrm{cl}_S(F) = \bigcup_{i \geq 0} L_i(S). \tag{1}$$

The sequence is increasing and stabilizes by stage $d = |M|$: equality at one stage persists, and each strict increase adds a molecule. Closure is monotone in the reaction set, $S \subseteq T \Rightarrow \mathrm{cl}_S(F) \subseteq \mathrm{cl}_T(F)$. Catalysis does not constrain the generation steps in (1).

Definition 2.1 (RAF, irreducible RAF)**.** A set $S \subseteq R$ is a *RAF* of Q if it is nonempty and every $a \in S$ satisfies

$$\rho(a) \subseteq \mathrm{cl}_S(F) \qquad \text{and} \qquad \exists x \in \mathrm{cl}_S(F) : \ (x, a) \in C.$$

It is *irreducible* if no proper subset is a RAF. $\mathcal{I}(Q)$ denotes the family of irreducible RAFs of Q.

Catalysts are checked in the *final* closure; they are not prerequisites for a reaction to contribute products. With food f, the single reaction $f \to x$ catalysed by x is a RAF. This is the ordinary RAF condition of [12], not a temporal construction condition, and both constructions below depend on it.

Irreducibility is inclusion-minimality, not minimum cardinality; members of $\mathcal{I}(Q)$ can have different sizes, and Example 5.6 exhibits a system whose irreducible RAFs have two different sizes. Finite minimality gives that every RAF contains an irreducible RAF, and $\mathcal{I}(Q)$ is an antichain. The empty set is never a RAF, and $\mathcal{I}(Q)$ may be empty.

2.2 Polynomial-time facts

For $U \subseteq R$ let $\mathrm{maxRAF}_Q(U)$ be the union of all RAFs contained in U, the empty set if there is none. A nonempty union of RAFs is a RAF by monotonicity of closure. The RAF algorithm [12] computes it: start with $T = U$, compute $\mathrm{cl}_T(F)$, remove every reaction with a reactant or all catalysts outside the closure, and repeat until stable. Every RAF inside U survives every removal, so a nonempty terminal set is the maxRAF and an empty one certifies that U contains no RAF. There are at most r pruning rounds, each with at most d closure stages over explicit incidence tables, so the running time is polynomial in the explicit input.

Lemma 2.2 (Irreducibility is polynomially verifiable)**.** *For nonempty $I \subseteq R$,*

$$I \in \mathcal{I}(Q) \iff \mathrm{RAF}_Q(I) \ \wedge \ \forall a \in I : \ \mathrm{maxRAF}_Q(I \setminus \{a\}) = \varnothing.$$

Proof. If a proper RAF $J \subsetneq I$ exists, pick $a \in I \setminus J$; then J survives inside $I \setminus \{a\}$. Conversely a nonempty residual maxRAF is a proper RAF subset. □

Testing whether each single deletion $I \setminus \{a\}$ is itself a RAF would be wrong: with food $\{f\}$ and reactions $a\colon f \to x$ (catalyst f), $b\colon x \to y$ (catalyst z), $c\colon x \to z$ (catalyst y), the full set is a RAF and no two-element subset is, yet $\{a\}$ is a proper RAF. Residual pruning finds it.

2.3 Two monotonicities

The predicate "S is a RAF" is not upward closed: adding an unsupported reaction destroys it. The appropriate monotone predicate is viability of a container,

$$V_Q(U) \iff \exists S \subseteq U : \ \mathrm{RAF}_Q(S) \iff \mathrm{maxRAF}_Q(U) \neq \varnothing.$$

Its minimal true sets are exactly the irreducible RAFs: a minimal viable container equals any RAF witnessing its viability, and a viable proper subset of an irreducible RAF would contain a smaller RAF. Theorem 1.1 can thus be read as follows: a polynomial-time monotone membership oracle can have minimal true points whose complete enumeration is as hard as SAT, even when charged per output.

3 The enumeration contract

Definition 3.1 (Explicit encoding)**.** Fix bijections $M \to \{0, \dots, d-1\}$ and $R \to \{0, \dots, r-1\}$. The input is

$$1^d\, 0\; 1^r\, 0\;\; \mathrm{bits}(F)\;\; \mathrm{bits}(\rho, \pi, C), \tag{2}$$

where $\mathrm{bits}(F)$ is d bits and $\mathrm{bits}(\rho, \pi, C)$ is $3dr$ bits listing, in fixed index order, reactant, product and catalyst incidence for every reaction–molecule pair. The input length is

$$N = 2d + r + 2 + 3dr. \tag{3}$$

For a list of distinct sets $I_1, \dots, I_K$ whose underlying family is $\mathcal{I}(Q)$, the required output is

$$1^K\, 0\;\; \mathrm{mask}(I_1) \cdots \mathrm{mask}(I_K), \qquad B = (r+1)K + 1, \tag{4}$$

where each mask is the r-bit characteristic vector of I_j. Any order of the distinct masks is allowed; for $K = 0$ the output is the single delimiter 0.

An *output-polynomial enumerator* is a deterministic Turing machine E with one input tape, one output tape and a fixed finite number k of work tapes, together with a polynomial $p \in \mathbb{N}[X]$, such that for every valid input (2), E started in its ordinary initial configuration halts within $p(N + B)$ transitions with output tape content exactly (4). Statement (i) of Theorem 1.1 is the existence of such a pair (E, p). Reading the dimensions, termination on a no-RAF input, and writing the output are all charged to $p(N + B)$. The contract does not prescribe behaviour on malformed strings; the SAT decider extracted in the lower bound does handle malformed SAT encodings by a syntax guard.

The count precedes the masks, so this contract is about total time and does not by itself assert polynomial delay before the first mask. Proposition 7.3 treats the streaming variant separately.

4 Completion as an avoiding-RAF query

For a family $\mathcal{G} \subseteq \mathcal{I}(Q)$ of known outputs and a container $U \subseteq R$ define

$$\mathrm{Avail}_Q(\mathcal{G}, U) \iff \exists S \subseteq U :\; \mathrm{RAF}_Q(S) \;\wedge\; \bigwedge_{I \in \mathcal{G}} I \not\subseteq S. \tag{5}$$

The witness need only be a RAF; irreducibility is not imposed on it.

Lemma 4.1 (Exact completion query)**.** *For $\mathcal{G} \subseteq \mathcal{I}(Q)$: $\mathrm{Avail}_Q(\mathcal{G}, R)$ holds if and only if $\mathcal{I}(Q) \setminus \mathcal{G} \neq \varnothing$. For fixed $\mathcal{G}$, $\mathrm{Avail}_Q(\mathcal{G}, U)$ is monotone in U, and avoidance passes to subsets of a witness.*

Proof. Given an avoiding RAF S, choose an irreducible RAF $J \subseteq S$. If $J \in \mathcal{G}$, its containment in S contradicts avoidance. Conversely let $J \in \mathcal{I}(Q) \setminus \mathcal{G}$. No $I \in \mathcal{G}$ is contained in J, because the antichain property would force $I = J$; so $S = J$ is an avoiding witness. Monotonicity in U and heredity of avoidance are immediate. □

Writing $\mathcal{G} = \{I_1, \dots, I_g\}$, avoidance can equivalently be expressed by omitting one reaction from each known set:

$$\mathrm{Avail}_Q(\mathcal{G}, U) \iff \exists (a_1, \dots, a_g) \in I_1 \times \cdots \times I_g :\; \mathrm{maxRAF}_Q\big(U \setminus \{a_1, \dots, a_g\}\big) \neq \varnothing. \tag{6}$$

This is the existential form of the completeness criterion of [32, Theorem 2]. The choice domain is the whole I_j, not $I_j \cap U$: an element already outside U discharges its avoidance condition, and restricting the choices to the container would be incorrect. For $g = 0$ there is one empty tuple and the query is ordinary viability.

4.1 A polynomial-size supported-trace formula

The following clause pattern is the one consumed by the formal positive proof. Let s_a ($a \in R$) indicate selection, $z_{i,x}$ ($0 \le i \le d$, $x \in M$) indicate membership in a supported row at stage i, and $f_{i,a}$ ($0 \le i < d$) indicate a supported firing of a at stage i. An empty disjunction is false. The formula $\Psi(Q, \mathcal{G}, U)$ is the conjunction of

$$\bigvee_{a \in R} s_a; \qquad \neg s_a \qquad (a \notin U), \tag{7}$$

$$\neg z_{0,x} \qquad (x \notin F), \tag{8}$$

$$\neg f_{i,a} \vee s_a; \qquad \neg f_{i,a} \vee z_{i,x} \qquad (i < d,\ a \in R,\ x \in \rho(a)), \tag{9}$$

$$\neg z_{i+1,x} \vee z_{i,x} \vee \bigvee_{a:\, x \in \pi(a)} f_{i,a} \qquad (i < d,\ x \in M), \tag{10}$$

$$\neg s_a \vee z_{d,x} \qquad (a \in R,\ x \in \rho(a)), \tag{11}$$

$$\neg s_a \vee \bigvee_{x:\,(x,a) \in C} z_{d,x} \qquad (a \in R), \tag{12}$$

$$\bigvee_{a \in I} \neg s_a \qquad (I \in \mathcal{G}). \tag{13}$$

Nothing forces the food variables to be true, the rows to be increasing, or every enabled reaction to fire; these omissions are deliberate.

Lemma 4.2 (SAT correspondence)**.** *$\Psi(Q, \mathcal{G}, U)$ is satisfiable if and only if* $\mathrm{Avail}_Q(\mathcal{G}, U)$ *holds. It has $r + d(d+1) + dr$ variables and $O(1 + r + d + d^2 + d^2 r + dr + |\mathcal{G}|\, r)$ literal occurrences, and it can be written in time polynomial in N plus the explicit length of $\mathcal{G}$.*

Proof. Given a satisfying assignment, put $S = \{a : s_a = 1\}$ and $D_i = \{x : z_{i,x} = 1\}$. By (8), $D_0 \subseteq F = L_0(S)$. Suppose $D_i \subseteq L_i(S)$. If $x \in D_{i+1}$ then by (10) either $x \in D_i$ or some a with $x \in \pi(a)$ has $f_{i,a} = 1$, and then by (9) $a \in S$ and $\rho(a) \subseteq D_i \subseteq L_i(S)$, so $x \in L_{i+1}(S)$. Hence $D_i \subseteq L_i(S)$ for all $i \le d$. Now (7), (11) and (12) say that S is a nonempty subset of U whose reactants and some catalyst of every member lie in $D_d \subseteq \mathrm{cl}_S(F)$; so S is a RAF in U, and (13) gives avoidance.

Conversely, given an avoiding RAF $S \subseteq U$, set $s_a = [a \in S]$, $D_i = L_i(S)$, and $f_{i,a} = 1$ exactly when $a \in S$ and $\rho(a) \subseteq L_i(S)$. The recurrence (1) verifies (9) and (10); stabilization at stage d and the RAF conditions verify (11) and (12); (7), (8) and (13) hold by hypothesis. The size bounds follow by counting the rows; exact counts are in Appendix A. Even unary encodings of the polynomially many variable indices yield a polynomial-length instance, and nested finite scans of the incidence tables construct it in polynomial time. □

Only one-way support implications are needed: a satisfying assignment may omit generable molecules but cannot invent unsupported ones. A finite supported trace is an existential certificate of the least fixed point, and soundness needs only the invariant $D_i \subseteq L_i(S)$, not equality of every row with the true closure.

The exclusion clause (13) is attached to a *reaction set*, not to a satisfying assignment. Blocking an assignment would exclude one support history, and many assignments encode the same reaction set. The object enumerated is the irreducible reaction set, not its derivation.

5 The lower bound: a SAT source with a known small complete family

5.1 A monotone predicate whose minimal sets are conflict pairs and satisfying assignments

Let φ be a CNF over variables $1, \ldots, n$ with clauses $C_1, \ldots, C_m$; pad with unused variables if necessary so that $n \ge 2$ (padding preserves satisfiability and multiplies the number of satisfying

assignments by a known power of two). A *choice* (i, b) means that variable i takes value $b \in \{0, 1\}$; a clause is a set of choices. For a set T of choices define the monotone predicate

$$H_\varphi(T) \iff [\exists i : (i,0),(i,1) \in T] \ \vee \ [\forall i \, \exists b : (i,b) \in T \ \wedge \ \forall j : T \cap C_j \neq \varnothing]. \tag{14}$$

Write $P_i = \{(i,0),(i,1)\}$ for the ith *conflict pair* and $T_\alpha = \{(i, \alpha(i)) : 1 \le i \le n\}$ for the choice set of an assignment α.

Lemma 5.1 (Minimal accepted choice sets)**.** *For $n \ge 2$, the inclusion-minimal sets T with $H_\varphi(T)$ are exactly the n conflict pairs P_i and the sets T_α for satisfying assignments $\alpha \models \varphi$.*

Proof. Every accepted set has at least two elements, since a conflict needs two and coverage needs $n \ge 2$; hence each P_i is accepted and minimal. Let T be minimal accepted. If $T \supseteq P_i$ for some i then $T = P_i$. Otherwise T is conflict-free, so acceptance means coverage and clause hitting; consistency and coverage give exactly one choice per variable, so $T = T_\alpha$ with $\alpha \models \varphi$. Conversely T_α for $\alpha \models \varphi$ is accepted, and each proper subset lacks a variable while remaining conflict-free, so is not accepted. □

Negation is not implemented as chemical inhibition. Inconsistent selections are made *non-minimal* by placing a known smaller accepted set, the conflict pair, inside them.

5.2 The literal construction

Construction 5.2 (The source Q_φ)**.** Introduce $w = 3n+m+1$ *wires*: literal wires $\ell_{i,b}$, coverage wires v_i, clause wires c_j, and an output wire o. Use the signal rules of Table 1. A clause rule (j, i, b) with $(i, b) \notin C_j$ produces no wire but is retained, so there are exactly $q = 3n + 2nm + 1$ rules independently of the clause contents. Starting from the literal wires of a choice set T, forward application of the rules reaches o if and only if $H_\varphi(T)$ holds.

The system Q_φ has molecules $M = \{f\} \cup \{\text{wires}\} \cup \{a_0, \ldots, a_q\}$, food $F = \{f\}$, and reactions:

- an *input reaction* $\text{in}_{i,b}$: $f \to \ell_{i,b}$ for each choice (i, b), with sole catalyst a_q;
- for each rule $j < q$ an *auxiliary reaction* A_j consuming the required wires of rule j and producing its produced wires together with the marker a_j, with sole catalyst a_{j+1};
- a *reset* A_q consuming o and producing every wire together with a_q, with sole catalyst a_0.

No marker is food, and A_j is the only reaction producing a_j. Write $\mathcal{A} = \{A_0, \ldots, A_q\}$ and, for a choice set T,

$$S(T) = \mathcal{A} \cup \{\text{in}_{i,b} : (i,b) \in T\}.$$

The dimensions are

$$d_\varphi = w + q + 2 = 6n + 2nm + m + 4, \qquad r_\varphi = 2n + q + 1 = 5n + 2nm + 2, \tag{15}$$

so all incidence tables, and the input string (2), are written in time polynomial in the size of φ.

5.3 Exact RAF normal form

Lemma 5.3 (Mandatory cycle)**.** *Every RAF of Q_φ contains $\mathcal{A}$, hence equals $S(T)$ for a unique choice set T.*

Proof. A RAF is nonempty, and each of its reactions needs its unique catalyst marker in the food closure. A marker enters closure only through its unique producer, which must be selected. Thus some A_j is selected; the catalyst condition for A_j forces $A_{j+1 \bmod (q+1)}$, and iterating around the finite cycle forces all of $\mathcal{A}$. The input reactions are disjoint from $\mathcal{A}$, so T is unique. □

Rule kind	Index range	Required wires	Produced wires
Conflict	$i \in [n]$	$\ell_{i,0}, \ell_{i,1}$	o
Coverage	$(i,b) \in [n] \times \{0,1\}$	$\ell_{i,b}$	v_i
Clause	$(j,i,b) \in [m] \times [n] \times \{0,1\}$	$\ell_{i,b}$	c_j if $(i,b) \in C_j$, otherwise none
Finish	one rule	all v_i and all c_j	o

Table 1: The $q = 3n + 2nm + 1$ signal rules of Construction 5.2.

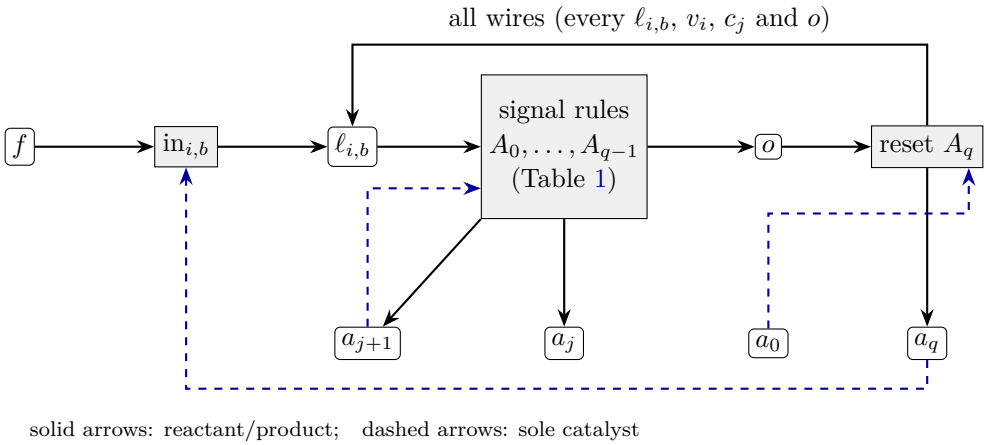

Figure 1: Schematic of Q_φ. Input reactions turn food into literal wires; the signal rules derive o exactly when H_φ accepts the selected choices; the reset turns o into every wire. Each reaction A_j produces its own marker a_j and is catalysed only by a_{j+1} (indices modulo $q+1$), and the input reactions are catalysed only by a_q; this single catalytic cycle forces every auxiliary reaction into every RAF.

Lemma 5.4 (The reset cannot enable itself)**.** *If $S(T)$ is a RAF then $H_\varphi(T)$ holds.*

Proof. Since the reset is selected and $S(T)$ is food-generated, $o \in \mathrm{cl}_{S(T)}(F)$. Consider the first closure stage i at which o appears. At all stages before i the reset is not enabled, so its products contribute nothing, and removing the reset leaves stages $0, \ldots, i$ unchanged. Hence o is derivable from the food by $S(T) \setminus \{A_q\}$. Reset-free generation is exactly forward application of the signal rules to the literal wires of T, which reaches o if and only if $H_\varphi(T)$. □

An equivalent closed-set argument: if $H_\varphi(T)$ fails, let W consist of food, all reset-free reachable wires, and all markers; $o \notin W$. Every non-reset reaction with reactants in W has products in W, and the reset has a reactant outside W, so $\mathrm{cl}_{S(T)}(F) \subseteq W$ and the reset is not food-generated. Admitting all markers into W is harmless because catalysts play no role in generation.

Lemma 5.5 (Exact normal form)**.** *The RAFs of Q_φ are exactly the sets $S(T)$ with $H_\varphi(T)$, and $S(T) \subseteq S(T')$ if and only if $T \subseteq T'$. Consequently*

$$\mathcal{I}(Q_\varphi) = \{S(P_i) : 1 \le i \le n\} \,\dot\cup\, \{S(T_\alpha) : \alpha \models \varphi\}, \qquad |\mathcal{I}(Q_\varphi)| = n + \#\textsc{Sat}(\varphi). \tag{16}$$

Proof. Lemmas 5.3 and 5.4 give one direction. Conversely, if $H_\varphi(T)$ holds, reset-free derivation reaches o, the reset then makes every wire available, every auxiliary reaction has its reactants in closure and produces its marker, and the selected input reactions consume food; all catalysts a_j are in closure, and the set is nonempty since it contains the reset. So $S(T)$ is a RAF. Inclusion of the sets $S(T)$ agrees with inclusion of choice sets because $\mathcal{A}$ is common and disjoint from

the inputs; applying the correspondence to every subset and Lemma 5.1 gives (16). The union is disjoint because $|T_\alpha| = n$ and T_α contains no conflict pair, so $T_\alpha \neq P_i$ even when $n = 2$. □

Preserving only existence of a RAF would not suffice for what follows; (16) describes the entire family. The mandatory cycle removes any freedom to realize the same choice set by different auxiliary subsets, which would otherwise change the number and identity of irreducible outputs. The reset is what makes every accepted $S(T)$ food-generated regardless of which branch of H_φ holds, and Lemma 5.4 is what prevents the auxiliary cycle from supporting a spurious RAF on its own.

Example 5.6 (A worked instance)**.** Take $n = 3$, $m = 2$. Then $q = 22$, $d_\varphi = 36$, $r_\varphi = 29$, and the input has $N = 3235$ bits. Every output has size $q + 1 + |T|$. For $\varphi = (x_1) \wedge (\neg x_1)$ there are no satisfying assignments: $\mathcal{I}(Q_\varphi)$ consists of the three conflict-pair sets, each of size 25, and the required output has $B = (29 + 1) \cdot 3 + 1 = 91$ bits. For $\varphi = (x_1) \wedge (x_2)$ over the same three-variable universe there are two satisfying assignments: $\mathcal{I}(Q_\varphi)$ has five members, three of size 25 and two of size 26, and $B = 151$. The size-generating polynomial of the family is $nz^{q+3} + \#\textsc{Sat}(\varphi)\, z^{q+n+1}$; for $n > 2$ the satisfying-assignment outputs are strictly larger than the baseline pairs, illustrating that inclusion-minimality is not minimum size.

5.4 A deadline computable before enumeration

Let $\mathcal{G}_0 = \{S(P_i) : 1 \leq i \leq n\}$. By (16) this is a polynomially constructible family of n distinct irreducible RAFs, and

$$\mathcal{I}(Q_\varphi) = \mathcal{G}_0 \iff |\mathcal{I}(Q_\varphi)| = n \iff \varphi \text{ is unsatisfiable.} \tag{17}$$

In the unsatisfiable case the required output length is the known polynomial

$$B_0 = (r_\varphi + 1)\, n + 1.$$

This is the encoded length of the complete output, not merely a bound on the number of outputs; the formal development bounds it by $20\,|\varphi|^3$ in terms of the encoded length $|\varphi|$ of the formula.

Theorem 5.7 (Lower bound)**.** *If a pair (E, p) satisfies the enumeration contract of Section 3, then* $\textsc{Sat} \in \mathsf{P}$*, and hence* $\mathsf{P} = \mathsf{NP}$*.*

Proof. Given φ (padded to $n \geq 2$), construct Q_φ, encode it, compute $T_\varphi = p(N_\varphi + B_0)$, and simulate E for at most T_φ transitions. Declare φ unsatisfiable if and only if E has halted by the deadline with output count exactly n.

If φ is unsatisfiable, then by (17) the actual output length is B_0, so the contract forces E to halt within T_φ steps with count n. If φ is satisfiable, then $|\mathcal{I}(Q_\varphi)| > n$, so a halted output has count larger than n, and a run that has not halted by the deadline has already excluded the unsatisfiable case. Both outcomes are classified correctly.

The encoding of Q_φ and the number T_φ have size polynomial in $|\varphi|$, because the dimensions (15) and B_0 are polynomial and p is fixed. Since E is a fixed machine, bounded simulation with a clock and inspection of the halted unary count cost polynomial overhead. This is a deterministic polynomial-time decider for SAT; NP-completeness of SAT [5, 19, 24] gives $\mathsf{P} = \mathsf{NP}$. □

The argument does not wait for a satisfiable instance's possibly enormous output and makes no assertion about when such runs finish. It requires no way of discovering an unknown enumerator's polynomial; it proves the existential implication that any fixed witnesses (E, p) yield a fixed polynomial SAT decider. The hard small-output instances are the *unsatisfiable*

formulas, so the theorem should not be summarized as a consequence of exponentially many outputs. In the formal development the source compiler, the bounded simulator, the clock, the syntax guard and the count test are all machine constructions with proved bounds; none is an assumed oracle.

6 The conditional upper bound

Assume $\mathsf{P} = \mathsf{NP}$ and fix, once and for all, a deterministic polynomial-time SAT decider. Lemma 4.2 then gives a polynomial-time procedure for $\mathrm{Avail}_Q(\mathcal{G}, U)$, uniformly in Q, $\mathcal{G}$ and U.

Procedure. Start with $\mathcal{G} = \varnothing$ and repeat: query $\mathrm{Avail}_Q(\mathcal{G}, R)$; if false, stop. Otherwise set $U = R$ and scan the reactions once in their fixed input order; for each a, replace U by $U \setminus \{a\}$ if $\mathrm{Avail}_Q(\mathcal{G}, U \setminus \{a\})$ holds. After the scan append U to $\mathcal{G}$. On termination write the unary count $|\mathcal{G}|$ and the stored masks in the format (4).

Lemma 6.1 (Filtered deletion)**.** *If a round starts with $\mathcal{G} \subseteq \mathcal{I}(Q)$ and $\mathrm{Avail}_Q(\mathcal{G}, R)$ true, its deletion scan ends at a set $U \in \mathcal{I}(Q) \setminus \mathcal{G}$.*

Proof. $\mathrm{Avail}_Q(\mathcal{G}, U)$ remains true after every accepted deletion. If deleting a is rejected, the container without a has no avoiding witness; later containers only shrink, so by monotonicity the deletion of a never becomes feasible. Hence the final U is a minimal container satisfying $\mathrm{Avail}_Q(\mathcal{G}, \cdot)$: any proper subset omits some retained a and lies inside the failed container $U \setminus \{a\}$.

Let $S \subseteq U$ be an avoiding RAF witnessing the final query. S is itself a feasible container, so $S = U$ by minimality; thus U is a RAF. If a proper RAF subset $V \subsetneq U$ existed it would still avoid all of $\mathcal{G}$ (an $I \subseteq V \subseteq U$ would contradict avoidance of U), so V would be a smaller feasible container. Hence $U \in \mathcal{I}(Q)$. Finally $U \notin \mathcal{G}$, since otherwise avoidance would require $U \not\subseteq U$. □

Theorem 6.2 (Conditional upper bound)**.** *If $\mathsf{P} = \mathsf{NP}$, there exist a fixed deterministic multitape machine E and a polynomial p satisfying the enumeration contract of Section 3.*

Proof. By Lemma 6.1 and induction, the stored list is duplicate-free, consists of irreducible RAFs, and grows by one in every successful round. There are exactly $K = |\mathcal{I}(Q)|$ successful rounds followed by one negative query, at which point Lemma 4.1 certifies completeness. The case $K = 0$ never enters a minimization round.

For the time bound set $W = N + B$. At every stage $g = |\mathcal{G}| \le K$, the stored masks and exclusion clauses have length polynomial in $N + gr \le W$, and Lemma 4.2 bounds each query's encoding by a polynomial in W; composing with the fixed SAT decider gives a fixed polynomial per query. Each successful round makes at most $r + 1$ queries and the final round one, so the number of queries is at most $(r + 1)K + 1 \le W + 1$. Constructing the queries, scanning masks, updating the stored family, and writing the final count and masks add polynomial overhead. Hence the total time is bounded by a polynomial in W that depends only on the chosen SAT decider. □

The filter must stay in force *during* minimization. Minimizing an arbitrary RAF from the full reaction set could return a known output and stall progress; in the procedure above every accepted container retains a witness that avoids all known outputs. Theorems 5.7 and 6.2 together prove Theorem 1.1.

7 Consequences

The results of this section are conventional deductions from the exact correspondence (16) and the verified finite semantics; their complexity-class statements have not separately been compiled in Lean (Section 8 states precisely what has).

7.1 Certification and counting

Let $\textsc{Exact}(Q, \mathcal{G})$ mean that a supplied explicit family $\mathcal{G}$ of reaction masks consists of exactly all irreducible RAFs of Q (repeated masks are deduplicated first), and let $\textsc{Extra}(Q, \mathcal{G})$ mean that $\mathcal{G} \subseteq \mathcal{I}(Q)$ and $\mathcal{I}(Q) \setminus \mathcal{G} \neq \varnothing$. Input size includes Q and all supplied masks.

Proposition 7.1 (Certification)**.** Exact *is* coNP*-complete, and* Extra *is* NP*-complete, under polynomial many-one reductions. Hardness holds already for systems with one food molecule and one catalyst per reaction.*

Proof. Validity of each supplied mask is polynomial by Lemma 2.2. If all members are valid, an avoiding RAF is a polynomial-size certificate that $\mathcal{G}$ is incomplete (Lemma 4.1), so $\textsc{Extra} \in \mathsf{NP}$ and $\textsc{Exact} \in \mathsf{coNP}$ (accept the complement when validity fails or an avoiding RAF exists). For hardness, map φ to $(Q_\varphi, \mathcal{G}_0)$: by (17) the family is complete exactly when φ is unsatisfiable, and $\mathcal{G}_0$ is always valid, so the reduction lands in the valid part for Extra as well. □

Consequently, unless $\mathsf{NP} = \mathsf{coNP}$, complete families do not in general admit polynomial-size certificates checkable in polynomial time; this does not exclude short certificates on particular instances, and the deletion criterion of [32] is exactly such a certificate when the product of the list sizes is small. The parameterized refinement with $|\mathcal{G}|$ as parameter is co-W[P]-complete [3]; that is a separate result and is not a consequence of Proposition 7.1.

Proposition 7.2 (Counting)**.** *The function $Q \mapsto |\mathcal{I}(Q)|$ is in $\#\mathsf{P}$ and is $\#\mathsf{P}$-hard under polynomial-time Turing reductions; hence it is $\#\mathsf{P}$-complete in that sense.*

Proof. Membership: guess a width-r mask and accept exactly when Lemma 2.2 holds; each output corresponds to one accepted mask, not to its support certificates. Hardness: by (16), $\#\textsc{Sat}(\varphi) = |\mathcal{I}(Q_\varphi)| - n$ for φ with $n \geq 2$ declared variables; if padding added u variables, divide by 2^u. Counting satisfying assignments is $\#\mathsf{P}$-complete [37]. □

The reduction is not parsimonious: it deliberately adds n baseline outputs. Note also that output-polynomial enumeration, even under $\mathsf{P} = \mathsf{NP}$, is not a polynomial-time counter, because its running time may still be exponential in the input through the output length.

7.2 Polynomial delay

The contract of Section 3 writes the count first. Under the usual convention in which outputs are streamed as records, the classification persists.

Proposition 7.3 (Streaming enumeration)**.** *Irreducible RAFs admit a polynomial-delay, polynomial-space enumeration algorithm if and only if $\mathsf{P} = \mathsf{NP}$.*

Proof. Assume $\mathsf{P} = \mathsf{NP}$. For a partial mask $\sigma \in \{0,1\}^{\leq r}$ (a decision on the first $|\sigma|$ reactions) the language "some $I \in \mathcal{I}(Q)$ extends σ" is in NP, since a full mask can be guessed and checked by Lemma 2.2; fix a polynomial-time decider for it. Traverse the binary prefix tree depth first, entering only prefixes with a valid extension. The depth is r, so between consecutive output leaves, and before the first or after the last, at most $O(r)$ extension tests are made; the traversal stores only the current path. This is the standard extension-oracle route to bounded delay [35, Section 4]. Conversely, a polynomial-delay enumerator has total time polynomial in N plus the number of records; buffering its masks and prefixing the unary count yields a machine satisfying the contract of Section 3, and Theorem 5.7 gives $\mathsf{P} = \mathsf{NP}$. □

7.3 Deletion survival and essential reactions

Proposition 7.4 (Survival under deletion)**.** *For every $D \subseteq R$,*

$$\mathrm{maxRAF}_Q(R \setminus D) \neq \varnothing \iff \exists I \in \mathcal{I}(Q) : \; I \cap D = \varnothing.$$

Hence the inclusion-minimal RAF-destroying deletion sets are exactly the minimal transversals of $\mathcal{I}(Q)$.

Proof. A surviving RAF contains an irreducible one, which is disjoint from D; a surviving irreducible RAF is itself a witness. □

A partial family thus gives a one-sided certificate: an unhit listed member proves survival, while hitting every listed member proves extinction only if the list is complete or a residual maxRAF test confirms it.

Proposition 7.5 (Universally required reactions)**.** *If Q contains a RAF, then for every $a \in R$,*

$$a \in \bigcap_{I \in \mathcal{I}(Q)} I \iff \mathrm{maxRAF}_Q(R \setminus \{a\}) = \varnothing.$$

The intersection of all irreducible RAFs is therefore computable with r maxRAF calls and no enumeration.

Proof. Apply Proposition 7.4 with $D = \{a\}$. □

Such indispensable reactions already appear in the lower-bound method of [32, Section 5]; Proposition 7.5 is their exact family interpretation. When no RAF exists the family is empty and the vacuous intersection should be reported as "no RAF", not as essentiality. The complete family does not reconstruct every surviving reaction: with $a\colon f \to x$ and $b\colon x \to y$ both catalysed by food f, the unique irreducible RAF is $\{a\}$ while the maxRAF is $\{a, b\}$, so $\bigcup \mathcal{I}(Q)$ need not equal $\mathrm{maxRAF}_Q(R)$, and the family determines survival but not the residual maxRAF or its size.

7.4 Structural indicators that cannot detect hardness

Proposition 7.6 (Overlap and low-order deletion response)**.** *Let φ have $n \geq 2$ variables and let Q_φ be as in Construction 5.2, with $a = |\mathcal{A}| = q + 1$.*

(a) *Any two members of $\mathcal{I}(Q_\varphi)$ have Jaccard similarity at least $a/(a+2n)$. Adding tautological clauses to φ leaves $\#\textsc{Sat}(\varphi)$ unchanged while increasing a, so for fixed n this bound tends to 1 along a family of instances whose completion problem remains NP-hard.*

(b) *Let D consist of input reactions only. If $|D| < n$ then $\mathrm{maxRAF}_{Q_\varphi}(R_\varphi \setminus D) = R_\varphi \setminus D$, independently of φ. If D deletes exactly one input of every conflict pair, leaving the choice set of an assignment α, then $\mathrm{maxRAF}_{Q_\varphi}(R_\varphi \setminus D)$ is $R_\varphi \setminus D$ if $\alpha \models \varphi$ and $\varnothing$ otherwise.*

Proof. (a) Members share $\mathcal{A}$ and differ in at most $2n$ input reactions. (b) If fewer than n inputs are deleted, some conflict pair survives intact, the remaining choice set is accepted, and by Lemma 5.5 the whole remaining canonical set is a RAF. If exactly one input of every pair is deleted, the surviving choice set T_α is conflict-free, so it and its proper subsets are accepted only if $\alpha \models \varphi$; the maxRAF is then the whole remaining set or empty by (16). □

Thus the SAT distinction can first appear at cooperative deletion order n while all lower-order input-deletion responses are identical, and pairwise output overlap can be arbitrarily close to one. These are limitations of proposed structural indicators, not claims about typical chemical networks.

7.5 Positive and negative frontiers

By Proposition 7.4 the negative frontier of the monotone viability predicate is the minimal transversal family of $\mathcal{I}(Q)$, and it need not be compact. For k independent two-reaction catalytic cycles, each consuming food and producing its partner's catalyst, the irreducible RAFs are the k pairs, and a minimal destructive deletion chooses one reaction from each pair: there are 2^k of them. An enumeration algorithm charged only for its k positive outputs therefore cannot be required to materialize the negative frontier. This is also why Theorem 1.1 says nothing about the explicit transversal-hypergraph problem [8, 10, 26]: a catalytic reaction system supplies a compact polynomial-time viability oracle, not a small explicit hypergraph whose transversals are its irreducible RAFs, and the distinction between learning both boundaries and recovering only the positive one is the one drawn in [2].

8 Formal verification

8.1 The exported theorem

The classification is realized in Lean 4 (toolchain `v4.30.0`, with a pinned `mathlib` revision) by the declaration

```
IrrRAFEnumeration.outputPolynomialEnumeration_iff_P_eq_NP :
  EnumerationContract.OutputPolynomialEnumeration ↔ Complexity.P =
Complexity.NP
```

in `proofs/IrrRAFEnumeration/Classification.lean`, assembled from the two directions:

```
theorem outputPolynomialEnumeration_of_P_eq_NP
    (h : Complexity.P = Complexity.NP) : OutputPolynomialEnumeration := by
  exact outputPolynomialEnumeration_of_initializer h
    (fun k => <24, originalInitializerTM k, originalInitializerPolynomial,
      originalInitializer_correct k>)
    (fun _ E p hE => enumerates_of_finEnumerates E p hE)

theorem outputPolynomialEnumeration_iff_P_eq_NP :
    EnumerationContract.OutputPolynomialEnumeration <->
      Complexity.P = Complexity.NP :=
  <P_eq_NP_of_outputPolynomialEnumeration, outputPolynomialEnumeration_of_P_eq_NP>
```

(angle brackets stand for Lean's anonymous-constructor brackets). The theorem has no outer parameters or hypotheses. Its recorded axiom dependencies are `propext`, `Classical.choice` and `Quot.sound` only; there is no `sorry` and no additional mathematical axiom.

The left-hand side is the literal contract of Section 3:

```
def Enumerates {k : Nat} (E : TM k) (p : Polynomial Nat) : Prop :=
  forall (M R : Type) [DecidableEq M] [DecidableEq R] [Fintype R]
    (a b : Nat) (em : M ~= Fin a) (er : R ~= Fin b)
    (Q : CRS M R) (C : Catalysis M R) [DecidableRel C],
    exists (rows : List (Finset R)) (c : Cfg k E.Q) (t : Nat),
      rows.Nodup /\ rows.toFinset = irrRAFFamily Q C /\
      t <= p.eval ((inputBits Q C em er).length
                  + (outputBits er rows).length) /\
      E.reachesIn t (E.initCfg (inputBits Q C em er)) c /\ E.halted c /\
      c.output.HasOutput (outputBits er rows)

def OutputPolynomialEnumeration : Prop :=
  exists (k : Nat) (E : TM k) (p : Polynomial Nat), Enumerates E p
```

Here `inputBits` is (2), `outputBits` is (4) with the proved length identity $(r+1)K+1$, `irrRAFFamily Q C` is the finite set of inclusion-minimal nonempty RAFs, and the quantification ranges over all finite types M, R with explicit bijections to initial segments of $\mathbb{N}$. The right-hand side uses the machine-based classes $\mathsf{P} = \bigcup_k \mathrm{DTIME}(n^k)$ and $\mathsf{NP} = \bigcup_k \mathrm{NTIME}(n^k)$ of `complexitylib` [28], from which the development also imports the Cook–Levin theorem `SAT.NPComplete_language`. That library is vendored at an immutable commit with its Apache-2.0 notices; its declarations are unchanged and are compiled locally.

8.2 Concordance

Table 2 maps the statements of this paper to the Lean declarations that realize them. All paths are relative to `proofs/IrrRAFEnumeration/`.

8.3 From prepared computation to the original input

The conditional direction first obtains, from $\mathsf{P} = \mathsf{NP}$, one SAT machine and one polynomial *before* quantifying over the system. Its prepared enumeration routine maintains the base chemistry formula, the accumulated exclusion clauses, the current container mask, the minimization state, the stored output masks and the output count; a successful round appends a new mask and its exclusion clause together, and the final writer realizes the count-first format. This routine has one uniform polynomial bound $p_1(N+B)$.

A separate initializer starts from the actual encoded input, reads the dimensions, constructs the chemistry CNF, installs the full reaction mask and counters, and establishes the exact entry configuration of the prepared routine on the physical tapes and head positions. Its proved bound is

$$p_0(N) = 50\,000\,000\,(N+1)^6 + 6N^2 + 21N + 54, \tag{18}$$

a conservative execution bound rather than a suggested implementation cost. Retained compiler scratch occupies a disjoint bank of work tapes that the prepared routine never reads. Sequential composition gives

$$p_0(N) + 1 + p_1(N+B) \le (p_0 + p_1 + 1)(N+B),$$

using nonnegative coefficients and $N \le N+B$, so initialization is charged on the original input and leaves no prepared-formula premise in the root theorem.

The finite-index proof is transported through the supplied molecule and reaction bijections: closure stages, catalyst availability, RAF status and minimality are preserved, and the input bits and output masks are identical under the change of interpretation. The same machine run therefore serves arbitrary finite identifiers with no uncharged conversion, and distinct reactions with identical incidence data remain distinct.

8.4 Trust boundary

Strict verification in the frozen environment accepted the root with warnings treated as errors. The receipt records the toolchain `leanprover/lean4:v4.30.0`, the SHA-256 of the pinned dependency manifest (`a8df0b60...`), of the root source (`3aee5560...`) and of the compiled root artifact (`1bbc72c4...`), and an elapsed time of 66.3 seconds for the final module; a subsequent audit rehashed the root and all 378 recorded local source/artifact dependency pairs. Full hashes and the verification command are in Appendix B.

What is trusted: the Lean 4 kernel and the pinned `mathlib` and `complexitylib` revisions, including the latter's Turing-machine model and its definitions of P, NP and SAT. What is proved: everything in Table 2, for concrete machines and actual tape contents. What is *not* proved in Lean: the complexity-class statements of Section 7 (conventional proofs from verified finite lemmas), and any property of the reference software of Section 9, which is not extracted

Paper statement	Lean declaration(s)
Theorem 1.1	`Classification.lean`: `outputPolynomialEnumeration_iff_P_eq_NP`
Encoding, Definition 3.1	`EnumerationContract.lean`: `inputBits`, `outputBits`, `outputBits_length`, `Enumerates`, `OutputPolynomialEnumeration`
RAF semantics, Definition 2.1	`proofs/RAF/Core/Closure.lean`, `proofs/RAF/Core/RAF.lean`; `RAFCompletion.lean`: `mem_irrRAFFamily`, `hasRAFWithin_iff_contains_irrRAF`
Lemma 4.1	`PositiveCompletion.lean`: `available_univ_iff_missing`, `available_mono`
Lemma 4.2	`PositiveSupportedTrace.lean`: `isRAF_iff_supported`; `PositiveCompletionCNF.lean`: `formula_eval_iff`, `formula_sound`; `PositiveCompletionReduction.lean`: `formula_complete`, `formula_satisfiable_iff`
Lemma 5.1	`SATCompletion.lean`: `minimal_eval_iff` (hypothesis $n \geq 2$)
Lemma 5.3	`CircuitSource.lean`: `raf_contains_all_aux`
Lemma 5.4	`CircuitClosure.lean`: `raf_output_reached_without_reset`, `isRAF_iff_all_aux_and_output`
Lemma 5.5, eq. (16)	`CircuitBijection.lean`: `canonical_irreducible_iff`; `SATSource.lean`: `accepts_encode_iff_eval`; `SATReduction.lean`: `sourceSet_irreducible_iff`, `irreducible_iff_exists_sourceSet`
Eq. (17), length B_0	`SATFamily.lean`: `baseline_card`, `baseline_subset_irrRAFFamily`, `irrRAFFamily_eq_baseline_iff_unsat`; `SATCountTimeout.lean`: `irrRAFFamily_card_eq_iff_cnf_unsat`, `countTimeout_length_polynomial`, `bounded_count_decides_cnf_unsat`
Theorem 5.7	`SourceEnumeratorRun.lean`: `source_enumerator_no_case`; `SATMembership.lean`: `SAT_mem_P_of_sourceCountBound`; `NegativeClassification.lean`: `P_eq_NP_of_outputPolynomialEnumeration`
Lemma 6.1	`PositiveCompletion.lean`: `sweep_minimal`, `minimal_available`, `sweep_new_minimal`, `raf_sweep_new`
Theorem 6.2, prepared machine	`CompletionUniformEnumerationCost.lean`: `preparedEnumeration_polynomial`; `CompletionConditionalEnumeration.lean`: `preparedEnumeration_of_P_eq_NP`
Original-input initializer	`CompletionOriginalInput.lean`: `originalInitializer_correct`, `initialFinalWork_eq`
Arbitrary identifiers	`CompletionLabelTransport.lean`: `enumerates_of_finEnumerates`
Theorem 6.2, composed	`Classification.lean`: `outputPolynomialEnumeration_of_P_eq_NP`

Table 2: Concordance between the paper and the Lean development.

from the development. Neither a solver experiment nor a numerical audit is a premise of Theorem 1.1.

9 A practical enumerator and a bounded pilot

The formal upper bound needs only a SAT *decider*. A practical solver returns a model, which yields an avoiding RAF at once and can be minimized by ordinary maxRAF calls instead of further SAT queries.

Model-assisted enumeration. Maintain a validated duplicate-free known family $\mathcal{G}$ (initially empty or user-supplied) and the base formula $\Psi(Q, \varnothing, R)$ with one exclusion clause (13) per member of $\mathcal{G}$. Repeat: solve. If unsatisfiable, return $\mathcal{G}$ as complete. If a model is returned, let S be its selected set, verify independently that S is a RAF avoiding $\mathcal{G}$, then scan reactions in fixed order and, for each $a \in T$ (initially $T = S$), replace T by $\mathrm{maxRAF}_Q(T \setminus \{a\})$ whenever that is nonempty. Verify irreducibility of the final T by Lemma 2.2, emit it, add its exclusion clause, and continue. If the solver reports unknown or a resource limit is hit, return $\mathcal{G}$ with status *incomplete*.

Proposition 9.1 (Solver-call count)**.** *Every set emitted by the model-assisted procedure is a new irreducible RAF, and no output is repeated. If $|\mathcal{I}(Q)| = K$ and g distinct valid members were supplied initially, a completed uninterrupted run makes exactly $K-g+1$ solver calls ($K-g$ satisfiable, one final unsatisfiable) and at most $r(K-g)$ maxRAF calls during minimization.*

Proof. Every intermediate T is a RAF contained in the avoiding witness S, so it remains avoiding by heredity. If deleting a leaves no RAF, subsequent shrinkage cannot change that, so one pass suffices and the final T is irreducible and new, as in Lemma 6.1. Adding $\bigvee_{a \in T} \neg s_a$ excludes exactly the supersets of T, which contain no other irreducible RAF by the antichain property, so every missing output stays reachable. Each satisfiable call therefore enlarges $\mathcal{G}$ by one distinct member, and the count follows. □

This is a count of solver invocations, not a running-time guarantee: the calls can be hard, as Theorem 5.7 shows they must sometimes be. Sampling can populate the initial family cheaply, after which one completion query settles whether anything was missed; this combines fast discovery with an exact stopping condition without making the sampler uniform. Compared with the tuple-deletion criterion (6), the SAT formulation expresses the choices and the chemical witness jointly instead of materializing the Cartesian product $I_1 \times \cdots \times I_g$.

A bounded pilot. A small Python reference implementation of this procedure over Z3 [6] was run on toy fixtures to check the semantics and the call-count identity, with every emitted mask re-checked by independent closure and single-deletion maxRAF tests. The fixtures are ligation-only polymer systems over binary strings of length at most $n \in \{2, 3\}$ with food $\{0, 1\}$ and independent catalyst assignments with probability 0.22 (three construction seeds each), and one published example network from Hordijk's RAF repository with each reversible reaction split into two separately selectable directions. Table 3 shows the $n = 3$ instances and the published example. On every completed run from an empty family the model-assisted variant made exactly $K + 1$ solver calls; the decision-only variant, which minimizes by further SAT queries as in Section 6, made many more (for the seven-core instance, 8 versus 148) and all complete methods agreed on the family. On the 392-reaction published example the unoptimized reference code exhausted its four-second budget while still *building* the formula, before any solver call, and reported unknown; sampling found three verified cores per seed and no completeness statement. These are single-run pilot measurements of an unoptimized script, intended to validate semantics and the identity of Proposition 9.1, not a performance study and not a comparison with the optimized implementations of [11, 16].

Instance	d/r	Method	Status	$\lvert\mathcal{G}\rvert$	SAT	maxRAF
polymer n=3, seed 11	14/20	model-assisted	complete	4	5	10
		decision-only	complete	4	85	0
		exhaustive	complete	4	0	0
polymer n=3, seed 23	14/20	model-assisted	complete	2	3	5
		decision-only	complete	2	43	0
		exhaustive	unknown	2	0	0
polymer n=3, seed 47	14/20	model-assisted	complete	7	8	38
		decision-only	complete	7	148	0
		exhaustive	complete	7	0	0
		sampling, 3 seeds	unknown	6, 7, 7	0	≈ 2550
published example, split	62/392	model-assisted	unknown	0	0	0
		sampling, 3 seeds	unknown	3, 3, 3	0	≈ 2980

Table 3: Bounded pilot with a four-second budget per run. $\lvert\mathcal{G}\rvert$ is the number of independently verified outputs; "SAT" and "maxRAF" count solver and maxRAF calls, the latter excluding output validation. "Unknown" means the budget expired without a completeness certificate.

10 Discussion

10.1 What the classification says for applications

RAF analysis has been applied to metabolic reconstructions such as the *E. coli* network [29] and to origin-of-metabolism questions [38]. In that setting Theorem 1.1 separates two tasks that are easily conflated: *discovering* minimal structural cores, which sampling and deletion sweeps do efficiently, and *establishing that none remain*, which in the worst case is a SAT problem. The practical recommendation is to report discovered cores together with their completion status, and to use a completion query (Section 4) when an exhaustive conclusion matters. Several structural questions do not require enumeration at all: whether any RAF survives a deletion (Proposition 7.4, one maxRAF call), and which reactions are required by every irreducible RAF (Proposition 7.5, r calls).

The object is a finite incidence model with explicit food and catalyst data. Structural survival and essentiality are combinatorial properties of that model, not measurements of gene or organism essentiality, kinetic frequency, mass balance, thermodynamic feasibility or sustained growth; flux-balance and hyperflow formalisms encode distinctions that incidence-only RAFs do not [1, 27], and the relationship between RAFs and chemical organizations is its own subject [17]. Modelling conventions change the family: splitting a reversible reaction into two selectable directions changes the choice space rather than merely relabelling it, and the identity policy for duplicate reactions or isoenzymes must be declared. Since $\bigcup\mathcal{I}(Q)$ can omit reactions of the maxRAF, screening only minimal cores through a downstream quantitative test can miss useful larger RAFs; a combined notion of minimality needs its own search and minimality guarantees.

10.2 What the classification does not say

Theorem 1.1 does not decide P versus NP, does not give an unconditional exponential lower bound, does not exclude useful heuristics or efficient restricted classes, and says nothing about average-case or randomized enumeration. The source uses one food molecule and one catalyst per reaction, so hardness holds under those restrictions, but its finish and reset reactions have unbounded arity; bounded-arity hardness is not established by this construction. The tractable models of [31] are a natural place to look for restricted classes with efficient complete enumeration, provided the restriction is stated on the full reaction model rather than inferred from catalytic topology alone, since Proposition 7.6 shows that overlap statistics and low-order

deletion responses cannot detect the interacting choices.

10.3 Open problems

1. **Bounded arity.** Is complete enumeration still hard when every reaction has at most two reactants and one product, as in the Set Cover source of [4]? The reset and finish reactions would have to be replaced by chains whose intermediate products do not create new irreducible RAFs.
2. **Restricted classes.** For which structurally defined classes (elementary systems, bounded catalytic in-degree, bounded treewidth of the incidence hypergraph) does $\mathcal{I}(Q)$ admit polynomial-delay enumeration unconditionally?
3. **Parameterized delay.** The certification problem is co-W[P]-complete in the list length [3]. Is there a natural parameter, such as the maximum size of an irreducible RAF or the number of reactions with several catalysts, under which enumeration becomes fixed-parameter tractable with polynomial delay?
4. **Certificates of completeness.** Proposition 7.1 rules out short certificates in general unless $\mathsf{NP} = \mathsf{coNP}$. Which instance classes admit checkable completeness certificates, for example resolution refutations of $\Psi(Q, \mathcal{I}(Q), R)$ of polynomial size?

A Exact formula sizes and the worked example

For implementation planning, let $A_\rho = \sum_a |\rho(a)|$, $A_\pi = \sum_a |\pi(a)|$, $A_C = |C|$, $h = |F|$, $t = |R \setminus U|$ and $H_\mathcal{G} = \sum_{I \in \mathcal{G}} |I|$, with $g = |\mathcal{G}|$. The displayed construction $\Psi(Q, \mathcal{G}, U)$ has

$$v = r + d(d+1) + dr \ \text{ variables}, \qquad c = 1 + t + (d-h) + dr + dA_\rho + d^2 + A_\rho + r + g \ \text{ clauses},$$

and, before duplicate removal or simplification,

$$\lambda = 2r + t + (d-h) + 2dr + 2(d+1)A_\rho + 2d^2 + dA_\pi + A_C + H_\mathcal{G}$$

literal occurrences. These count the rows of (7)–(13); empty clauses contribute zero literals but count as clauses. For $r = 0$ the nonemptiness clause is empty and the formula is unsatisfiable; for $d = 0$ no catalyst exists and the catalyst clauses exclude all selections; a reaction with no reactants is enabled at once but still needs a catalyst.

For Example 5.6 the dimension arithmetic is $q = 3{\cdot}3{+}2{\cdot}3{\cdot}2{+}1 = 22$, $d_\varphi = 6{\cdot}3{+}12{+}2{+}4 = 36$, $r_\varphi = 15 + 12 + 2 = 29$, $N = 2 \cdot 36 + 29 + 2 + 3 \cdot 36 \cdot 29 = 3235$, $B_0 = 30 \cdot 3 + 1 = 91$, and for two satisfying assignments $B = 30 \cdot 5 + 1 = 151$. A conflict-pair output has $q + 1 + 2 = 25$ reactions; a satisfying-assignment output has $q + 1 + n = 26$.

B Reproduction

The Lean development lives in the repository directory `proofs/IrrRAFEnumeration/` with the RAF core in `proofs/RAF/Core/` and the vendored complexity library in `proofs/Complexitylib/` (upstream commit `43d4bb1a30bd93846eab5480bec8a37113586109`). The root was verified with the repository's strict bridge, which enters the frozen `mathlib4_project` environment and compiles with `-DwarningAsError=true`:

```
python scripts/verify_proof.py proofs/IrrRAFEnumeration/Classification.lean \
  --declaration IrrRAFEnumeration.outputPolynomialEnumeration_iff_P_eq_NP \
  --declaration IrrRAFEnumeration.outputPolynomialEnumeration_of_P_eq_NP \
  --timeout 600 --output classification.verify.json
```

The successful receipt records the following.

Toolchain	`leanprover/lean4:v4.30.0`
`mathlib` revision	`c5ea00351c28e24afc9f0f84379aa41082b1188f`
Dependency manifest SHA-256	`a8df0b606ec258561c226df89856884b` `e433c19b49fb0a27335d69bb2b6e9fac`
Root source SHA-256	`3aee5560a7b19cdd7f2d2ef5e5806989` `fe0758a8b899f780821d4d5aadd0d473`
Root artifact SHA-256	`1bbc72c4bf4b5b44a23ee9aab87014f9` `85f4c007160c796e0ef307b455d2925c`
Exported theorems	`outputPolynomialEnumeration_iff_P_eq_NP` `outputPolynomialEnumeration_of_P_eq_NP`
Outer binders	none
Axioms	`propext`, `Classical.choice`, `Quot.sound`
Elapsed (final module, strict)	66.297 s

Each 64-character hash is printed on two lines. The pilot of Section 9 used Z3 version 4.15.4 through its Python bindings, one process, and a four-second wall budget per run; its exact inputs (including the pinned published example, source commit `2e63a44a`) and per-run event logs are distributed with the reference script. A dependency-free executable specification of the finite semantics, used to check the semantic distinctions of Section 2 on six tiny fixtures, is included as well. None of this software is extracted from the Lean development, and completeness outcomes reported by it trust the solver's unsatisfiability answer and the Python translation.

References

[1] Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. Defining autocatalysis in chemical reaction networks, 2021. arXiv:2107.03086.

[2] Jan C. Bioch and Toshihide Ibaraki. Complexity of identification and dualization of positive Boolean functions. *Information and Computation*, 123(1):50–63, 1995. doi:10.1006/inco.1995.1157.

[3] Anonymous manuscript. Certifying complete families of irreducible autocatalytic sets is co-W[P]-complete. Manuscript, 7 September 2026, 2026.

[4] Anonymous manuscript. Minimum autocatalytic networks cannot be approximated within any constant factor. Manuscript, 7 September 2026, 2026.

[5] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC 1971)*, pages 151–158. ACM, 1971. doi:10.1145/800157.805047.

[6] Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2008)*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008. doi:10.1007/978-3-540-78800-3_24.

[7] 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.

[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] Yannick Forster, Fabian Kunze, and Maximilian Wuttke. Verified programming of Turing machines in Coq. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020)*, pages 114–128. ACM, 2020. doi:10.1145/3372885.3373816.

[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] Wim Hordijk. Algorithms for enumerating irreducible and closed autocatalytic sets. Preprints.org, version 1, posted 24 April 2026; not peer reviewed, 2026.

[12] 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.

[13] 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.

[14] Wim Hordijk and Mike Steel. Autocatalytic networks at the basis of life's origin and organization. *Life*, 8(4):62, 2018. doi:10.3390/life8040062.

[15] 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.

[16] 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.

[17] Wim Hordijk, Mike Steel, and Peter Dittrich. Autocatalytic sets and chemical organizations: modeling self-sustaining reaction networks at the origin of life. *New Journal of Physics*, 20:015011, 2018. doi:10.1088/1367-2630/aa9fcd.

[18] David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988. doi:10.1016/0020-0190(88)90065-8.

[19] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

[20] Stuart A. Kauffman. Autocatalytic sets of proteins. *Journal of Theoretical Biology*, 119 (1):1–24, 1986. doi:10.1016/S0022-5193(86)80047-9.

[21] Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York, 1993.

[22] Dimitris J. Kavvadias, Martha Sideri, and Elias C. Stavropoulos. Generating all maximal models of a Boolean expression. *Information Processing Letters*, 74(3–4):157–162, 2000. doi:10.1016/S0020-0190(00)00023-5.

[23] Eugene L. Lawler, Jan Karel Lenstra, and Alexander H. G. Rinnooy Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9(3):558–565, 1980. doi:10.1137/0209042.

[24] Leonid A. Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9 (3):115–116, 1973.

[25] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2):223–250, 2016. doi:10.1007/s10601-015-9183-0.

[26] Arnaud Mary. Enumeration of minimal transversals of hypergraphs of bounded VC-dimension, 2024. arXiv:2407.00694, version 4 (2026).

[27] Jeffrey D. Orth, Ines Thiele, and Bernhard Ø. Palsson. What is flux balance analysis? *Nature Biotechnology*, 28:245–248, 2010. doi:10.1038/nbt.1614.

[28] Samuel Schlesinger. complexitylib: computational complexity theory in Lean 4. Source repository, https://github.com/SamuelSchlesinger/complexitylib, commit `43d4bb1a`; Apache-2.0 license, 2025.

[29] Filipa L. Sousa, Wim Hordijk, Mike Steel, and William F. Martin. Autocatalytic sets in *E. coli* metabolism. *Journal of Systems Chemistry*, 6:4, 2015. doi:10.1186/s13322-015-0009-7.

[30] Mike Steel. Interior operators and their relationship to autocatalytic networks. *Acta Biotheoretica*, 71(4):21, 2023. doi:10.1007/s10441-023-09472-8.

[31] Mike Steel and Wim Hordijk. Tractable models of self-sustaining autocatalytic networks, 2018. arXiv:1801.03953.

[32] 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.

[33] 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.

[34] 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.

[35] Yann Strozecki. Enumeration complexity: incremental time, delay and space, 2023. Habilitation manuscript, arXiv:2309.17042.

[36] 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.

[37] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979. doi:10.1137/0208032.

[38] 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.