

Complete enumeration of irreducible autocatalytic sets with bounded supplier choice

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. A companion classification shows that no uniform output-polynomial algorithm lists the complete family of irrRAFs unless $P = NP$, so unconditional tractability must come from structure in the input. We identify such structure in the multiplicity of *suppliers*. After polynomial maximal-RAF pruning, let d_x be the number of reactions producing a nonfood species x and let a_r be the number of effective catalyst alternatives of a reaction r (one, when a food catalyst is available). The supplier excess $\beta = \sum_x (d_x - 1) + \sum_r (a_r - 1)$ is computed from the input and bounds the number $L = \prod_x d_x \prod_r a_r \leq 2^\beta$ of deterministic supplier resolutions. We prove that every irrRAF survives at least one resolution, chosen by a globally shared earliest-producer rule; that the irrRAFs of a pruned resolution are exactly the sink strongly connected components of its consumer-to-supplier dependency graph; and that validating each such component in the original system, with one maximal-RAF call per reaction, yields precisely the complete original family. The algorithm runs in time and space $2^\beta N^{O(1)}$ for explicit input length N ; it places no bound on the length of food-generation chains, so arbitrarily large deterministic systems have $\beta = 0$. Because the components of one resolution are disjoint, every reaction lies in at most L irrRAFs and the total output length is at most mL for a maximal RAF of m reactions; a coupled gate family shows that both bounds are attained up to a constant factor. Deleting reactions filters the family exactly and cannot increase β ; incidence-separated components let the parameter be replaced by its largest local value, giving fixed-parameter tractability in $\beta_{\max}$; and destructive reaction sets are exactly the transversals of the catalogue, which yields a $(1 + \beta \ln 2)$ -approximate greedy intervention and exact cut-cost, cut-count and reliability formulas at zero excess. The catalogue theorem, its candidate bounds, the overlap bounds and the restriction identity are machine-checked in Lean 4; the cost, component and intervention results have conventional proofs. Bounded exact tests and an instrumented implementation illustrate the mechanisms.

Keywords. autocatalytic sets; RAF theory; irreducible RAFs; enumeration algorithms; fixed-parameter tractability; hitting sets; set cover; formal verification.

MSC 2020. 92E20, 68Q27, 05C85, 68W40, 68V20.

1 Introduction

Reflexively autocatalytic and food-generated sets (RAFs) are a finite combinatorial model of collectively self-sustaining reaction networks. The concept originates in Kauffman’s autocatalytic sets of proteins [21], was given its present form by Hordijk and Steel [16], and has grown into a structural theory with efficient algorithms and applications to early metabolism and the origin of life [17, 18, 24, 27, 31]. 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 reaction set 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 food molecule or by a molecule the set produces.

Unions of RAFs are RAFs, so a system containing a RAF contains a largest one, the *maximal RAF*, computable in polynomial time by the RAF algorithm of Hordijk and Steel [16]. The minimal organizations point the other way. Steel, Hordijk and Smith [26] studied *irreducible* RAFs (irrRAF), which contain no proper RAF, and showed that one irrRAF can be found in polynomial time by a deletion sweep while a RAF of minimum size is NP-hard to find. They also gave a criterion certifying that a supplied list of k irrRAFs is complete, at a cost growing as the product of the listed sizes, and asked whether this dependence can be improved [26, Section 6.1]. Hordijk’s recent preprint develops exhaustive and randomized enumeration procedures [15]. A companion manuscript settles the unrestricted question: a uniform deterministic algorithm that writes every irrRAF exactly once in time polynomial in the input and output lengths exists if and only if $P = NP$ [3]; exact certification of a supplied family is coNP-complete, and its parameterized version is co-W[P]-complete [2]. Under standard assumptions the complete family is therefore out of reach for arbitrary inputs, and the practical question becomes which computable features of a reaction system make it reachable.

This paper answers that question for one natural feature: the multiplicity of *suppliers*. In a system where every nonfood species has exactly one producer and every reaction has exactly one relevant catalyst, the requirements of a reaction determine the reactions it depends on, and inclusion-minimal RAFs become a graph problem. When some species have several producers or some reactions several catalysts, each alternative is a binary choice that the complete family may or may not exploit. We measure this ambiguity by the *supplier excess* β (Definition 2.3) and show that the complete irrRAF family can be listed exactly in time $2^\beta N^{O(1)}$. The parameter is measured on the input after one polynomial pruning, never on the unknown output, and it places no restriction on how long a chain of food-generation steps may be: a linear chain of a thousand reactions, each producing the reactant of the next, has $\beta = 0$.

1.1 Main results

Throughout, Q is a finite catalytic reaction system with reaction identifier set R , $\mathcal{I}(Q)$ is its family of irrRAFs, K is its maximal RAF, $m = |K|$, $n = |R|$, and N is the length of an explicit encoding of Q (Section 2). The supplier excess β and the resolution count $L \leq 2^\beta$ are defined in Section 2.3.

- **Complete enumeration (Theorem 4.2, Proposition 4.3).** A supplier catalogue algorithm returns exactly $\mathcal{I}(Q)$, each member once, in total time and space $2^\beta N^{O(1)}$. It inspects at most mL candidate sets in total, and validates candidates with at most m original maximal-RAF calls per resolution.
- **Overlap bounds (Theorem 4.2).** $\mathcal{I}(Q)$ can be partitioned into at most L classes of pairwise disjoint sets. Hence every reaction lies in at most L irrRAFs, and $|\mathcal{I}(Q)| \leq \sum_{I \in \mathcal{I}(Q)} |I| \leq mL \leq m2^\beta$.
- **Sharpness (Proposition 5.1).** A family of systems with $3k$ reactions and $\beta = k$ has exactly 2^k irrRAFs, each of size $2k$, and each of its k gate reactions lies in all of them. Any algorithm that writes every irrRAF explicitly needs time exponential in β on this family.
- **Restriction (Theorem 6.1).** Deleting reactions filters the family exactly, $\mathcal{I}(Q|U) = \{I \in \mathcal{I}(Q) : I \subseteq U\}$, and cannot increase β . Counting, minimum-size and minimum-nonnegative-cost RAFs, required/forbidden queries and deletion survival are then direct scans of the catalogue.

- **Components (Theorem 7.2).** If the nonfood incidence graph of the pruned system splits into blocks, the family is the disjoint union of the block families, β adds and L multiplies, but the algorithm pays the *sum* of the block resolution counts. Enumeration is fixed-parameter tractable in the largest local excess $\beta_{\max}$.
- **Interventions (Corollary 8.1, Theorem 8.2).** A reaction set destroys every RAF exactly when it meets every irrRAF. Weighted greedy deletion is within a factor $H_d \leq 1 + \ln L \leq 1 + \beta \ln 2$ of the cheapest destructive set. At $\beta = 0$ the irrRAFs are pairwise disjoint and the optimal cut cost, the number of inclusion-minimal cuts and the probability that a RAF survives independent reaction failures have closed forms.

The enumeration theorem, its candidate and resolution bounds, both overlap bounds and the restriction identity are checked in Lean 4 [8, 30] with no hypotheses beyond the standard axioms (Section 9). The complexity account, the monotonicity of β under restriction, the component theorem and the intervention results are conventional proofs from those verified statements and are labelled as such.

1.2 Proof idea

A *resolution* chooses one producer for each nonfood species and one effective catalyst for each reaction, deletes every other product occurrence, and replaces every catalyst list by its chosen member. Three facts make resolutions useful.

Soundness. Thinning products and catalysts can only shrink food closures, so every RAF of a resolved system is a RAF of the original.

Preservation by earliest producers. Fix an original RAF I . For each nonfood molecule in the closure of I , choose a producer in I whose reactants are all present at the stage *before* the molecule first appears, and use that one producer at every later stage. This single global choice per molecule keeps every closure stage of I exactly equal to the original one, so I remains a RAF, and by soundness an irreducible I remains irreducible. An arbitrary producer choice does not suffice: choosing the self-loop $x \rightarrow x$ as the only producer of x destroys the ability to start from food (Example 3.5).

Sink components. In a resolution pruned to its maximal RAF, every required nonfood molecule has a unique producer. Drawing an arc from each reaction to the producer of each of its nonfood reactants and of its chosen catalyst, the RAFs are exactly the nonempty sets closed under outgoing arcs, and the irrRAFs are exactly the sink strongly connected components. Such components are disjoint and there are at most m of them.

Every original irrRAF therefore appears as a sink component of some resolution. The converse fails: a sink component may be an original RAF that is no longer minimal once the deleted alternatives are restored (Figure 1). A single original maximal-RAF call per reaction of the candidate decides this, and the disjointness of components charges those calls to disjoint reactions. The same disjointness gives the overlap bounds and, through a hitting-set reduction, the intervention guarantees.

1.3 Related work

Structural tractability in RAF theory. Steel and Hordijk [25] study elementary systems, in which every reactant is food, and characterize their irrRAFs through chordless cycles in a catalysis graph. Elementary systems remove nonfood generation; supplier bounds keep generation chains of any length and instead control alternative witnesses. Neither “one catalyst per reaction” nor “one producer per molecule” alone gives the present theorem: the companion SAT source of [3] has one catalyst per reaction and carries all of its choices through alternative producers, while its set-cover source carries choices through catalysis, so both excess terms are needed.

Enumeration complexity. The output-sensitive standard for listing problems goes back to Johnson, Yannakakis and Papadimitriou [20]; see Strozecki [28] for a survey of incremental time, delay and space. The present algorithm is a total-time bound with a parameter-dependent exponential factor, not an output-polynomial or polynomial-delay guarantee, and Section 4.4 says precisely what is and is not claimed about delay.

Parameterized complexity. Fixed-parameter tractability with respect to a parameter measured on the input is the framework of Downey and Fellows [10] and Cygan et al. [7]. Courcelle’s theorem [5] and its enumeration form [6] give a general logical baseline for bounded-width inputs; Section 11.3 records a monadic second-order translation of original irreducibility and shows that supplier excess and incidence treewidth are incomparable parameters.

Hitting sets and set cover. Destructive reaction sets are transversals of the catalogue hypergraph. The greedy set-cover guarantee is due to Johnson, Lovász and Chvátal [1, 19, 22] and is essentially optimal in general [9, 12]; our contribution is the exact reduction and the structural bound on set sizes. Listing all minimal destructive cuts is the transversal hypergraph problem [11], for which Fredman and Khachiyan give an incremental quasi-polynomial algorithm [14].

1.4 Evidence levels and organization

Three kinds of evidence appear and are kept apart. Universal finite statements checked in Lean are named with their declarations in Section 9. Conventional proofs are complete mathematical arguments in this paper that have not been ported to Lean. Computations are bounded exact tests and instrumented runs of a Python implementation that is not extracted from Lean; they illustrate and discriminate, but do not prove.

Section 2 fixes the model, maximal-RAF pruning, effective catalysts and the parameter. Section 3 develops resolutions, earliest producers, dependency graphs and sink components. Section 4 states the algorithm, the enumeration and overlap theorem and the cost bound. Section 5 gives the sharpness family and zero-excess families of unbounded size, depth and width. Sections 6–8 treat restriction and catalogue queries, independent components and interventions. Section 9 describes the formal development, Section 10 the computational checks, and Section 11 the scope of the result and open problems.

2 Catalytic reaction systems and irreducible RAFs

2.1 Closure and the RAF condition

A *catalytic reaction system* $Q = (M, R, F, \text{in}, \text{out}, \text{Cat})$ consists of a finite set M of molecule types, a finite set R of reaction identifiers, a food set $F \subseteq M$, and for each $r \in R$ finite sets $\text{in}(r), \text{out}(r), \text{Cat}(r) \subseteq M$ of reactants, products and catalysts. Distinct identifiers remain distinct even if their incidence sets agree, and empty product sets are permitted; both conventions are used by the intermediate constructions below.

For $S \subseteq R$ the *food closure* is computed in synchronous stages,

$$X_0(S) = F, \quad X_{t+1}(S) = X_t(S) \cup \bigcup_{\substack{r \in S \\ \text{in}(r) \subseteq X_t(S)}} \text{out}(r), \quad \text{cl}_Q(S) = \bigcup_{t \geq 0} X_t(S). \quad (1)$$

The stages stabilize because M is finite. Catalysts do not gate this iteration. A nonempty $S \subseteq R$ is a *RAF* if for every $r \in S$

$$\text{in}(r) \subseteq \text{cl}_Q(S) \quad \text{and} \quad \text{Cat}(r) \cap \text{cl}_Q(S) \neq \emptyset. \quad (2)$$

An *irreducible RAF* (irrRAF) is a RAF with no proper RAF subset; $\mathcal{I}(Q)$ denotes the family of irrRAFs. If $S \subseteq T$ then $X_t(S) \subseteq X_t(T)$ for every t , by induction in (1), so closures are monotone.

Remark 2.1 (Semantic conventions). Four conventions are essential and are those of the Lean development. (i) Catalysts do not gate the closure: with food f , the single reaction $f \rightarrow x$ catalysed by x is a RAF, although its first firing has no catalyst present; requiring catalytic startup is a stronger predicate. (ii) Catalysis is disjunctive: one available member of $\text{Cat}(r)$ suffices. (iii) Irreducible means inclusion-minimal, not minimum cardinality; irrRAFs of one system may have different sizes. (iv) Reactants and products record occurrence only. Stoichiometry, concentrations, rates, reversibility and conservation laws are outside the predicate. A set closed under the dependency arcs of Section 3 is not automatically a “closed RAF” in the separate RAF-theoretic sense of containing every reaction it enables.

2.2 Maximal RAFs and effective catalysts

For a container $U \subseteq R$ define $\mu_Q(U)$ by repeatedly removing from the current set every reaction that violates (2) with respect to the closure of the current set. At most $|U|$ strict removals occur. Every RAF $I \subseteq U$ survives every round, since its own closure is contained in the current closure; the fixed point is empty or a RAF, and it contains every RAF inside U . Thus $\mu_Q(U)$ is the unique maximal RAF in U when one exists, and μ_Q is monotone in U . This is the RAF algorithm of [16] restricted to a container.

Set $K = \mu_Q(R)$, $m = |K|$, $n = |R|$ and $M_K = F \cup \bigcup_{r \in K} \text{out}(r)$. Every RAF is contained in K , and $\text{cl}_Q(K) = M_K$: the inclusion $\subseteq$ is the closure definition, and every reaction of the RAF K eventually fires, so all of its products appear. If $K = \emptyset$ both sides equal F .

For $x \in M_K \setminus F$ let $P_x = \{r \in K : x \in \text{out}(r)\}$ be its producers in K and $d_x = |P_x| \geq 1$. For $r \in K$ fix

$$A_r = \begin{cases} \{\text{a fixed canonical member of } \text{Cat}(r) \cap F\}, & \text{Cat}(r) \cap F \neq \emptyset, \\ \text{Cat}(r) \cap M_K, & \text{Cat}(r) \cap F = \emptyset, \end{cases} \quad a_r = |A_r|. \quad (3)$$

The list A_r is nonempty: a reaction of the RAF K has a catalyst in $\text{cl}_Q(K) = M_K$. The canonical choice uses a fixed order on species and adds no chemical condition.

Lemma 2.2 (Effective catalysts). *For every $S \subseteq K$, S is a RAF of Q if and only if S is a RAF of the system obtained by replacing $\text{Cat}(r)$ with A_r for all $r \in K$.*

Proof. Closures do not depend on catalysts. Food is in every closure, so a reaction with a food catalyst satisfies the catalyst condition in both systems. If r has no food catalyst, $\text{cl}_Q(S) \subseteq \text{cl}_Q(K) = M_K$, so any catalyst of r in $\text{cl}_Q(S)$ lies in $\text{Cat}(r) \cap M_K = A_r$. Every member of A_r is an original catalyst. $\square$

The lemma applies to every selected set inside K , not only to K itself; that uniformity is what allows candidates to be tested for original minimality later.

2.3 The supplier parameter

Definition 2.3 (Supplier excess and resolution count).

$$\beta(Q) = \sum_{x \in M_K \setminus F} (d_x - 1) + \sum_{r \in K} (a_r - 1), \quad L(Q) = \prod_{x \in M_K \setminus F} d_x \prod_{r \in K} a_r. \quad (4)$$

Empty products are 1, so $K = \emptyset$ gives $\beta = 0$, $L = 1$ and $\mathcal{I}(Q) = \emptyset$.

For every positive integer b one has $b \leq 2^{b-1}$, so multiplying over all coordinates gives

$$L \leq 2^\beta. \quad (5)$$

The exact count L is often much smaller than 2^β : one species with b producers contributes $b - 1$ to β but only a factor b to L . Both quantities are computed from the input after one polynomial pruning; their bit lengths are polynomial in N .

Example 2.4 (Deep systems with zero excess). Let $r_1: f \rightarrow x_1$ and $r_i: x_{i-1} \rightarrow x_i$ for $2 \leq i \leq \ell$, all catalysed by x_ℓ . Every nonfood species has one producer and every reaction one catalyst, so $\beta = 0$. Any RAF must generate x_ℓ , which forces the whole chain, so the unique irrRAF has ℓ reactions and its closure needs ℓ stages. Supplier excess bounds neither the size of an output nor the depth of its generation.

Remark 2.5 (Encoding). Let N be the bit length of an indexed explicit encoding of M , R , F and the three incidence lists. Let $T_\mu(N)$ be a polynomial bound on one maximal-RAF computation, enlarged to cover the construction of resolutions, graph scans and strongly connected components. Such a polynomial exists directly from the finite closure and pruning loops; strongly connected components take linear time in the graph size [23, 29].

3 Supplier resolutions

3.1 Resolutions and soundness

Definition 3.1 (Resolution). A *resolution* $\sigma = (p, c)$ chooses a producer $p(x) \in P_x$ for every $x \in M_K \setminus F$ and a catalyst $c(r) \in A_r$ for every $r \in K$. The *resolved system* Q_σ has reaction set K , the same food and reactant sets, product sets

$$\text{out}_\sigma(r) = (\text{out}(r) \cap F) \cup \{x \in \text{out}(r) \setminus F : p(x) = r\},$$

and catalyst sets $\text{Cat}_\sigma(r) = \{c(r)\}$. There are exactly L resolutions.

A resolution is a combinatorial construction, not a chemical intervention: it selects which witness each requirement will use.

Lemma 3.2 (Soundness). *For every resolution σ and every $S \subseteq K$, $X_t^{Q_\sigma}(S) \subseteq X_t^Q(S)$ for all t . Consequently every RAF of Q_σ is a RAF of Q .*

Proof. Induction on t in (1): resolved product sets are subsets of the original ones, and a reaction enabled at a resolved stage is enabled at the corresponding original stage by the induction hypothesis. Every resolved catalyst $c(r)$ is an original catalyst, so both conditions of (2) transfer. $\square$

3.2 Earliest producers preserve every RAF

Lemma 3.3 (Earliest producers). *Let $S \subseteq K$ and let $x \notin F$ lie in $\text{cl}_Q(S)$. There is a reaction $p \in S$ with $x \in \text{out}(p)$ such that for every t , if $x \in X_{t+1}(S)$ then $\text{in}(p) \subseteq X_t(S)$.*

Proof. Let $u + 1$ be the first stage containing x ; $u + 1 \geq 1$ since $x \notin F$. Some $p \in S$ with $\text{in}(p) \subseteq X_u(S)$ produces x at that stage. If $x \in X_{t+1}(S)$ then $t \geq u$, and stages are monotone in t , so $\text{in}(p) \subseteq X_u(S) \subseteq X_t(S)$. $\square$

The point is that the same p serves at *every* later stage and for every consumer of x ; a collection of unrelated witnesses, one per requirement, would not give a single producer function.

Lemma 3.4 (Preserving resolution). *For every original RAF I there is a resolution σ such that $X_t^{Q_\sigma}(I) = X_t^Q(I)$ for all t and I is a RAF of Q_σ . If I is an original irrRAF, then I is an irrRAF of Q_σ .*

Proof. For each $x \in \text{cl}_Q(I) \setminus F$ let $p(x) \in I$ be a producer as in Lemma 3.3; note $p(x) \in P_x$ because $I \subseteq K$. For every other $x \in M_K \setminus F$ choose any member of P_x . We claim $X_t^{Q_\sigma}(I) = X_t^Q(I)$ for every t and every catalyst choice c . The inclusion $\subseteq$ is Lemma 3.2. For $\supseteq$ induct on t ; stage 0 is F on both sides. Let $x \in X_{t+1}(I)$. If $x \in F$ it is present. Otherwise Lemma 3.3 gives $\text{in}(p(x)) \subseteq X_t(I) = X_t^{Q_\sigma}(I)$, so $p(x)$ fires at resolved stage t , and it retains the product x because $p(x)$ is the chosen producer. Hence $x \in X_{t+1}^{Q_\sigma}(I)$.

For $r \in I$ choose $c(r)$ as the canonical food catalyst if $\text{Cat}(r) \cap F \neq \emptyset$, and otherwise any catalyst in $\text{Cat}(r) \cap \text{cl}_Q(I)$; this set is nonempty because I is a RAF and is contained in $\text{Cat}(r) \cap M_K = A_r$. For $r \in K \setminus I$ choose any member of A_r . Every reactant and every chosen catalyst of a reaction in I lies in the preserved closure, so I is a resolved RAF. If I is originally irreducible and $J \subsetneq I$ were a resolved RAF, then J would be an original RAF by Lemma 3.2, a contradiction. $\square$

No single resolution is claimed to preserve all RAFs at once. Lemma 3.4 is an existence statement for each RAF separately, and the earliest-stage condition does real work.

Example 3.5 (Arbitrary producer choice can remove food startup). Let $F = \{f\}$, $r_0: f \rightarrow x$ and $r_1: x \rightarrow x$, both catalysed by x . Originally $\{r_0\}$ is a RAF. Choosing r_1 as the sole producer of x deletes x from the products of r_0 ; the closure of $\{r_0, r_1\}$ in the resolved system is $\{f\}$, and no resolved RAF exists. The earliest producer of x in the closure of $\{r_0\}$ is r_0 , and that choice preserves it.

3.3 Deterministic dependency graphs and sink components

Fix a resolution σ and let $K_\sigma = \mu_{Q_\sigma}(K)$ be the maximal RAF of the resolved system. If a reaction $r \in K_\sigma$ has a nonfood reactant x , or a nonfood chosen catalyst $c(r) = x$, then $x \in \text{cl}_{Q_\sigma}(K_\sigma)$ and, since thinning left one producer, $p(x)$ is the unique reaction of K_σ producing x . The *dependency graph* G_σ has vertex set K_σ and an arc $r \rightarrow p(x)$ for every nonfood reactant x of r and for its nonfood chosen catalyst. Arcs point from consumer to supplier; a reaction may have several outgoing arcs, and “deterministic suppliers” does not mean outdegree at most one. A set $S \subseteq K_\sigma$ is *outgoing-closed* if every arc leaving a vertex of S ends in S .

Lemma 3.6 (Deterministic dependency characterization). *A nonempty $S \subseteq K_\sigma$ is a RAF of Q_σ if and only if it is outgoing-closed in G_σ .*

Proof. If S is a RAF and $r \in S$ requires the nonfood molecule x as reactant or chosen catalyst, then $x \in \text{cl}_{Q_\sigma}(S)$, so some reaction of S produces x ; by uniqueness that reaction is $p(x)$, the head of the arc. Hence S is outgoing-closed.

Conversely let S be outgoing-closed and use the closure stages $X_t(K_\sigma)$ of the RAF K_σ . We show by strong induction on t : whenever $r \in S$ has $\text{in}(r) \subseteq X_t(K_\sigma)$, then $\text{in}(r) \subseteq \text{cl}_{Q_\sigma}(S)$ and $\text{out}_\sigma(r) \subseteq \text{cl}_{Q_\sigma}(S)$. For $t = 0$ the reactants are food. For the step, a nonfood reactant x of r first appeared at a stage $u + 1 \leq t$ through its unique producer $p(x)$, whose reactants lie in $X_u(K_\sigma)$ with $u < t$. The arc $r \rightarrow p(x)$ keeps $p(x) \in S$, so by induction $x \in \text{cl}_{Q_\sigma}(S)$. Thus r fires in the closure of S and its products appear. Since K_σ is a RAF, every $r \in S$ has its reactants at some stage, so the reactant condition holds for all of S . The chosen catalyst of r is food or the product of $p(c(r)) \in S$, which fires by the same argument; so the catalyst condition holds. No catalyst was assumed to precede its consumer in generation time. $\square$

Pruning to K_σ is essential for the converse: a dependency cycle can be closed under outgoing arcs without any way to start from food, and the generation stages of K_σ rule this out.

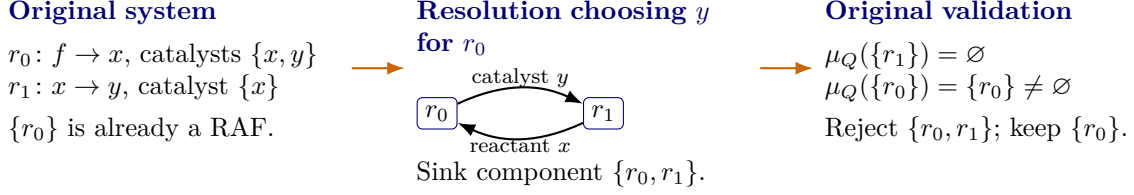


Figure 1: Original validation is necessary. With food f , choosing the nonfood catalyst y for r_0 creates a two-vertex dependency cycle whose sink component is a resolved irrRAF. Restoring the original alternative x reveals the proper RAF $\{r_0\}$, so the pair is rejected by the one-deletion test (6).

Lemma 3.7 (Minimal sets are sink components). *The irrRAFs of Q_σ are exactly the sink strongly connected components of G_σ , isolated vertices included as singleton components. They are pairwise disjoint and there are at most $|K_\sigma| \leq m$ of them.*

Proof. In a finite digraph the set reachable from a vertex is nonempty and outgoing-closed. If S is a minimal nonempty outgoing-closed set and $v \in S$, the set reachable from v is contained in S and is itself nonempty and outgoing-closed, hence equals S ; so every vertex of S reaches every other, and no arc leaves S : S is a sink strongly connected component. Conversely a nonempty outgoing-closed subset of a sink component contains everything reachable from one of its vertices, that is, the entire component. Apply Lemma 3.6. Distinct strongly connected components are disjoint. $\square$

3.4 Original validation is necessary

Lemmas 3.4 and 3.7 show that every original irrRAF is a sink component of some resolution. The converse fails.

Example 3.8 (Resolved minima need not be original minima). (a) Figure 1: $r_0: f \rightarrow x$ with catalysts $\{x, y\}$ and $r_1: x \rightarrow y$ with catalyst $\{x\}$. Choosing y for r_0 makes $\{r_0, r_1\}$ a sink component, but $\{r_0\}$ is an original RAF. (b) Producer choices cause the same effect: $r_0: f \rightarrow \{x, z\}$ catalysed by x and $r_1: f \rightarrow x$ catalysed by z . Choosing r_1 as the unique producer of x deletes x from $\text{out}(r_0)$; the resolved system has the single irrRAF $\{r_0, r_1\}$, whereas $\{r_0\}$ is already an original RAF.

The original exact pilot of Section 10 rejected 111 such candidate occurrences among 2,065; they are expected behaviour of a necessary filter, not exceptions to the theorem.

4 The enumeration theorem

4.1 Testing original irreducibility

For a set C already known to be an original RAF,

$$C \in \mathcal{I}(Q) \iff \mu_Q(C \setminus \{r\}) = \emptyset \text{ for every } r \in C. \quad (6)$$

Indeed a proper RAF subset omits some r and survives the pruning of $C \setminus \{r\}$; conversely a nonempty $\mu_Q(C \setminus \{r\})$ is a proper RAF subset. This is the one-deletion criterion of [26], applied inside the container C . It asks whether any RAF *remains* after a deletion, not whether the immediate deletion is itself a RAF.

Example 4.1 (Immediate deletion is the wrong test). Let $a: f \rightarrow x$ have the food catalyst f , let $b: x \rightarrow y$ be catalysed by z , and let $c: x \rightarrow z$ be catalysed by y . The set $\{a, b, c\}$ is a

RAF with closure $\{f, x, y, z\}$. None of $\{a, b\}$, $\{a, c\}$, $\{b, c\}$ is a RAF, yet $\{a\}$ is, so $\{a, b, c\}$ is reducible although every single deletion leaves a non-RAF. Criterion (6) detects this because $\mu_Q(\{a, b\}) = \{a\} \neq \emptyset$.

4.2 The supplier catalogue algorithm

Supplier catalogue algorithm

1. Compute $K = \mu_Q(R)$ and the lists P_x ($x \in M_K \setminus F$) and A_r ($r \in K$). If $K = \emptyset$, return the empty catalogue.
2. For each resolution $\sigma \in \prod_x P_x \times \prod_r A_r$: construct Q_σ , compute $K_\sigma = \mu_{Q_\sigma}(K)$, build G_σ and find its sink strongly connected components.
3. For each sink component C : test (6) in the *original* system Q . Insert the original-identifier mask of every accepted C into a binary trie [13].
4. After all resolutions, return the distinct masks stored in the trie.

4.3 Completeness and overlap

For $r \in R$ let $f_r = |\{I \in \mathcal{I}(Q) : r \in I\}|$ be the *frequency* of r .

Theorem 4.2 (Complete enumeration and overlap). *The supplier catalogue is exactly $\mathcal{I}(Q)$, each member returned once. At most mL candidate occurrences are inspected in total. The family $\mathcal{I}(Q)$ can be partitioned into at most L classes whose members are pairwise disjoint. Consequently*

$$f_r \leq L \quad (r \in R), \quad |\mathcal{I}(Q)| \leq \sum_{I \in \mathcal{I}(Q)} |I| = \sum_{r \in K} f_r \leq mL \leq m2^\beta. \quad (7)$$

Proof. Every accepted set is a RAF of some Q_σ , hence an original RAF by Lemma 3.2, and passes (6), hence lies in $\mathcal{I}(Q)$. Conversely, an original irrRAF I is an irrRAF of a preserving resolution σ by Lemma 3.4. It lies in K_σ because K_σ contains every RAF of Q_σ inside K , it is a sink component by Lemma 3.7, and it passes (6). The trie removes duplicates without removing a distinct member.

Within one resolution the sink components are nonempty and pairwise disjoint, so there are at most $|K_\sigma| \leq m$ of them; summing over the L resolutions bounds the candidate occurrences. Fix an order of the resolutions and assign each $I \in \mathcal{I}(Q)$ to the first resolution in which it is a sink component. Sets assigned to one resolution are disjoint, which gives the partition. A reaction occurs in at most one set of each class, so $f_r \leq L$; reactions outside K have frequency 0. Double counting the pairs (r, I) with $r \in I$ gives the identity in (7), the first inequality uses that every irrRAF is nonempty, and the last is (5). $\square$

The assignment in the proof is a proper L -colouring of the intersection graph of the catalogue (vertices $\mathcal{I}(Q)$, edges between intersecting members). It is not a degree bound for that graph: one large member may meet many disjoint members of another class. If every output has at least $s \geq 1$ reactions, (7) gives $|\mathcal{I}(Q)| \leq \lfloor mL/s \rfloor$. Frequencies count distinct minimal organizations; they are not probabilities of a reaction being active, and sampling resolutions uniformly does not sample irrRAFs uniformly, since different irrRAFs survive different numbers of resolutions.

4.4 Cost

Within one resolution, criterion (6) makes at most

$$\sum_{C \text{ a sink component of } G_\sigma} |C| \leq |K_\sigma| \leq m$$

original maximal-RAF calls, because the components are disjoint. This charges validation to disjoint reactions across the whole resolution instead of paying m separately for each candidate.

Proposition 4.3 (Conventional cost bound). *With original n -bit masks, the supplier catalogue algorithm runs in time*

$$O(T_\mu(N) + L((m+1)T_\mu(N) + mn)) \quad (8)$$

and uses $N^{O(1)} + O(mnL)$ bits of storage, up to polynomial identifier overhead. In particular both are $2^\beta N^{O(1)}$, with an exponent of the polynomial factor independent of β .

Proof. Preprocessing costs $O(T_\mu(N))$. Each resolution costs one resolved pruning, at most m original validation prunings, and graph work absorbed into T_μ . At most m candidate occurrences are accepted per resolution, each requiring an n -step trie traversal; a trie creates at most one node per traversed bit. Only one resolved system need be kept at a time. Counters and indices have polynomial bit length since $\log_2 L \leq \beta \leq N^{O(1)}$. Pessimistic implementations of finite-set operations add only an input-polynomial factor. Substitute (5). $\square$

Remark 4.4 (What is and is not claimed about delay). The implementation buffers its catalogue: the time to the first output includes the entire construction, and later masks are emitted in $O(n)$ each. This gives parameter-dependent total time and parameter-dependent delay after preprocessing. It does not establish polynomial delay for unrestricted β , which by [3] would imply $P = NP$; nor polynomial space independent of β . Relabelling K would replace n by m in (8), but the reported implementation keeps original identifiers, including those removed by the initial pruning. Sparse output lists have total length at most mL by (7), although the simple trie uses more space.

Remark 4.5 (The parameter is sufficient, not necessary). Two reactions with distinct food reactants, the same q nonfood products and a food catalyst have exactly two irrRAFs, the two singletons, but every shared product has two producers, so $\beta = q$ and $L = 2^q$. Unused shared outputs create supplier coordinates that no RAF requirement exploits. Theorem 4.2 remains correct and its bound is loose. A preprocessing step that deletes irrelevant products would need its own preservation proof and is not assumed by the algorithm.

5 Sharpness and zero-excess families

5.1 A family in which every supplier choice is essential

For $k \geq 1$ take food f , nonfood species $x_1, \dots, x_k, z_1, \dots, z_k$ and $3k$ reactions: two producers $a_i, b_i: f \rightarrow x_i$ for each i , and the gates

$$g_1: x_1 \rightarrow z_1, \quad g_i: z_{i-1} + x_i \rightarrow z_i \quad (2 \leq i \leq k),$$

where $+$ denotes a joint reactant requirement. Every reaction has the sole catalyst z_k .

Proposition 5.1 (Exponential output and frequency). *This system has $m = 3k$, $\beta = k$ and $L = 2^k$. Its irrRAFs are exactly the 2^k sets consisting of all k gates and one member of each pair $\{a_i, b_i\}$. Every gate has frequency L , every output has size $2k$, and $\sum_{I \in \mathcal{I}(Q)} |I| = 2k2^k = \frac{2}{3}mL$.*

Proof. The whole system is a RAF, so $K = R$. Any RAF must generate its only catalyst z_k , whose unique producer is g_k ; backwards along the gate chain, all gates and at least one producer of each x_i are required. Conversely such a selection generates every x_i at stage 1, then $z_1, \dots, z_k$ in order, so it is a RAF, and minimality removes every surplus pair member. Only the k species x_i have two producers; all catalysts and all gate products are unique. The counts follow. $\square$

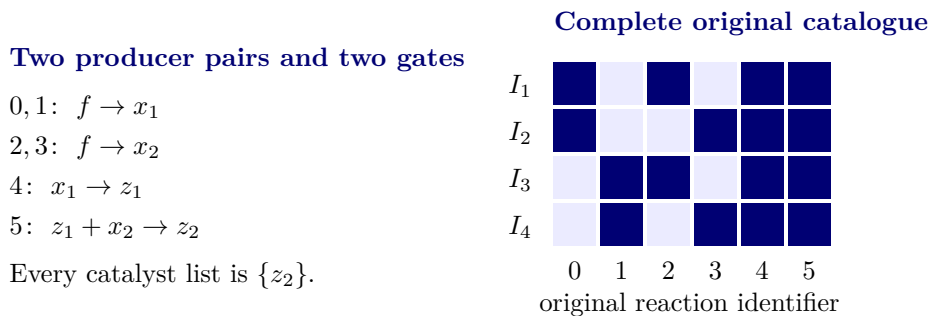


Figure 2: The $k = 2$ gate construction with reaction identifiers $0, \dots, 5$. The four rows on the right are the complete original catalogue; dark cells are included reactions. Both gates occur in every output and each producer alternative occurs in two.

The two producers of x_i may instead consume distinct food substrates $f_{i,0}, f_{i,1} \in F$; the proof and all parameters are unchanged, so the phenomenon does not depend on chemically identical reactions. The family shows that an exponential dependence on β is unavoidable for any algorithm required to write every irrRAF explicitly, that the frequency bound $f_r \leq L$ of (7) is attained, and that the total-membership bound mL is attained within the factor $2/3$. It is not a lower bound for counting or for compressed representations. Figure 2 shows the case $k = 2$, which is also the worked example of Section 6.2.

5.2 Zero-excess families of unbounded size, depth and width

The following systems all have $\beta = 0$ and therefore a single resolution, in which the irrRAFs are simply the sink components of one dependency graph.

Example 5.2 (Attaining the factor m). m reactions $f \rightarrow x_i$, each catalysed by food, form m disjoint singleton irrRAFs, so $|\mathcal{I}(Q)| = mL = m$ and (7) is exact. This system also has $2^m - 1$ RAFs in total: a small complete *minimal* family does not imply a small family of all RAFs.

Example 5.3 (Unbounded depth). The chain of Example 2.4 has one irrRAF of size ℓ and generation depth ℓ .

Example 5.4 (Unbounded incidence treewidth). Replace the reactants of r_i in Example 2.4 by $\{f, x_1, \dots, x_{i-1}\}$, keeping the sole catalyst x_ℓ . Producers and catalysts are still unique, so $\beta = 0$, and the unique irrRAF is still the whole chain. But the reactions $r_{\lceil \ell/2 \rceil + 1}, \dots, r_\ell$ and the species $x_1, \dots, x_{\lceil \ell/2 \rceil}$ span a complete bipartite subgraph of the species–reaction incidence graph, whose treewidth therefore grows linearly in ℓ .

Conversely, the gate family of Proposition 5.1 has bounded width: the bags $B_i = \{f, z_k, x_i, z_i, g_i, a_i, b_i, z_{i-1}\}$ (with z_0 omitted) cover every reactant, product and catalyst incidence, consecutive bags share z_i , and f, z_k lie in all bags, so the incidence graph has pathwidth at most 7 while $\beta = k$ and the output is exponential. Supplier excess and incidence width are incomparable parameters (Section 11.3).

6 Reaction restriction and catalogue queries

For $U \subseteq R$ let $Q|U$ be the system with reaction set U , the same food, and the original incidence sets of the retained reactions. Retained identifiers are identified with their originals.

Theorem 6.1 (Restriction and parameter monotonicity).

$$\mathcal{I}(Q|U) = \{I \in \mathcal{I}(Q) : I \subseteq U\}, \quad \beta(Q|U) \leq \beta(Q). \quad (9)$$

Proof. For $S \subseteq U$ the closure stages of S in $Q|U$ and in Q agree, so RAF membership agrees. If $I \subseteq U$ has a proper original RAF subset, that subset lies in U ; conversely every proper restricted RAF subset is an original one. This is the family identity.

Write $K_U = \mu_Q(U) \subseteq K$, by monotonicity of μ_Q . The nonfood species produced by K_U are among those produced by K , and for each of them the producer list inside K_U is a nonempty subset of the list inside K ; so every surviving term $d_x - 1$ does not increase and vanished species contribute 0. For $r \in K_U$, whether r has a food catalyst is unchanged, giving contribution 0 in both systems; otherwise the effective list changes from $\text{Cat}(r) \cap M_K$ to its subset $\text{Cat}(r) \cap M_{K_U}$, nonempty because r is supported in K_U . Removed reactions contribute 0. Sum. $\square$

Unchanged food and unchanged catalysis are hypotheses. Deleting a catalysis edge rather than a reaction can increase the excess: a single reaction $f \rightarrow \{x, y\}$ with catalysts $\{f, x, y\}$ has $\beta = 0$, and removing only the food catalysis leaves a RAF with two effective nonfood catalysts and $\beta = 1$. Equation (9) filters *original* irrRAFs; it is not a search for sets minimal under an additional required-reaction constraint.

6.1 Queries answered by the catalogue

Once the catalogue is built, the following are direct scans, each polynomial in the catalogue size.

- *Counting and minimum size.* $|\mathcal{I}(Q)|$ and $\min_I |I|$ are read off; every finite RAF contains an irrRAF, so the latter is the minimum RAF cardinality [26].
- *Minimum nonnegative cost.* For nonnegative additive reaction costs, a minimum-cost nonempty RAF may be taken irreducible, since replacing a RAF by an irreducible subset cannot increase cost. With signed costs this fails: for $a: f \rightarrow x$ and $b: x \rightarrow y$, both catalysed by x , the unique irrRAF is $\{a\}$ while the maximal RAF $\{a, b\}$ is cheaper when $c_b < 0$. The same example shows $\bigcup \mathcal{I}(Q) \subsetneq K$ in general.
- *Required and forbidden reactions.* For disjoint $A, B \subseteq R$, the irrRAFs with $A \subseteq I$ and $I \cap B = \emptyset$ are the records passing the two filters.
- *Deletion survival.* By (9) and finite minimal-subset existence, a RAF survives the deletion of $D \subseteq R$ if and only if some record is disjoint from D , and the surviving family is exactly the filtered catalogue. Survival alone is also answered by one maximal-RAF call; the catalogue adds the full list of alternatives.
- *Universally necessary reactions.* If $\mathcal{I}(Q) \neq \emptyset$, a reaction lies in every irrRAF exactly when deleting it destroys every RAF; this needs at most n maximal-RAF calls and no enumeration [3]. For an empty family the vacuous intersection must be reported as “no RAF”, not as necessity.

6.2 A worked example

Take the $k = 2$ system of Figure 2: $F = \{f\}$, reactions $0, 1: f \rightarrow x_1$, $2, 3: f \rightarrow x_2$, $4: x_1 \rightarrow z_1$, $5: z_1 + x_2 \rightarrow z_2$, all catalysed by z_2 . Here $K = R$, $m = n = 6$, $\beta = 2$, $L = 4$, and

$$\mathcal{I}(Q) = \{\{0, 2, 4, 5\}, \{0, 3, 4, 5\}, \{1, 2, 4, 5\}, \{1, 3, 4, 5\}\},$$

with frequencies $(2, 2, 2, 2, 4, 4)$ summing to 16. Both gates are universally necessary. Requiring 0 and forbidding 2 selects $\{0, 3, 4, 5\}$ only; deleting 0 leaves two alternatives. The inclusion-minimal destructive cuts are $\{4\}$, $\{5\}$, $\{0, 1\}$ and $\{2, 3\}$: a gate is hit, or both members of a producer pair are; a single alternative producer is never destructive.

With deletion costs $(c_0, \dots, c_5) = (2, 3, 4, 1, 7, 6)$ the four minimal cuts cost 7, 6, 5, 5, so the optimum is 5. Weighted greedy (Corollary 8.1) first takes reaction 3 at ratio $1/2$, then,

breaking a tie between 0 and 2 at ratio 2 by the smaller identifier, takes 0, then 1, for total cost 6: a correct approximation that is not exact. The same numbers as construction costs give minimum RAF cost 16, attained by $\{0, 3, 4, 5\}$; cutting and building are different objectives. If each reaction is available independently with probability p_r , a RAF survives with probability $p_4 p_5 [1 - (1 - p_0)(1 - p_1)] [1 - (1 - p_2)(1 - p_3)]$, which is $9/64$ at $p_r = 1/2$. This is a direct calculation for an overlapping family; the closed form of Theorem 8.2 does not apply because $\beta = 2$.

7 Independent nonfood components

Replace the catalyst lists on K by the effective lists A_r (Lemma 2.2). Form the undirected bipartite *nonfood incidence graph* with vertex set $K \cup (M_K \setminus F)$, joining $r \in K$ to every nonfood species in $\text{in}(r) \cup \text{out}(r) \cup A_r$. Let $B_1, \dots, B_q$ be the reaction sets of its connected components, isolated reaction vertices included, and let X_j be the nonfood species incident with B_j . The X_j are pairwise disjoint; food may be shared freely. All three incidence types must be included: omitting product or effective-catalyst incidences would not justify the separation.

Lemma 7.1 (Closure locality). *For $S \subseteq K$ and $S_j = S \cap B_j$, at every stage t ,*

$$X_t(S) \cap X_j = X_t(S_j) \cap X_j. \quad (10)$$

Consequently S is a RAF if and only if $S \neq \emptyset$ and every nonempty S_j is a RAF.

Proof. Stage 0: both intersections are empty since X_j is nonfood. Suppose (10) holds at stage t . A reaction producing a member of X_j belongs to B_j , and all its nonfood reactants lie in X_j ; by the induction hypothesis they are available in $X_t(S)$ exactly when they are available in $X_t(S_j)$, and food reactants are available in both. So the reactions producing X_j fire at stage t in S exactly when they fire in S_j , proving (10) at $t + 1$. The nonfood reactants and effective catalysts of a reaction in B_j lie in X_j , so both RAF conditions for S_j are decided by (10), and Lemma 2.2 returns to the original catalysts. $\square$

Theorem 7.2 (Additive component enumeration). *Every B_j is a RAF. With the local parameters $\beta_j = \beta(Q|B_j)$ and $L_j = L(Q|B_j)$,*

$$\mathcal{I}(Q) = \bigsqcup_{j=1}^q \mathcal{I}(Q|B_j), \quad \beta = \sum_j \beta_j, \quad L = \prod_j L_j. \quad (11)$$

Running the supplier catalogue algorithm on each block and concatenating the outputs is a complete algorithm with total time and space

$$N^{O(1)} \left(1 + \sum_j L_j \right), \quad (12)$$

total output length at most $\sum_j |B_j| L_j$, and frequency at most L_j for every reaction of B_j . In particular enumeration is fixed-parameter tractable in $\beta_{\max} = \max_j \beta_j$ (with $\beta_{\max} = 0$ when $K = \emptyset$).

Proof. Apply Lemma 7.1 to the RAF K : each $B_j = K \cap B_j$ is a RAF. If an irrRAF met two blocks, either nonempty intersection would be a proper RAF subset by the lemma, a contradiction; and a local irrRAF has no proper global RAF subset because all of its subsets lie in its block. With (9) this gives the disjoint union.

Every producer coordinate belongs to one block, since all producers of a species are adjacent to that species; every effective catalyst coordinate belongs to its reaction's block. The maximal

RAF of $Q|B_j$ is B_j itself, and $M_{B_j} \setminus F = X_j$, so the effective lists recomputed inside the block are unchanged: a nonfood catalyst in $\text{Cat}(r) \cap M_K$ is adjacent to r and hence lies in X_j . Products and excesses therefore factor coordinatewise, giving the sum and product in (11).

Build the components in polynomial time and apply Proposition 4.3 to each block with the ambient N . No combination of local outputs is ever formed, and no cross-block duplicate can arise because a nonempty set lies in at most one block. Summing (7) over blocks gives the output and frequency bounds. Finally $\sum_j L_j \leq q2^{\beta_{\max}}$ with $q \leq m \leq N^{O(1)}$. $\square$

Example 7.3 (Sum versus product). Take t modules, module j having two producers $f \rightarrow x_j$ and one gate $x_j \rightarrow z_j$, all catalysed by z_j , with fresh species per module. Then $n = 3t$, $\beta = t$, $L = 2^t$, $\beta_{\max} = 1$, $\sum_j L_j = 2t$, and $|\mathcal{I}(Q)| = 2t$. The global loop combines independent choices unnecessarily; the component loop does not. By contrast the coupled gate family of Proposition 5.1 is a single block with 2^k outputs. Separation is a sufficient incidence criterion computed from the input, not an assumption that the family factorizes; shared food is harmless only because the model makes food unconditionally available, and modules sharing a nonfood reactant, product or effective catalyst are not split.

8 Reaction interventions and the zero-excess boundary

Call $D \subseteq R$ *destructive* if $Q|(R \setminus D)$ contains no RAF. By Theorem 6.1 and finite minimal-subset existence,

$$D \text{ is destructive} \iff D \cap I \neq \emptyset \text{ for every } I \in \mathcal{I}(Q). \quad (13)$$

Destructive sets are the transversals of the hypergraph $\mathcal{I}(Q)$, and the inclusion-minimal destructive cuts are its minimal transversals. Given the catalogue, all minimal cuts can be listed in incremental quasi-polynomial time by the algorithm of Fredman and Khachiyan [14]; Example 8.3 shows that this dual family can be exponentially larger than the catalogue. Equation (13) is not a claim that every reaction set surviving a non-destructive deletion is itself a RAF.

For an eligible reaction r let $\mathcal{S}_r = \{I \in \mathcal{I}(Q) : r \in I\}$. Choosing a destructive set is weighted set cover of the universe $\mathcal{I}(Q)$ by the sets $\mathcal{S}_r$, feasible exactly when their union is the universe. Put $d = \max_r |\mathcal{S}_r|$ over eligible reactions; for a nonempty family and a feasible instance, $1 \leq d \leq L$ by (7).

Corollary 8.1 (Greedy destructive deletion). *For a feasible reaction-deletion instance with positive rational costs, repeatedly deleting a reaction of minimum cost per newly hit irrRAF yields a destructive set of cost at most*

$$H_d \text{ OPT} \leq (1 + \ln L) \text{ OPT} \leq (1 + \beta \ln 2) \text{ OPT}, \quad H_d = \sum_{i=1}^d \frac{1}{i}. \quad (14)$$

Under the component criterion, L may be replaced by $\max_j L_j$ and β by $\beta_{\max}$. Nonnegative costs are handled by first taking every free reaction that hits an uncovered member.

Proof. This is the weighted greedy guarantee of Chvátal [1] applied to (13); we include the charging argument for completeness. Charge each newly covered member the ratio (cost of the chosen reaction)/(number of members it newly covers); the total charge equals the greedy cost. Consider an eligible reaction of cost c whose set has $s \leq d$ members and order those members by the step at which greedy covers them, breaking ties arbitrarily. When the i -th of them is covered, at least $s - i + 1$ of them were still uncovered, so greedy's ratio at that step is at most $c/(s - i + 1)$; the total charge on the set is at most $cH_s \leq cH_d$. Summing over the reactions of an optimal cover, every member is counted at least once and charges are nonnegative, so the greedy cost is at most $H_d \text{ OPT}$. Then $H_d \leq 1 + \ln d$ and $d \leq L \leq 2^\beta$. Free reactions cover their members at zero cost and can be removed from the residual instance before the argument. $\square$

This is the classical approximation theorem with a structural bound on the number of minimal organizations one reaction can hit; it is not an exact optimization algorithm, and the $\ln$ dependence cannot be improved for general set cover [9, 12]. A gene deletion that disables several reactions is a different action whose coverage need not be bounded by L .

Theorem 8.2 (Zero-excess formulas). *Let $\beta = 0$ and write $\mathcal{I}(Q) = \{C_1, \dots, C_h\}$; these sets are pairwise disjoint. For nonnegative reaction costs c_r with every reaction eligible,*

$$\text{OPT}_{\text{cut}} = \sum_{j=1}^h \min_{r \in C_j} c_r. \quad (15)$$

The inclusion-minimal destructive cuts are the sets containing exactly one reaction of each C_j , and there are $\prod_j |C_j|$ of them. If reactions are available independently, reaction r with probability p_r , then

$$\Pr(\text{some RAF remains}) = 1 - \prod_{j=1}^h \left(1 - \prod_{r \in C_j} p_r\right). \quad (16)$$

For $h = 0$ the unique minimal destructive cut is empty, the optimum is 0 and the survival probability is 0.

Proof. $\beta = 0$ gives $L = 1$, so Theorem 4.2 puts the whole family in one class of pairwise disjoint sets. A destructive set meets every C_j , and disjointness allows the cheapest choice in each block independently, proving (15). Two chosen reactions in one block, or a chosen reaction outside all blocks, can be dropped without losing the hitting property, so exactly one per block is necessary and sufficient for inclusion-minimality, and the independent choices give the product count. By (9), C_j survives exactly when all of its reactions are available; these events depend on disjoint sets of independent variables, hence are independent with probabilities $\prod_{r \in C_j} p_r$, and (16) is the complement of the event that none survives. The empty-family cases follow from the empty-product convention. $\square$

Example 8.3 (Few irrRAFs, many cuts). Take h independent pairs $f \rightarrow x_j$ and $x_j \rightarrow z_j$, both catalysed by z_j . Each pair is a two-vertex dependency cycle and an irrRAF, so $\beta = 0$ and $|\mathcal{I}(Q)| = h$, but there are 2^h minimal destructive cuts, and at $p_r = 1/2$ the survival probability is $1 - (3/4)^h$. A short catalogue of minimal organizations does not promise a short catalogue of minimal interventions.

These probabilities concern static structural availability; nothing about kinetics, concentrations or flux persistence is inferred from them.

9 Formal verification

9.1 The development

The Lean 4 development [8, 30] lives in the directory `proofs/RAFStructuredEnumeration/` of the companion repository [4] and builds on an existing formalization of the finite RAF semantics: a structure `RAF.CRS` with reactant and product maps and a food set, a catalysis relation `RAF.Catalysis`, the stage closure `RAF.closureAt` of (1), the predicate `RAF.IsRAF` of (2), an executable pruning `RAFQueryCompilation.evaluate` for μ_Q with its fixed-point and containment theorems, and the irreducibility predicate `IsIrreducibleRAF` (a RAF all of whose RAF subsets equal it). Species and reactions are finite types with decidable equality; a linear order on species selects the canonical food catalyst.

Eleven modules realize Sections 2–6. `SupplierResolution.lean` defines product thinning `thin` and proves stage containment (`thin_closure_subset`), soundness (`thin_raf_sound`), the earliest-producer lemma and the stage equality of Lemma 3.4; the latter two are the declarations `earliest_producer` and `earliestChoice_closure`. `SupplierParameter.lean` defines `producerOptions`, `catalystOptions`, the finite set `resolutions` of all resolutions, `supplierExcess`, and proves `resolutions_card`, nonemptiness of every option list and availability of an effective catalyst inside the closure of any supported set (`catalystOption_available`). `EarliestProducer.lean` assembles `preserving_resolution` and `resolved_raf_sound`. `SinkComponents.lean` is a pure finite-graph module: `DependencyClosed`, reachability, `sinkCandidates`, the bound `sinkCandidates_card` and the characterization `minimal_closed_iff_sink` of Lemma 3.7. `DeterministicSuppliers.lean` defines `UniqueSuppliers`, the `dependencies` map and proves `deterministic_raf_iff` (Lemma 3.6), using an existing ranked-support theorem for the sufficiency direction. `CandidateCoverage.lean` proves that thinning yields unique suppliers, that resolved catalyst lists are subsingletons, `deterministic_irr_iff_sink`, and the coverage statement `original_irr_candidate`. `Cost.lean` proves $b \leq 2^{b-1}$ and `resolutions_excess_bound`, the inequality (5). `Enumeration.lean` defines `originalMax`, `allCandidates` and `supplierCatalogue` and proves `supplierCatalogue_correct`, `allCandidates_card` and `supplierCatalogue_card`. `Resolution.lean` exports the root below. `OverlapBounds.lean` proves that two sink candidates sharing a vertex coincide (`sink_eq_of_common`) and derives `catalogue_frequency_bound` and `catalogue_membership_bound`. `Restriction.lean` proves `catalogue_restriction` and `catalogue_restriction_family`, using the existing subtype restriction `restrictCRS` with its stage-preservation theorems `restrict_closureAt`, `restrict_isRAF`, the lemma `minimal_within_iff` and the filtered family `familyWithin` of the module `IrrRAFEnumeration/RestrictedCompletion.lean`.

9.2 The exported theorems

The root declaration is

```
theorem supplierEnumeration_correct {M R : Type*} [DecidableEq M] [DecidableEq R]
  [LinearOrder M] [Fintype M] [Fintype R] (Q : CRS M R) (cats : R -> Finset M) :
  (forall I, I in supplierCatalogue Q cats
    <-> IsIrreducibleRAF Q (fun x r => x in cats r) I)
  /\ (allCandidates Q cats).card
    <= (originalMax Q cats).card * (resolutions Q cats (originalMax Q cats)).card
  /\ (resolutions Q cats (originalMax Q cats)).card
    <= 2 ^ (supplierExcess Q cats (originalMax Q cats))
  /\ (supplierCatalogue Q cats).card
    <= (originalMax Q cats).card * 2 ^ (supplierExcess Q cats (originalMax Q cats))
```

(ASCII rendering of the Unicode source.) Its inputs are the finite system and its catalyst lists only; there is no enumeration oracle, no coverage certificate and no supplied list of irrRAFs among the hypotheses. The formal option encoding is total: a coordinate outside the effective domain (a food species, a species with no producer in K , a reaction outside K) has the single option `none`, contributing factor 1 and excess 0, and active choices are wrapped injectively by `some`; this reconciles the total-function encoding with (4). The catalogue in the root is defined by filtering `allCandidates` with the semantic predicate `IsIrreducibleRAF`; the equivalence of that predicate with the one-deletion pruning test (6) is the separately compiled theorem `IrrRAFEnumeration.irreducibleRAF_iff_oneDeletion`.

The overlap endpoints state that, for every reaction r , the number of catalogue members containing r is at most the number of resolutions, and that $\sum_I |I|$ over the catalogue is at most $|K|$ times the number of resolutions; these are the L -bounds of (7), not merely their 2^β consequences. The restriction endpoints state that a set lies in the catalogue and inside U

if and only if it is minimal for the predicate “ $S \subseteq U$ and S is a RAF”, and that the filtered catalogue equals `familyWithin`.

9.3 Concordance and receipts

Table 1 maps the statements of this paper to their formal or conventional evidence. Every declaration listed was compiled with the repository’s strict verification bridge, which enters a frozen `mathlib` environment and treats warnings as errors, and each receipt reports empty compiler output and only the axioms `propext`, `Classical.choice` and `Quot.sound`; no admitted proposition is used. Table 2 lists the receipts.

Table 1: Evidence map. A dash means that no Lean formalization is claimed; the conventional proof in the indicated section is part of the mathematical argument.

Paper statement	Lean declaration (namespace RAFStructuredEnumeration)	Computational support
Lemma 2.2	<code>catalystOption_available</code> , <code>catalystOption_sound</code>	—
Definition 2.3, (5)	<code>supplierExcess</code> , <code>resolutions_card</code> , <code>resolutions_excess_bound</code>	122-system suite
Lemma 3.2	<code>thin_closure_subset</code> , <code>thin_raf_sound</code>	original pilot
Lemmas 3.3, 3.4	<code>earliest_producer</code> , <code>earliestChoice_closure</code> , <code>preserving_resolution</code> , <code>resolved_raf_sound</code>	original pilot
Lemma 3.6	<code>deterministic_raf_iff</code>	original pilot
Lemma 3.7	<code>minimal_closed_iff_sink</code> , <code>sinkCandidates_card</code> , <code>deterministic_irr_iff_sink</code>	original pilot
Criterion (6)	<code>IrrRAFEnumeration.irreducibleRAF_</code> <code>iff_oneDeletion</code>	Example 4.1 fixture
Theorem 4.2, catalogue and candidate bounds	<code>supplierEnumeration_correct</code> (root)	828-system comparison
Theorem 4.2, overlap (7)	<code>catalogue_frequency_bound</code> , <code>catalogue_membership_bound</code>	122-system suite
Proposition 4.3	—; Section 4.4	instrumented counters
Proposition 5.1	—; Section 5	$k \leq 4$, both variants
Theorem 6.1, family	<code>catalogue_restriction</code> , <code>catalogue_restriction_family</code>	2,899 containers
Theorem 6.1, β	—; Section 6	2,899 containers
Lemma 7.1, Theorem 7.2	—; Section 7	component comparison
Corollary 8.1, Theorem 8.2	—; Section 8	366 cuts, 340 states

9.4 Trust boundary

The formal statements are universal finite theorems about the semantics of Section 2; they do not mention running time. The cost bound of Proposition 4.3 is a conventional account of the loop that the root theorem specifies, the component theorem and the intervention results are conventional proofs from the verified finite lemmas, and the Python implementation of Section 10 is independently written and tested, not extracted from Lean. None of the

Table 2: Strict verification receipts (toolchain `leanprover/lean4:v4.30.0`, dependency manifest SHA-256 `a8df0b606ec258561c226df89856884be433c19b49fb0a27335d69bb2b6e9fac`). Durations include the compilation of every local dependency not already cached. Source hashes are printed as their first sixteen hexadecimal digits; the full values are in the receipt files.

Module	Exported declarations	Deps	Seconds	Source SHA-256
<code>SupplierResolution.lean</code>	<code>thin_raf_sound,</code> <code>earliestChoice_closure</code>	6	81.406	<code>e65b00f9114798ea</code>
<code>EarliestProducer.lean</code>	<code>preserving_resolution</code>	—	67.546	<code>a48438b64eec444b</code>
<code>Deterministic</code> <code>Suppliers.lean</code>	<code>deterministic_raf_iff</code>	—	73.829	<code>322fdb3d4da5bf53</code>
<code>Resolution.lean</code> (root)	<code>supplierEnumeration_correct</code>	36	133.609	<code>fe36718cca5bf6dc</code>
<code>OverlapBounds.lean</code>	<code>catalogue_frequency_bound,</code> <code>catalogue_membership_bound</code>	37	79.766	<code>f3da277b505912f0</code>
<code>Restriction.lean</code>	<code>catalogue_restriction,</code> <code>catalogue_restriction_family</code>	42	94.062	<code>8dfce241a61f3acd</code>

conventional results should be read as Lean-checked because they appear in the same paper, and the executable checks should not be read as proofs.

10 Computational checks and a bounded mechanism study

10.1 Implementation

The Python implementation iterates supplier choices, prunes each resolved system with the maximal-RAF loop, computes sink components with an iterative Kosaraju–Sharir traversal [23], validates each candidate in the original system with criterion (6), and stores accepted original-identifier masks in a binary trie. Its production loop never enumerates reaction subsets; exhaustive subset search occurs only in an independent small-instance oracle with its own closure routine, used as the reference in the exact tests below. A second implementation adds the component decomposition of Section 7 and operation counters. Sources, seeds and raw records are distributed with the repository [4].

10.2 Exact regression

Two original campaigns, preserved unchanged, checked the enumerator against the exhaustive oracle on 827 systems (700 random proposals, 120 planted non-elementary RAF systems, three boundary fixtures and four exponential-output fixtures) and, in a separate implementation comparison, on 828 systems; the two runs overlap and are not 1,655 independent tests. The pilot examined 2,019 resolutions and 2,065 candidate occurrences and rejected 111 of them as originally non-minimal (Section 3.4); the largest excess encountered was 5 and the largest family had 16 members.

The post-proof regression suite (seed 14092026) uses 80 random systems, 40 planted systems and two degenerate fixtures, 122 systems in all with at most six reactions each and an enforced cap of 512 resolutions. For every system the global and the component enumerators returned exactly the exhaustive family, and the per-resolution counters satisfied the disjoint-size and validation bounds of Section 4.4. The suite also verified the restriction identity and $\beta(Q|U) \leq \beta(Q)$ on all 2,899 allowed containers U of these systems; the greedy guarantee (14) against the exact optimum for 366 cost assignments (unit, nonnegative and positive costs, three per system); the reliability formula of Example 8.3 on all $4 + 16 + 64 + 256 = 340$ availability states for $h = 1, \dots, 4$; the gate construction and its distinct-food variant for $k = 1, \dots, 4$, by exhaustive comparison for $k \leq 3$ and against the explicit family for $k = 4$; and the worked

Table 3: Selected instrumented runs from the 29-run study. $n = m$ in every row. V is the number of original validation prunings, B the peak traced Python allocation (not process memory). Each row is a single run with allocation tracing enabled; times are mechanism illustrations, not benchmarks.

Family	Loop	n	β	Res.	Outputs	V	Seconds	B (KiB)
Disjoint pairs	global	128	0	1	64	128	0.0814	1130.5
Chain	global	128	0	1	1	128	0.0205	218.0
Independent	global	24	8	256	16	4096	0.4241	88.2
Independent	component	24	8	16	16	32	0.0090	56.9
Independent	component	192	64	128	128	256	0.3026	468.4
Coupled gates	global	24	8	256	256	4096	0.2650	994.4

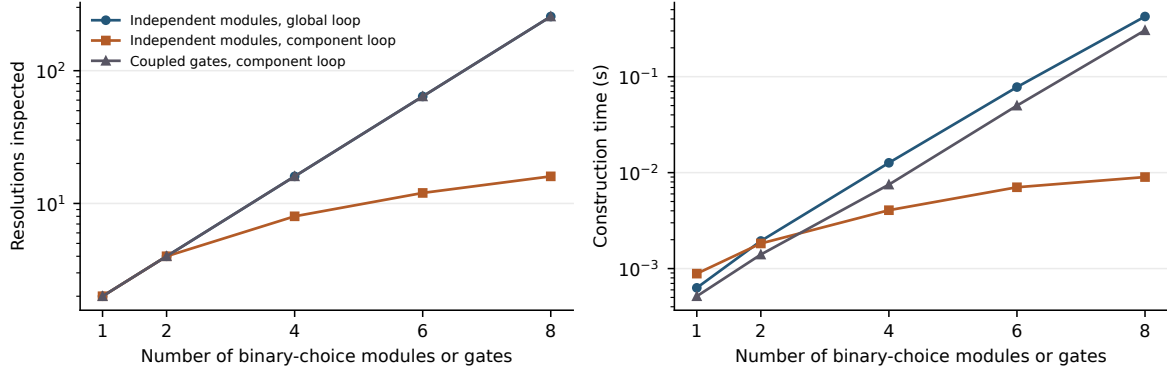


Figure 3: Resolutions inspected (left) and instrumented construction time (right) as the number t of independent binary-choice modules, or the number k of coupled gates, grows. Independent modules cost a sum of local resolution counts in the component loop and a product in the global loop; coupled gates form one component and retain the exponential output requirement. Global runs stop at parameter eight under the declared cap of 256 resolutions.

example of Section 6.2, with its four records and exactly the four minimal cuts. All checks passed. Small exhaustive tests of this kind target mismatched semantics, indexing mistakes and incorrect counters; they do not estimate how common low excess is in real networks.

10.3 Instrumented mechanism study

Three controlled families were run with operation counters and allocation tracing: deterministic systems (disjoint two-reaction cycles, and chains of increasing depth), the independent modules of Example 7.3, and the coupled gates of Proposition 5.1. Global runs were capped at 256 resolutions and larger independent systems were run in component mode only; all 29 calls finished within a 180-second guard. Table 3 and Figure 3 report the results.

The counters, not the timings, carry the argument. At $t = 8$ independent modules the global loop inspects $2^8 = 256$ resolutions and 4,096 validation prunings and generates 2,048 candidate occurrences, of which 2,032 are duplicates of the 16 distinct outputs; the component loop inspects 16 resolutions and 32 prunings for the same 16 outputs. At $t = 64$ the component loop uses 128 resolutions for 128 outputs on 192 reactions, while the global product 2^{64} is computed as a parameter and never enumerated. At $k = 8$ coupled gates both loops use 256 resolutions for 256 distinct outputs, as Proposition 5.1 requires. The deterministic runs show zero-excess systems of 128 reactions handled in one resolution, and the chain family shows generation depth 128 without supplier ambiguity. Raw records also give an incidence-count

proxy and the serialized input length; neither is equated with the abstract bit length N .

11 Discussion

11.1 What the theorem says

Supplier excess is a computable, input-side parameter that controls the complete minimal family while permitting arbitrarily deep food-generation structure. Its exponential dependence is necessary on the coupled family when every output is written explicitly. The overlap theorem explains both the sharper validation account and the bounded coverage that drives the intervention guarantee. Incidence separation refines the parameter to local ambiguity and removes unnecessary combinations. Reaction restriction, counting, minimum-size and minimum-nonnegative-cost RAFs, required/forbidden queries, deletion survival and destructive cuts all reduce to scans of the catalogue. The exponential-output family and the two-cycle family show, respectively, that the catalogue can be large when the parameter is, and that the dual catalogue of minimal cuts can be large even when the parameter and the catalogue are small.

11.2 What the theorem does not say

The proved object is the exact family of original irreducible RAFs under the ordinary finite semantics of Section 2. Structural minimality asserts nothing about kinetic productivity, thermodynamic realizability, concentrations, persistence, gene essentiality or fitness. Filtering the catalogue by a stronger property can miss a larger RAF that is minimal for that stronger property, as the union example of Section 6 shows. The component criterion is sufficient, not necessary: systems sharing nonfood species across modules may be tractable for other reasons, but Theorem 7.2 does not split them. No polynomial delay is claimed without parameter-dependent preprocessing, no extracted implementation is claimed, and no biological data study is reported. Nothing here reopens the unrestricted classification of [3].

11.3 Supplier excess and width

Bounded-width inputs admit a general logical route. On the two-sorted incidence structure with unary predicates for species, reactions and food and binary relations for reactant, product and catalyst incidence, define for a reaction set S and a species set U

$$\text{Closed}(S, U) \iff F \subseteq U \wedge \forall r \in S (\text{in}(r) \subseteq U \Rightarrow \text{out}(r) \subseteq U).$$

Stage induction places every generated molecule in every such U , and the least closure is itself closed, so $x \in \text{cl}_Q(S)$ if and only if x belongs to every U with $\text{Closed}(S, U)$. Substituting into (2) and quantifying over proper nonempty subsets for minimality gives a fixed monadic second-order formula for original irreducibility, and Courcelle’s enumeration theorem [5, 6] then yields linear-delay enumeration after preprocessing on inputs of bounded incidence treewidth. Section 5.2 shows that the two parameters are incomparable: the gate family has pathwidth at most 7 and unbounded β , while the dense chain has $\beta = 0$ and unbounded treewidth. The metatheorem’s constants are not a practical state-size result, and an explicit width-based extension oracle with a useful state representation for original minimality remains open.

11.4 Open problems

- (1) An explicit dynamic-programming extension oracle for original irrRAF’s on bounded-width inputs, with a proved state bound and delay.

- (2) Preprocessing that removes irrelevant or dependent supplier choices (Remark 4.5) while preserving every original irrRAF.
- (3) Streaming or memory-bounded enumeration with guarantees stronger than the buffered $O(mnL)$ storage.
- (4) Exact minimum-cost destructive deletion for classes beyond $\beta = 0$; Corollary 8.1 is an approximation.
- (5) Output-sensitive enumeration of all minimal destructive cuts, beyond the incremental quasi-polynomial bound inherited from [14].
- (6) Whether the frequency bound $f_r \leq L$ and the membership bound mL are simultaneously attainable, and the exact extremal constant in place of $2/3$.
- (7) Biological instantiations under explicit reaction, food, catalyst, reversibility and gene-action conventions, and the distribution of β and $\beta_{\max}$ on curated metabolic networks.

These are successor directions, not missing assumptions of the supplier theorem.

A Reproduction

All reusable proofs are in `proofs/RAFStructuredEnumeration/` of [4]. Experiment sources, verification receipts, exact test records and the raw study data are in the workspace directory `problem_workspaces/RAF_tractable_irrRAF_enumeration_structural_restrictions/` of the repository: `experiments/supplier_enumerator.py` (global loop), `experiments/component_enumerator.py` (component loop with counters), `experiments/postproof_checks.py` (regression suite, independent oracle), `experiments/postproof_scaling.py` (mechanism study), `results/postproof_checks.json` and `results/postproof_scaling.json` (records), and the receipts `results/root.verify.json`, `results/overlap.verify.json`, `results/restriction.verify.json`, `results/supplier_resolution.verify.json`, `results/earliest.verify.json` and `results/deterministic.verify.json`.

From the repository root, the script `reproduce_postproof.ps1` in that workspace checks the locked Python environment, reruns the regression suite and the mechanism study under a process guard, audits the preserved root receipt against the current sources, and rebuilds the original manuscript; with the switch `-Recompile` it also recompiles the overlap and restriction endpoints through the strict bridge. The original `reproduce.ps1` rebuilds the root itself. A receipt is produced by

```
python scripts/verify_proof.py proofs/RAFStructuredEnumeration/Resolution.lean \
--declaration RAFStructuredEnumeration.supplierEnumeration_correct \
--timeout 300 --output root.verify.json
```

and records the toolchain, the dependency manifest hash, the source hash, the elaborated type and axioms of every exported declaration, and the compiler output under `-DwarningAsError=true`.

The source of this paper is accompanied by a script that replays every printed number from the saved records and recomputes every example of Sections 3–8 with a standard-library exhaustive reference, and by the script that regenerates Table 3 and Figure 3 from the saved 29-run data.

References

- [1] Vašek Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. doi:10.1287/moor.4.3.233.

- [2] Anonymous. Certifying complete families of irreducible autocatalytic sets is co-W[P]-complete. Manuscript, 7 September 2026, 2026.
- [3] Anonymous. The complexity of complete irreducible autocatalytic families: output-polynomial enumeration of irreducible RAFs if and only if $P = NP$. Manuscript, 14 September 2026, 2026.
- [4] Anonymous. Companion RAF proof and experiment repository. <https://github.com/mmislai/Erdos>, 2026. Lean sources under `proofs/RAFStructuredEnumeration/`; experiment sources, verification receipts and data under `problem_workspaces/RAF_tractable_irrRAF_enumeration_structural_restrictions/`. Local snapshot of 14 September 2026.
- [5] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:[10.1016/0890-5401\(90\)90043-H](https://doi.org/10.1016/0890-5401(90)90043-H).
- [6] Bruno Courcelle. Linear delay enumeration and monadic second-order logic. *Discrete Applied Mathematics*, 157(12):2675–2700, 2009. doi:[10.1016/j.dam.2008.08.021](https://doi.org/10.1016/j.dam.2008.08.021).
- [7] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, Cham, 2015. doi:[10.1007/978-3-319-21275-3](https://doi.org/10.1007/978-3-319-21275-3).
- [8] 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](https://doi.org/10.1007/978-3-030-79876-5_37).
- [9] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 624–633. ACM, 2014. doi:[10.1145/2591796.2591884](https://doi.org/10.1145/2591796.2591884).
- [10] 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](https://doi.org/10.1007/978-1-4471-5559-1).
- [11] 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](https://doi.org/10.1137/S0097539793250299).
- [12] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998. doi:[10.1145/285055.285059](https://doi.org/10.1145/285055.285059).
- [13] Edward Fredkin. Trie memory. *Communications of the ACM*, 3(9):490–499, 1960. doi:[10.1145/367390.367400](https://doi.org/10.1145/367390.367400).
- [14] 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](https://doi.org/10.1006/jagm.1996.0062).
- [15] Wim Hordijk. Algorithms for enumerating irreducible and closed autocatalytic sets. Preprints.org, manuscript 202604.1787, version 1, posted 24 April 2026; not peer reviewed, 2026.
- [16] 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](https://doi.org/10.1016/j.jtbi.2003.11.020).

- [17] 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](https://doi.org/10.1007/s10441-012-9165-1).
- [18] 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](https://doi.org/10.1186/s13015-015-0042-8).
- [19] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974. doi:[10.1016/S0022-0000\(74\)80044-9](https://doi.org/10.1016/S0022-0000(74)80044-9).
- [20] 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](https://doi.org/10.1016/0020-0190(88)90065-8).
- [21] Stuart A. Kauffman. Autocatalytic sets of proteins. *Journal of Theoretical Biology*, 119(1):1–24, 1986. doi:[10.1016/S0022-5193\(86\)80047-9](https://doi.org/10.1016/S0022-5193(86)80047-9).
- [22] László Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13(4):383–390, 1975. doi:[10.1016/0012-365X\(75\)90058-8](https://doi.org/10.1016/0012-365X(75)90058-8).
- [23] Micha Sharir. A strong-connectivity algorithm and its applications in data flow analysis. *Computers & Mathematics with Applications*, 7(1):67–72, 1981. doi:[10.1016/0898-1221\(81\)90008-0](https://doi.org/10.1016/0898-1221(81)90008-0).
- [24] 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](https://doi.org/10.1186/s13322-015-0009-7).
- [25] Mike Steel and Wim Hordijk. Tractable models of self-sustaining autocatalytic networks, 2018. arXiv:1801.03953v2, version dated 27 March 2018.
- [26] 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](https://doi.org/10.1016/j.jtbi.2013.04.032).
- [27] 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](https://doi.org/10.1098/rsif.2018.0808).
- [28] Yann Strozecki. Enumeration complexity: incremental time, delay and space, 2023. Habilitation manuscript, arXiv:2309.17042.
- [29] Robert E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972. doi:[10.1137/0201010](https://doi.org/10.1137/0201010).
- [30] 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).
- [31] 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).