

Exact common-activity compatibility of autocatalytic cores: finite cycle acceleration, monotone elimination, and fixed-parameter tractability in the longest path

20 September 2026

Abstract

Autocatalytic cores that are individually realizable under thermodynamically consistent mass action need not be realizable together, because every reaction touching a shared species sees the same activity. We study this compatibility problem for weighted two-reaction cores $X_u \rightleftharpoons X_v$, $X_v + F \rightleftharpoons 2X_u$ placed on the edges of an *arbitrary* finite oriented graph, with fixed positive rational kinetic factors and an independent rational activity box for every species: is there one activity vector at which both internal species of every core are strictly net produced? At a fixed margin the constraints are monotone two-variable implications whose feasible set has a least element, and we give two exact algorithms that compute it, decide strict compatibility, and return a rational witness. The first accelerates label-correcting propagation by *least admissible fixed points of rooted cycle maps*: every jump is forced on all feasible states, cycle-root tokens prove termination where plain iteration converges forever, and every value that arises has *bounded symbolic ancestry*, one box endpoint or one fixed cycle root transported along one simple path. Keeping the roots free during one-block quantifier elimination reduces every comparison to three free variables, the margin is handled as a positive infinitesimal by deciding each comparison for all sufficiently small margins, and the cost is $(J+1)n(h+2)^{O(h)}\text{poly}(B)$ bit operations, where n is the number of species, B the input bit length, h the number of edges of a longest undirected simple path, and J the number of jumps; always $J \leq (n+2)^{O(h)}$, and $J = O(n)(h+2)^{O(h)}$ on forests, windmills and every class with linearly many simple cycles. The second algorithm eliminates the species bottom-up along a depth-first forest. Because monotone two-variable implications are closed under exact elimination of a variable, with a *next-admissible-value* map recording the disconnected projections, all intermediate constraints stay two-variable, their envelopes have near-linearly many pieces by Davenport–Schinzel theory, and strict compatibility is decided in $f(h)n^2\text{poly}(B)$ bit operations: the problem is fixed-parameter tractable in h , equivalently in the treedepth. Polynomial time in n alone would require deciding the sign of an n -fold composition of rational quadratics at a rational point, already on a directed path. We also derive the least-inventory property of the returned state, an exact capacity scale with universal ceiling $(3 - 2\sqrt{2})/8$, a reduction of interval-ratio robustness to the same algorithms with the sharp obstruction $r^+ \geq 2r^-$, an all-size certified example with tolerance radius $19/301$, and an extension to cores of any fixed order. The order-theoretic core, the forced-root and finite-progress lemmas, the vertex-elimination lemma, the source bridge, the quantitative corollaries and all example arithmetic are compiled in Lean 4 with Mathlib; the real-algebraic execution and the complexity analysis are conventional proofs and are marked as such.

1 Introduction

Background. A *potential autocatalytic core* is a minimal motif of a reaction network that admits a strictly productive reaction current [6, 22]. Whether that current is realized by an actual concentration vector is a separate question. Under mass action with fixed barrier

factors, the net current of a reversible reaction is a positive multiple of the difference of its reactant and product activity monomials, so the stoichiometric powers of the activities constrain the direction and the magnitude of every current at once. Kosci, Kuperberg, Rajon and Charlat proved that every isolated core is realizable in this sense, and exhibited two cores that share a species and admit a common productive current but no common concentration vector [22]. Companion manuscripts quantified this phenomenon for pairs of cores [10], showed that minimally incompatible families of two-reaction cores exist in every order, so that no test on bounded subfamilies decides compatibility [9], and decided compatibility exactly for *junction-path assemblies*, in which the cores form directed paths with private interiors between k shared junction species, at a cost exponential in k [8].

This paper removes the architectural restriction. The cores sit on the edges of an arbitrary finite oriented graph; every vertex may be shared by any number of cores, undirected cycles are allowed, the kinetic factors differ from core to core, and every species has its own activity box. The question is the same: is there *one* activity vector, with one value per species, at which *every* core is strictly productive?

The problem. Let $D = (V, E)$ be a finite oriented simple graph: the underlying undirected graph has no loops or multiple edges, and every edge has one orientation. An edge $e: u \rightarrow v$ owns two reversible channels

$$X_u \rightleftharpoons X_v, \quad X_v + F \rightleftharpoons 2X_u, \quad (1)$$

with fixed factors $a_e, b_e \in \mathbb{Q}_{>0}$ and food F held at activity one. With endpoint activities $x = x_u$ and $y = x_v$ the two net currents and the net productions of the two internal species of the core are

$$p_e = a_e(x - y), \quad q_e = b_e(y - x^2), \quad R_{e,u} = 2q_e - p_e, \quad R_{e,v} = p_e - q_e. \quad (2)$$

Each species carries a rational box $[\ell_v, u_v]$ with $0 < \ell_v \leq u_v \leq 1$; singleton boxes are allowed. A state $x \in \prod_v [\ell_v, u_v]$ is *productive* if $R_{e,u} > 0$ and $R_{e,v} > 0$ for *every* edge, separately. Sums of residuals over the edges incident to a species define a different, weaker problem, which is not studied here.

Throughout, $n = |V|$, $B \geq 1$ bounds the binary lengths of the numerators and denominators of all input rationals, and h is the largest number of edges of a simple path in the underlying undirected graph ($h = 0$ if $E = \emptyset$). Bounded h is the same as bounded treedepth (Proposition 8.1); it does not bound the number of species, of cores, of undirected cycles, or of vertices of high degree. The parameter is never supplied to the algorithms.

Theorem 1.1 (Cycle acceleration). *For every instance, Algorithm 2 decides exactly whether a productive state exists and, if so, returns one in $\mathbb{Q}^V$. There are absolute constants C, c such that it uses at most*

$$(J + 1) n (h + 2)^{C(h+1)} (B + \log(n + 2) + 1)^c \quad (3)$$

bit operations, where J is the number of cycle jumps it performs. If σ denotes the number of rooted simple cycles of the implication graph, then $J \leq \sigma (h + 2)^{C(h+1)}$ and $\sigma \leq 2hn + (n + 1)^{h+1}$, so the cost is always at most $(n + 2)^{C'(h+1)} (B + \log(n + 2) + 1)^c$, with an exponent linear in h . On every class of graphs with $\sigma = O(n)$, such as forests and the windmills of Section 8, the cost is $n^2 (h + 2)^{O(h)} \text{poly}(B)$.

Theorem 1.2 (Fixed-parameter tractability). *There is a computable function f and an absolute constant c such that Algorithm 3 decides exactly whether a productive state exists, and returns a rational one if so, in at most*

$$f(h) n^2 (B + \log(n + 2) + 1)^c$$

bit operations. Strict common-activity compatibility is therefore fixed-parameter tractable in the longest path h , equivalently in the treedepth.

Proposition 1.3 (What polynomial time would require). *Let $x_0, \lambda \in \mathbb{Q} \cap (0, 1)$ and let $F_i(x) = (\rho_i x + x^2)/(\rho_i + 1)$ with $\rho_i \in \mathbb{Q}_{>0}$, $i = 1, \dots, m$. The directed path $v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_m$ with ratios ρ_i , box $\{x_0\}$ at v_0 and boxes $[\lambda, 1]$ elsewhere is compatible if and only if*

$$F_m \circ \dots \circ F_1(x_0) > \lambda.$$

Hence an algorithm for strict compatibility that is polynomial in n and B on all graphs decides the sign of $F_m \circ \dots \circ F_1(x_0) - \lambda$, a rational number whose exact representation has $2^m B$ bits, in time polynomial in m and B .

We know of no such sign algorithm; general sign problems for numbers given by arithmetic circuits are only known to lie in the counting hierarchy [1]. We do not claim hardness: the compositions here are special, and the instance has $h = n - 1$. The proposition explains why a structural parameter enters, and why the dependence on it is at least as bad as the cost of exact arithmetic with numbers of degree and height exponential in h .

Decidability itself is not new: the question is a sentence of the existential theory of the reals, decidable [7, 32] in time singly exponential in the number of variables [4, 19, 27]. What is new is the order structure, two algorithms that exploit it, a finite description of the answer, and the resulting dependence on the graph.

Mechanism of Theorem 1.1.

- (i) *Order structure* (Section 2). A core is productive exactly when $G_e(x_u) < x_v < F_e(x_u)$ for two increasing quadratic responses. At a fixed margin $t \geq 0$ each edge therefore gives two monotone implications $x_v \geq G_e(x_u) + t$ and $x_u \geq F_e^{-1}(x_v + t)$. The feasible set $\mathcal{S}_t$ is closed under coordinatewise minimum, so it has a least element when nonempty.
- (ii) *Exact cycle acceleration* (Section 3). Label-correcting propagation of lower bounds either stabilizes within $h + 1$ rounds or closes an implication cycle on which the bound would keep growing; plain iteration then converges to a fixed point of the cycle map without reaching it. We jump to the *least fixed point of the cycle map above the current value*. Every feasible state dominates the jump target, each jump consumes a pair (rooted simple cycle, root) that can never be used again, and analyticity makes the set of such pairs finite.
- (iii) *Bounded ancestry and branch-preserving elimination* (Section 4). Only *anchors* are raised; propagated values are recomputed from the anchors. Every value is therefore an original lower endpoint or one cycle root, transported along one simple path, no matter how many jumps occurred. The internal values of such a label are uniquely determined by its seed, so they can be eliminated in one existential block while the seed stays free, and every comparison lives in the margin and at most two seeds.
- (iv) *The margin as a positive infinitesimal* (Section 5). Testing smaller and smaller margins never certifies strict infeasibility. Instead every comparison is decided *for all sufficiently small positive margins*, by a cylindrical decomposition of its own three-variable formula. The run is then simultaneously the run at every real margin below an explicit dyadic threshold determined by the comparisons actually made, which decides strictness and yields a rational witness by downward rounding.

Mechanism of Theorem 1.2. Eliminating one variable z_w from a system of monotone implications $z_j \geq f(z_i)$ gives again such a system: every incoming map is composed with

every outgoing map through the *next-map* ν_w , which sends a lower bound a to the least admissible value of z_w at or above a (Lemma 7.1, compiled in Lean). The admissible set of z_w is in general a disconnected union of intervals, which is why interval messages fail on cyclic graphs; the next-map records it without leaving the class of monotone two-variable constraints. Along a depth-first forest every generated constraint joins two ancestors, the maps between a pair are merged into one upper envelope, and envelope sizes add up along the forest, up to a Davenport–Schinzel factor, instead of multiplying. This is the nonlinear counterpart of Fourier–Motzkin elimination for monotone linear systems with two variables per inequality [20].

The two algorithms are complementary. Cycle acceleration has single-exponential arithmetic in h , works with three free variables, and produces an explicit description of the least state of bounded ancestry, but its guarantee depends on the number of jumps. Elimination is fixed-parameter tractable, but composes maps along walks of length up to 2^{h+1} , so that f is an iterated exponential.

Further results. The least state minimizes every nondecreasing inventory cost at its margin (Corollary 9.1). The largest feasible margin δ_* satisfies $\delta_* \leq \min_e \rho_e / (8(\rho_e + 1)(\rho_e + 2)) \leq (3 - 2\sqrt{2})/8$ in the ratios $\rho_e = a_e/b_e$, with equality in the first inequality for a single edge (Proposition 9.2). If every ratio is only known to lie in an interval $[r_e^-, r_e^+]$, one state serves all realizations exactly when $G_{r_e^+}(x_u) < x_v < F_{r_e^-}(x_u)$ on every edge; this is decided by the same algorithms, and $r_e^+ \geq 2r_e^-$ is a sharp local obstruction (Proposition 9.3). Section 10 gives a rational certificate for a three-core module, with its sharp relative factor tolerance $19/301$, a joint activity–factor uncertainty box, exact food accounting, and its replication to a family of all sizes with $h = 4$. Section 11 isolates the hypotheses used and extends the results to cores $X_u \rightleftharpoons X_v$, $X_v + (d - 1)F \rightleftharpoons dX_u$ of any fixed order $d \geq 2$.

Related work. Systems of *linear* inequalities with two variables per inequality have a classical theory in which cycles of the constraint graph are closed into single-variable bounds, the loop residues of Shostak [30], leading to polynomial and strongly polynomial algorithms [2, 13, 20]. Our implications are two-variable and monotone but quadratic or inverse-quadratic; composing them along a cycle gives an algebraic map of exponential degree with several fixed points, so that closing a cycle means selecting a particular root, feasible projections are disconnected, binary search on a variable is unsound, and strictness is not a linear-programming matter. Feasible sets closed under coordinatewise minimum are the min-closed constraints of Jeavons and Cooper [21], a tractable class over finite ordered domains [11]; here the domains are real intervals and the difficulty is the exact finite representation of the least solution. Least fixed points of monotone polynomial systems are computed by Kleene and Newton iteration, which in general only converge [16, 17], and by strategy or policy iteration in static analysis [12, 18]. Remark 3.6 records an exact example, compiled in Lean, in which the least solution of a policy that is tight at the true least state is not the true least state; this is why our jumps are forced lower bounds rather than solutions of selected subsystems. Upper envelopes of functions with boundedly many pairwise crossings are governed by Davenport–Schinzel sequences [29]. Graphs without long paths are the graphs of bounded treedepth [25]. We make no claim of a lower bound against any method.

What is verified, and what is not claimed. Section 12 lists the Lean 4 [15, 33] declarations that accompany the paper. They cover the source identities, the explicit nonnegative inverse and the equivalence of boxed margins with the two implications, least-state existence, the forced-root and no-root-rejection lemmas, finiteness of cycle roots under analyticity, the token argument for finite progress, the vertex-elimination lemma with its next-map, transport of local

tests to a global state, the equivalence of strict productivity with weak feasibility at a stable threshold, the one-sided rounding estimate, the capacity ceiling, the ratio monotonicity and robustness obstruction, the order- d band, the inventory corollary, and all example arithmetic. Graph traversal, quantifier elimination, cylindrical decomposition, envelope bounds and both complexity theorems are conventional mathematics; no end-to-end verified solver is claimed, and no implementation of the complete algorithms is benchmarked. The results are static: they concern the existence of one activity vector at which every core is productive, not steady states, persistence or long-time behaviour. The kinetic model is a stipulated normalized one (Section 10.4), not a calibrated biochemical mechanism.

2 Model, response band and fixed-margin implications

2.1 The one-edge band

For factors $a, b > 0$ define the *lower* and *upper responses* on $[0, \infty)$,

$$G_{a,b}(x) = \frac{ax + 2bx^2}{a + 2b}, \quad F_{a,b}(x) = \frac{ax + bx^2}{a + b}, \quad (4)$$

and write G_e, F_e for the responses of the edge e .

Lemma 2.1 (One-edge band). *For all real x, y ,*

$$R_{e,u} = (a + 2b)(y - G(x)), \quad R_{e,v} = (a + b)(F(x) - y), \quad (5)$$

so the core is productive at $(x_u, x_v) = (x, y)$ if and only if $G(x) < y < F(x)$. Both responses are strictly increasing bijections of $[0, \infty)$, fix 0 and 1, depend on (a, b) only through $\rho = a/b$, and satisfy

$$x^2 < G(x) < F(x) < x, \quad F(x) - G(x) = \frac{abx(1-x)}{(a+b)(a+2b)} = \frac{\rho x(1-x)}{(\rho+1)(\rho+2)} \quad (0 < x < 1). \quad (6)$$

On $[0, 1]$ one has $0 < G' \leq 2$ and $0 < F' \leq 2$. Moreover $R_{e,u} + R_{e,v} = q_e$.

Proof. The identities (5) and (6) are rearrangements of (2). Both responses are convex combinations of x and x^2 , which gives $x^2 < G, F < x$ on $(0, 1)$, the fixed points, and monotonicity. The derivatives are $G'(x) = (a + 4bx)/(a + 2b)$ and $F'(x) = (a + 2bx)/(a + b)$. $\square$

A productive core therefore has $x_v < x_u < 1$: activity strictly decreases along every productive edge. Two consequences are used repeatedly. A *directed* cycle of cores is never compatible. An *undirected* cycle with an acyclic orientation may or may not be: with unit factors the path $A \rightarrow B \rightarrow C$ with the shortcut $A \rightarrow C$ is incompatible, because $G(x) - F(F(x)) = x(1-x)(3x^2 + 9x + 2)/24 > 0$ on $(0, 1)$ [8, 9], whereas Section 10 exhibits the same graph with other factors and a rational productive state. Compatibility thus depends on the factors and the boxes and not only on the graph.

2.2 Margins and the two implications

For $t \geq 0$ let

$$\mathcal{S}_t = \left\{ z \in \prod_v [\ell_v, u_v] : G_e(z_u) + t \leq z_v \text{ and } z_v + t \leq F_e(z_u) \text{ for every } e: u \rightarrow v \right\}. \quad (7)$$

By (5), $z \in \mathcal{S}_t$ means that the *normalized residuals* $R_{e,u}/(a_e + 2b_e)$ and $R_{e,v}/(a_e + b_e)$ are all at least t . The sets decrease in t . If $E \neq \emptyset$, a productive state has finitely many positive

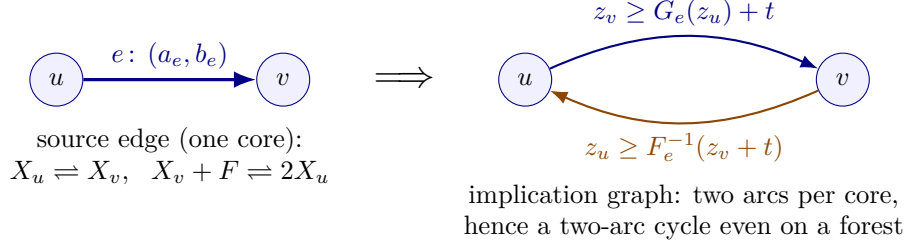


Figure 1: Three kinds of cycle must be kept apart. *Directed source cycles* are never compatible. *Undirected source cycles* with acyclic orientation may be compatible and are allowed. *Implication cycles* live in the bidirected graph used by the algorithms and exist for every instance with at least one core.

normalized residuals, hence lies in $\mathcal{S}_\delta$ for their minimum $\delta > 0$; conversely every element of $\mathcal{S}_t$ with $t > 0$ is productive. So

$$\begin{aligned} \text{a productive state exists} &\iff \mathcal{S}_t \neq \emptyset \text{ for some } t > 0 \\ &\iff \mathcal{S}_t \neq \emptyset \text{ for all sufficiently small } t > 0. \end{aligned} \tag{8}$$

Since F_e is an increasing bijection of $[0, \infty)$ and all activities are positive, each edge $e: u \rightarrow v$ is equivalent to two *implications* of the form $z_j \geq f(z_i)$,

$$\begin{aligned} z_v &\geq g_{e,t}(z_u), \quad g_{e,t}(x) = G_e(x) + t, \\ z_u &\geq k_{e,t}(z_v), \quad k_{e,t}(y) = F_e^{-1}(y + t) = \frac{-\rho + \sqrt{\rho^2 + 4(\rho + 1)(y + t)}}{2}, \end{aligned} \tag{9}$$

where $\rho = a_e/b_e$ and F_e^{-1} is the nonnegative inverse. Both maps are strictly increasing on $[0, \infty)$, map it into itself, and are real analytic on a neighbourhood of it, since the radicand is at least $\rho^2 > 0$ there. For $t > 0$ both are positive at 0. Finite compositions inherit all of these properties.

The *implication graph* has vertex set V and both arcs $u \rightarrow v$ and $v \rightarrow u$ for every source edge (Figure 1); it has $2|E|$ arcs. A *simple implication path* π from s to v visits no vertex twice; the path of length zero is included, with the identity as its map; $f_{\pi,t}$ denotes the composition of the implication maps along π . A *rooted simple implication cycle* c at j returns to j and repeats no other vertex; the two-arc cycles $u \rightarrow v \rightarrow u$ are included; $T_{c,t}$ is its composite map. The subscript t is dropped when the margin is fixed. Because the source graph is simple, an implication cycle with at least three arcs is a simple cycle of the underlying graph, so

$$\text{a simple path has at most } h \text{ arcs, and a rooted simple cycle at most } h + 1. \tag{10}$$

Lemma 2.2 (Min-closure and least state). *For every $t \geq 0$ the set $\mathcal{S}_t$ is compact and closed under coordinatewise minimum. If it is nonempty it has a least element $z_*(t)$, and every coordinate of $z_*(t)$ either equals ℓ_v or is tight in some incoming implication, $z_*(t)_v = f(z_*(t)_i)$.*

Proof. Compactness is clear. Let $z, z' \in \mathcal{S}_t$ and $w = \min(z, z')$, and consider an implication $z_j \geq f(z_i)$ with f increasing. If $w_j = z_j$ then $w_j = z_j \geq f(z_i) \geq f(w_i)$, and symmetrically otherwise. Boxes are preserved. A compact min-closed set has a least element: a minimizer of $\sum_v z_v$ must lie below every other element, since otherwise the minimum with that element would have a smaller sum. If a coordinate of z_* were above ℓ_v and above all incoming implication values, lowering it to the maximum of those would preserve all constraints, as outgoing implications only become easier. $\square$

3 The least state at a fixed margin by exact cycle acceleration

Fix $t \geq 0$. The statements of this section hold for real t . The algorithm maintains a vector of *anchors* $A \in \mathbb{R}^V$, initially $A_v = \ell_v$, and a vector $D \geq A$ of *labels*: each D_v is stored as a vertex s together with a simple implication path π from s to v , and has the value $f_\pi(A_s)$.

Algorithm 1 (fixed-margin closure by label correcting with cycle detection).

1. *Start a phase.* Set $D_v \leftarrow A_v$ with the path of length zero, for every v .
2. *One round.* If $D_v > u_v$ for some v , return INFEASIBLE. Scan all implication arcs $e: i \rightarrow j$ using the values of D at the beginning of the round. If $f_e(D_i) > D_j$, the arc is *violated*; then
 - (a) if j lies on the path of the label D_i , a cycle is *detected*: let c be the rooted simple cycle at j formed by the part of that path from j to i followed by e , and go to step 3;
 - (b) otherwise the path of D_i extended by e is a candidate label for j .

If no arc is violated, return D . Otherwise give every j with candidates the candidate of largest value and start the next round.

3. *Jump.* Let α be the least solution of $T_c(\alpha) = \alpha$ in $(D_j, u_j]$. If there is none, return INFEASIBLE. Otherwise set $A_j \leftarrow \alpha$, leave all other anchors unchanged, and go to step 1.

Three design decisions matter. Labels are only ever extended to simple paths, so no path enumeration is needed and a phase is short (Lemma 3.1). The jump target is the least fixed point *strictly above the current value* D_j , which need not be the least fixed point of T_c . And D is recomputed from the anchors in every phase but never installed as the new anchor vector, so that an anchor is always an original lower endpoint or a root of one fixed cycle map, whose coefficients do not depend on the state of the computation.

Lemma 3.1 (Short phases). *All labels are simple paths from anchors. A label created in round r of a phase has at least r arcs. Consequently a phase has at most $h + 1$ rounds, each of which scans the $2|E| \leq 2hn$ arcs once.*

Proof. Step 2(b) extends a simple path by an arc to a vertex not on it. A label created in round one has one arc. Suppose the arc $e: i \rightarrow j$ is violated in round $r + 1 \geq 2$. If the label D_i had not been created in round r , its value was the same at the beginning of round r , when the arc was scanned. Either it was not violated then, and $D_j \geq f_e(D_i)$ has held ever since because labels only increase; or it was violated, no cycle was detected since the phase continued, and D_j received a candidate of value at least $f_e(D_i)$. Both contradict the violation in round $r + 1$. Hence D_i was created in round r and has at least r arcs by induction, and a label created from it in round $r + 1$ has at least $r + 1$. By (10) a simple path has at most h arcs, so every violated arc in round $h + 1$ leads to a detection. The bound on $|E|$ is Proposition 8.1(a). $\square$

Lemma 3.2 (Forced cycle root). *Assume every element of $\mathcal{S}_t$ dominates A . Then every element of $\mathcal{S}_t$ dominates D throughout the phase. At a detection, $T_c(D_j) > D_j$. If $\mathcal{S}_t \neq \emptyset$, step 3 finds a root α , and every element of $\mathcal{S}_t$ dominates the updated anchors.*

Proof. Let $z \in \mathcal{S}_t$. Along a path $\pi: s \rightsquigarrow v$, monotonicity and the implications satisfied by z give $f_\pi(A_s) \leq f_\pi(z_s) \leq z_v$, hence $D \leq z$. At a detection, the part of the path of D_i that ends at j was the label of j in an earlier round of the phase; its value a satisfies $a \leq D_j$ because labels only increase within a phase. By construction $T_c(a) = f_e(D_i) > D_j \geq a$, and monotonicity of T_c gives $T_c(D_j) \geq T_c(a) > D_j$. Now let $z \in \mathcal{S}_t$. Going around c gives $T_c(z_j) \leq z_j$, and $D_j \leq z_j \leq u_j$. The continuous function $T_c(x) - x$ is positive at D_j and nonpositive at z_j , so it

vanishes somewhere in $(D_j, z_j]$. Its zero set in $[D_j, u_j]$ is closed and does not contain D_j , so it has a least element $\alpha > D_j$, and $\alpha \leq z_j$. Since z was arbitrary, the updated anchor vector is dominated by all of $\mathcal{S}_t$. $\square$

Lemma 3.3 (Finite progress and correctness). *Algorithm 1 terminates. It returns INFEASIBLE if and only if $\mathcal{S}_t = \emptyset$, and otherwise returns the least element $z_*(t)$ of $\mathcal{S}_t$. The number J of jumps is at most the number of pairs (c, α) with c a rooted simple cycle whose map is not the identity and $\alpha \in [0, u_j]$ a fixed point of T_c .*

Proof. Each T_c is real analytic on a connected neighbourhood of $[0, u_j]$. By the identity theorem [23], either T_c is the identity there or $T_c(x) - x$ has finitely many zeros in the compact interval. A detected cycle has $T_c(D_j) > D_j$ and is not of the first kind. There are finitely many rooted simple cycles, so the set of pairs (c, α) in the statement is finite. A jump with token (c, α) sets $A_j = \alpha > D_j \geq A_j^{\text{old}}$, and anchors never decrease, so afterwards $D_j \geq A_j \geq \alpha$ forever. A later jump on the same rooted cycle selects a root strictly above the then current $D_j \geq \alpha$. Hence no token is used twice, and by Lemma 3.1 the algorithm stops.

By Lemma 3.2 the invariant “ $\mathcal{S}_t$ dominates A ” holds throughout. An overflow exit contradicts $D \leq z \leq u$ for $z \in \mathcal{S}_t$, and an exit in step 3 contradicts the existence of a root; so both occur only when $\mathcal{S}_t = \emptyset$. When no arc is violated, D lies in the boxes, because $\ell \leq A \leq D \leq u$, and satisfies all implications, so $D \in \mathcal{S}_t$; being dominated by every element of $\mathcal{S}_t$, it is the least one. $\square$

Margin zero is included: at $t = 0$ some cycle maps can be identities, with a continuum of fixed points (Remark 3.6), but such a cycle is never detected, because detection exhibits a strict violation.

Corollary 3.4 (Bounded ancestry). *Every value that Algorithm 1 computes, and in particular each coordinate of its output, has the form $f_\pi(\sigma)$, where π is a simple implication path from a vertex s and the seed σ is either ℓ_s or a fixed point in $(\ell_s, u_s]$ of the map of one rooted simple cycle at s . By (10) such a description involves at most $2h + 1$ implication maps, independently of the number of jumps.*

Example 3.5 (One unit core). Take one edge $u \rightarrow v$ with $a = b = 1$, boxes $[\frac{1}{10}, \frac{9}{10}]$ and $[\frac{1}{100}, \frac{9}{10}]$, and $t = \frac{1}{100}$. Here $G(x) = (x + 2x^2)/3$ and $F^{-1}(y) = (-1 + \sqrt{1 + 8y})/2$. In round one only the arc $u \rightarrow v$ is violated, and D becomes $(\frac{1}{10}, \frac{1}{20})$ with $D_v = G(\frac{1}{10}) + t$ labelled by the path $u \rightarrow v$. In round two the arc $v \rightarrow u$ is violated, since $F^{-1}(\frac{1}{20} + t) \approx 0.1083 > D_u$, and u lies on the path of D_v : the two-arc cycle at u with map $T_c = F^{-1} \circ (G + 2t)$ is detected. Its fixed points solve $F(x) - G(x) = 2t$, that is $x(1 - x) = 12t$, so the least root above D_u is $\alpha = (5 - \sqrt{13})/10 \approx 0.1394$. In the next phase $D = (\alpha, \alpha - \frac{7}{100})$ satisfies both implications with equality, and the algorithm returns it: the least state is irrational, and it was reached in one jump. Iterating T_c from D_u instead converges to α linearly and never reaches it (Figure 2). In the margin, the two roots of $x(1 - x) = 12t$ merge at $t = \frac{1}{48}$, beyond which $\mathcal{S}_t = \emptyset$.

Remark 3.6 (Why simpler schemes fail). Three exact examples shaped the algorithm; each is replayed by the script of Appendix A.

(a) *Policies.* A tempting scheme selects, for each coordinate, one incoming implication that is tight at the least state (Lemma 2.2), and solves the selected equalities. Take unit cores $x \rightarrow y$ and $x \rightarrow z$, margin $t = \frac{1}{64}$, and boxes $x \in [\frac{1}{8}, \frac{7}{8}]$, $y \in [\frac{1}{16}, \frac{7}{8}]$, $z \in \{\frac{41}{64}\}$. The least state is $(\frac{3}{4}, \frac{41}{64}, \frac{41}{64})$. The policy that uses the two implications of the core $x \rightarrow y$ is tight there, but its own least solution is $(\frac{1}{4}, \frac{9}{64}, \frac{41}{64})$, which violates the omitted constraint $z + t \leq F(x)$ by exactly $\frac{1}{2}$. These statements are compiled (module **Interaction**). A six-species instance in the repository shows that the same failure occurs when the tight policy is unique. This is why

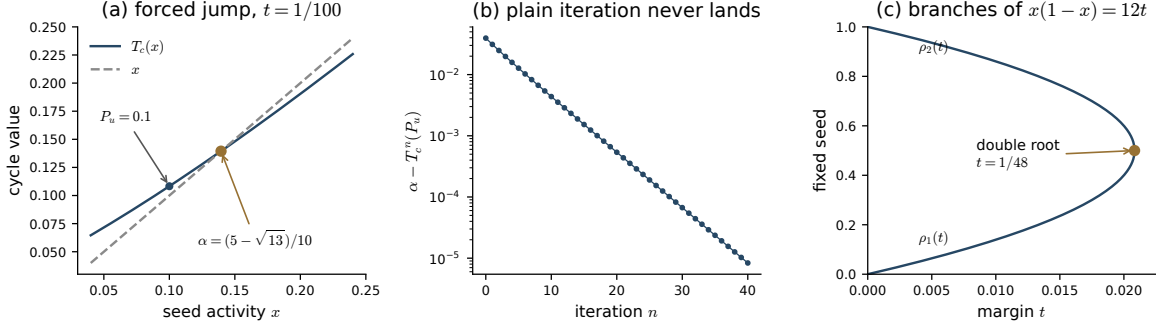


Figure 2: Example 3.5. (a) The detected cycle map lies above the diagonal at the current value, which forces every feasible seed above the next fixed point α . (b) Plain iteration of the same map approaches α geometrically without terminating. (c) The two root branches of the cycle as functions of the margin; they are ordered and continuous for small t and merge in a double root at $t = 1/48$. All curves are exact formulas.

Algorithm 1 never solves a selected subsystem: it only installs lower bounds that are forced on every feasible state, and it keeps every original inequality in the final test.

(b) *Identity cycles at margin zero.* Put unit ratios on a two-core path and ratio 2 on a parallel two-core path with the same endpoints. Since $G_2 = F_1 =: F$, the upper response of the first path equals the lower response of the second, and at $t = 0$ the corresponding implication cycle is the identity, with a continuum of fixed points. For every $t > 0$ the two paths are contradictory, since $F(F(x) - t) - t - (F(F(x) + t) + t) = -t(x^2 + x + 3) < 0$. Positive margins therefore separate finite root sets from identity continua.

(c) *Intervals and binary search.* On forests, feasible projections are intervals. With undirected cycles, two paths between the same species give responses whose order can switch several times, so the projection of the feasible set to one species is a disconnected union of intervals. Interval messages are then unsound, and so is the binary search on a variable that underlies the linear two-variable theory. Section 7 represents such projections exactly by next-maps.

4 Labels and branch-preserving elimination

Algorithm 1 decides $\mathcal{S}_t \neq \emptyset$ for a given t . By (8) strict compatibility asks about all small positive t , and running the algorithm at $t = 1, \frac{1}{2}, \frac{1}{4}, \dots$ never certifies that none works. This section shows that every comparison the algorithm makes has a constant outcome for small positive t , with an explicit meaning of “small”.

4.1 Labels and local tests

A *seed specification* at a vertex s is either the constant ℓ_s , or a rooted simple cycle c at s ; in the second case the admissible seeds at margin t are the elements of

$$\text{Fix}_c(t) = \{\sigma \in [0, u_s] : T_{c,t}(\sigma) = \sigma\}.$$

A *label template* λ at v is a seed specification at some s together with a simple implication path π from s to v ; its value is $\text{val}_\lambda(t, \sigma) = f_{\pi,t}(\sigma)$, where σ is the seed. Intermediate values of a transport are nonnegative but are not required to lie in the species boxes: a path may overshoot, and the box tests detect it.

Lemma 4.1 (Finite fibres). *For $t > 0$ no cycle map is the identity, and every $\text{Fix}_c(t)$ is finite.*

Proof. For $t > 0$ every implication map is positive at 0 and increasing, so $T_{c,t}(0) > 0$. Analyticity on a neighbourhood of $[0, u_s]$ and the identity theorem give finiteness. $\square$

A *local test* is one of the following statements about actual label values:

- (B) a box test $\ell_v \leq \text{val}_\lambda$ or $\text{val}_\lambda \leq u_v$ for a label at v ;
- (E) an edge test $G_e(\text{val}_\lambda) + t \leq \text{val}_\mu$ or $\text{val}_\mu + t \leq F_e(\text{val}_\lambda)$ for an edge $e: u \rightarrow v$, a label λ at u and a label μ at v ;
- (O) an order test $\text{val}_\lambda \leq \text{val}_\mu$ or $\text{val}_\lambda < \text{val}_\mu$ for two labels at the same vertex, which includes the comparison of a cycle root (a label with a path of length zero) with another label.

By Corollary 3.4 these are all the comparisons of Algorithm 1: the violation test of an arc is of type (E), the choice of the largest candidate and the selection of the least root above D_j are of type (O), and the overflow and root-range tests are of type (B). Each test involves at most two seeds.

4.2 Quadratic descriptions and projection with the seeds kept free

With $a = a_1/a_2$, $b = b_1/b_2$ in lowest terms, a forward step $y = G_{a,b}(x) + t$ and a reverse step $y = F_{a,b}^{-1}(x + t)$ are described by

$$(a + 2b)y - ax - 2bx^2 - (a + 2b)t = 0, \quad x, y \geq 0, \quad (11)$$

$$ay + by^2 - (a + b)x - (a + b)t = 0, \quad x, y \geq 0, \quad (12)$$

which after multiplication by a_2b_2 are integer polynomials of degree two with coefficients of at most $2B + 2$ bits. The sign condition $y \geq 0$ in (12) is essential: $ay + by^2$ is strictly increasing from 0 to ∞ on $y \geq 0$, so for nonnegative x and t there is exactly one admissible y .

Lemma 4.2 (Projection with fixed seeds). *Let Θ be a local test, or the membership statement $\sigma \in \text{Fix}_c(t)$. There is a quantifier-free formula Θ^{qf} in t and the at most two seed variables such that, for all $t \geq 0$ and all nonnegative seed values, Θ^{qf} holds if and only if Θ holds for the labels transported from those seeds. It is obtained by eliminating one block of at most $4h + 6$ existentially quantified variables from a conjunction of at most $8h + 16$ polynomial equations and weak or strict inequalities of degree at most two.*

Proof. Introduce one variable for the output of every step of the cycles and paths involved, state (11) or (12) for every step, and add the final comparison, or for membership the equation “last value = σ ” and the bounds $0 \leq \sigma \leq u_s$. By (10) two cycles and two paths have at most $4h + 2$ steps, and a test of type (E) adds one more quadratic inequality. Once t and the seeds are fixed, every step has exactly one admissible output, so by induction there is exactly one admissible tuple of internal values, namely the actual transport. The existential closure over the internal variables is therefore equivalent to the statement about the actual labels. In particular it does *not* replace the given seed by some other root of the same cycle, because the seed is not quantified. Real quantifier elimination [4, 32] provides Θ^{qf} . $\square$

Eliminating the seeds as well would answer the question whether *some* root passes a test. That is not the question Algorithm 1 asks, and it would lose the identity of the branch; the six-species instance of Remark 3.6(a) shows that the wrong root can pass a test that the right root fails.

4.3 Ordered branches for small margins

For a local test Θ with seed variables among (σ, σ') , let $\mathcal{P}_\Theta \subset \mathbb{Z}[t, \sigma, \sigma']$ consist of t , the polynomials of Θ^{qf} , and the polynomials of the membership formulas of the cycles involved. Fix a cylindrical algebraic decomposition (CAD) of $\mathbb{R}^3$, $\mathbb{R}^2$ or $\mathbb{R}$ adapted to $\mathcal{P}_\Theta$, with variable order t, σ, σ' [4, 7]: a partition into finitely many connected cells, arranged in stacks of sections (graphs of continuous functions) and sectors (bands between them) over the cells of the decomposition induced on the first coordinates, such that every polynomial of $\mathcal{P}_\Theta$ has constant sign on every cell. We use complete decompositions, with all cells of lower dimension; no genericity, squarefreeness or well-orientedness is assumed. Do the same for the membership formula of each cycle involved, alone, in the variables (t, σ) . Let $\mathcal{U}_\Theta$ be the finite list of all nonzero univariate polynomials in t that define the base decompositions of these CADs, divided by the largest power of t that divides them, and let $H_\Theta \geq 1$ bound the binary lengths of their integer coefficients. A polynomial $\sum_{i \leq m} c_i t^i$ with integer coefficients and $c_0 \neq 0$ has no complex root of modulus less than $|c_0|/(|c_0| + \max_{i \geq 1} |c_i|)$, by Cauchy's bound applied to the reciprocal polynomial [24]; this is more than $1/(1 + 2^{H_\Theta}) \geq 2^{-(H_\Theta+1)}$. Hence the interval

$$I_\Theta = (0, d_\Theta), \quad d_\Theta = 2^{-(H_\Theta+1)}, \quad (13)$$

contains no root of any member of $\mathcal{U}_\Theta$ and lies in one open base cell of every decomposition attached to Θ .

Lemma 4.3 (Consistent ordered branches). *Let Θ be a local test.*

- (a) *For every cycle c involved in Θ there are an integer $m_c \geq 0$ and continuous functions $\rho_{c,1} < \dots < \rho_{c,m_c}$ on I_Θ such that $\text{Fix}_c(t) = \{\rho_{c,1}(t), \dots, \rho_{c,m_c}(t)\}$ for every $t \in I_\Theta$. Here $\rho_{c,i}(t)$ is the i -th smallest element of $\text{Fix}_c(t)$, so the indexing does not depend on the decomposition or on Θ .*
- (b) *For every choice of branch indices for its seeds, the truth value of Θ at the seeds $\rho_{c,i}(t)$, $\rho_{c',i'}(t)$ is the same for all $t \in I_\Theta$.*

Proof. (a) A polynomial in (t, σ) has constant sign on every cell of the decomposition of the (t, σ) -plane, so the membership formula has a constant truth value on each cell over I_Θ . A sector on which it held would make $\text{Fix}_c(t)$ infinite, contradicting Lemma 4.1. Hence the set $\{(t, \sigma) : t \in I_\Theta, \sigma \in \text{Fix}_c(t)\}$ is a union of sections over I_Θ , which are graphs of continuous functions that do not meet; ordering them gives the $\rho_{c,i}$.

(b) Consider a test with two seeds and its CAD of $\mathbb{R}^3$; the cases with fewer seeds are simpler. Every cell of $\mathbb{R}^3$ over a cell C of the induced decomposition of the (t, σ) -plane projects onto all of C , so a polynomial of $\mathcal{P}_\Theta$ that does not involve σ' has constant sign on C . Thus the induced decomposition is adapted to the membership polynomials of c , and the graph C_i of $\rho_{c,i}$ over I_Θ is one of its sections: a section over I_Θ on which membership holds is the graph of a continuous function with values in $\{\rho_{c,1}(t), \dots, \rho_{c,m_c}(t)\}$, hence one of the $\rho_{c,i}$ by connectedness. In the stack over C_i , the membership formula of c' is constant on each cell and has finite fibres, so it holds only on sections. Such a section is the graph of a continuous function ζ on the connected set C_i with $\zeta(t, \sigma) \in \{\rho_{c',1}(t), \dots, \rho_{c',m_{c'}}(t)\}$; since the $\rho_{c',i'}$ are continuous and strictly ordered on I_Θ , the index is locally constant, hence constant. Consequently the curve $t \mapsto (t, \rho_{c,i}(t), \rho_{c',i'}(t))$, $t \in I_\Theta$, stays inside a single cell of $\mathbb{R}^3$. The formula Θ^{qf} has a constant truth value on that cell, and by Lemma 4.2 this is the truth value of Θ on the actual labels. $\square$

5 The margin as a positive infinitesimal, and rational output

Algorithm 2 (strict decision). If $E = \emptyset$, return ℓ . Otherwise run Algorithm 1 *symbolically* at $t = 0^+$: an anchor is stored as the constant ℓ_s or as a pair (rooted simple cycle, branch index),

a label as an anchor with a simple path, and every local test Θ is answered by its constant truth value on I_Θ from Lemma 4.3; in step 3 the branches of c are enumerated in increasing order and the first one that passes the tests $D_j < \rho_{c,i}$ and $\rho_{c,i} \leq u_j$ is selected. Let $\Theta_1, \dots, \Theta_N$ be the tests performed and put

$$H = \max_k H_{\Theta_k}, \quad t_1 = 2^{-(H+2)}. \quad (14)$$

If the symbolic run returns INFEASIBLE, report that no productive state exists. Otherwise evaluate the returned labels at the margin t_1 and return the rounding of Lemma 5.2.

Theorem 5.1 (Trace invariance). *The symbolic run terminates after at most $\sum_c m_c$ jumps. For every real $t \in (0, 2^{-(H+1)})$, Algorithm 1 at margin t , with the same scanning order, performs the same sequence of tests with the same outcomes, where the branch index i of a cycle c stands for the root $\rho_{c,i}(t)$. Consequently, if the symbolic run returns INFEASIBLE then $\mathcal{S}_t = \emptyset$ for all $t > 0$ and no productive state exists; otherwise its labels evaluated at t form the least element of $\mathcal{S}_t$ for every $t \in (0, 2^{-(H+1)})$, and a productive state exists.*

Proof. Let $t \in (0, 2^{-(H+1)})$, so that $t \in I_{\Theta_k}$ for every k . By induction on the number of tests, the real run at t and the symbolic run are in corresponding states: same anchors and labels under the correspondence $i \mapsto \rho_{c,i}(t)$, which is a bijection onto $\text{Fix}_c(t)$ preserving order. The next test is therefore the same local test, and by Lemma 4.3(b) its real outcome at t is the symbolic outcome. In step 3 the real algorithm selects the least root of T_c in $(D_j, u_j]$, which is the first branch passing the two tests. For the termination of the symbolic run, apply the same argument to each finite initial segment of it, whose tests define some H' : the real run at any $t < 2^{-(H'+1)}$ agrees with that segment, real runs never reuse a token (c, α) by Lemma 3.3, and tokens correspond to pairs (c, i) . So no pair (c, i) is used twice in the symbolic run. The conclusions follow from Lemma 3.3 at each such t , together with the monotonicity of $\mathcal{S}_t$ in t and (8). $\square$

Only the tests that are actually performed are ever decomposed, and the threshold t_1 is as large as those tests allow. A zero optimal margin, as in Remark 3.6(b), is rejected by one terminating run.

Lemma 5.2 (Downward rational reconstruction). *Let $t > 0$, $z \in \mathcal{S}_t$, and let N be a power of two with $1/N \leq t/4$. The vector*

$$q_v = \max\{\ell_v, \lfloor Nz_v \rfloor / N\}$$

is rational, lies in the boxes, and belongs to $\mathcal{S}_{t/2}$; in particular it is productive. Singleton boxes are reproduced exactly. Each q_v is found with $O(\log N)$ comparisons of z_v with dyadic rationals.

Proof. Put $\eta = 1/N$. By construction $\ell_v \leq q_v \leq z_v \leq u_v$ and $z_v - q_v < \eta$. Since rounding is downward and G_e is increasing, $q_v - G_e(q_u) \geq z_v - \eta - G_e(z_u) \geq t - \eta$. Since $0 < F'_e \leq 2$ on $[0, 1]$, $F_e(q_u) - q_v \geq F_e(z_u) - 2\eta - z_v \geq t - 2\eta \geq t/2$. The integer $\lfloor Nz_v \rfloor \in [0, N]$ is located by bisection, each step being a comparison of z_v with a dyadic rational. One rounded value is used for every occurrence of a species. $\square$

The one-sided estimate, compiled as `round_down_margin`, loses 2η where a two-sided perturbation would lose 3η . It shows that an algebraic state of margin $t > \mu \geq 0$ rounds to a rational state of margin at least μ as soon as $\eta \leq (t - \mu)/2$. It does not give a rational state at a prescribed *weak* margin: the least state at margin t can be irrational (Example 3.5).

Corollary 5.3 (A uniform threshold). *Let $H_{\max}$ be the maximum of H_Θ over all local tests of the instance and $t_0 = 2^{-(H_{\max}+2)}$. Then a productive state exists if and only if $\mathcal{S}_{t_0} \neq \emptyset$, and $\mathcal{S}_t \neq \emptyset$ either for all or for no $t \in (0, 2t_0)$.*

Proof. Let $t, t' \in (0, 2t_0)$ and $\mathcal{S}_t \neq \emptyset$. By Corollary 3.4 and Lemma 3.3, each coordinate of $z_*(t)$ is the value of a label template at a seed $\rho_{c,i}(t)$ or ℓ_s . The vector of the same templates and indices at t' passes all tests (B) and (E) that express membership in $\mathcal{S}_{t'}$, by Lemma 4.3(b). The rest is (8). $\square$

An independent proof of the completeness of labels avoids the algorithm: by Lemma 2.2 each coordinate of the least state not at its lower endpoint has a tight incoming implication, and following tight predecessors either reaches a lower endpoint or closes a rooted simple cycle of which the value is a fixed point. The corollary is the form in which the Lean root theorem of Section 12 is stated. Algorithm 2 does not compute t_0 , which would require all $(n+2)^{O(h)}$ templates.

6 Bit complexity of cycle acceleration

A cylindrical decomposition in all n variables would be doubly exponential in n [14]. The analysis rests on two facts from Section 4: the variables eliminated in a block are uniquely determined, so that the block costs singly exponentially in its length $O(h+1)$, and the decompositions live in at most three variables.

We use the following published bounds. (QE) For s polynomials of degree at most d in $m + \ell$ variables with integer coefficients of at most τ bits, one block of m existential quantifiers is eliminated with $s^{(m+1)(\ell+1)} d^{O(m)O(\ell)}$ arithmetic operations; the output has at most that many polynomials, of degree $d^{O(m)}$, and all integers in the computation and the output have at most $\tau d^{O(m)O(\ell)}$ bits [3, Theorem 2.27], [4, Chapter 14]. (CAD) For a fixed number of variables, a complete cylindrical decomposition adapted to S integer polynomials of degree at most D and coefficient length at most L , with sample points and the signs of the polynomials on all cells, is computed with a number of bit operations polynomial in S , D and L [7], [4, Chapter 11]. (RI) The real roots of an integer polynomial are isolated, and compared with rationals and with each other, in bit time polynomial in its degree and coefficient length [4, 28].

Lemma 6.1 (Local arithmetic). *Put $k = h + 1$. There are absolute constants C_0, c_0 such that, with $Q = (k + 1)^{C_0 k}$:*

- (a) *every formula Θ^{af} of Lemma 4.2 has at most Q polynomials, of degree at most Q and coefficient length at most $(B+1)Q$, and is computed in at most $Q(B+1)^{c_0}$ bit operations;*
- (b) *the same bounds hold for the decompositions attached to Θ , so that $H_\Theta \leq (B+1)Q$, every cycle has $m_c \leq Q$ branches, and the outcome of Θ on I_Θ for given branch indices is found within $Q(B+1)^{c_0}$ bit operations;*
- (c) *for a rational t of $O((B+1)Q)$ bits, each local test at given branch indices, and each comparison of a label at given branch index with a rational of $O((B+1)Q)$ bits, is decided in at most $Q(B+1)^{c_0}$ bit operations.*

Proof. (a) Apply (QE) with $m \leq 4h + 6$, $s \leq 8h + 16$, $d = 2$, $\ell \leq 3$ and $\tau \leq 2B + 2$. The operation count is $(8k + 8)^{4(4k+3)} 2^{O(k)} = (k + 1)^{O(k)}$, degrees are $2^{O(k)}$, and integers have $(2B + 2)2^{O(k)}$ bits. Multiplying the operation count by the cost of one arithmetic operation on such integers gives the bit cost.

(b) Apply (CAD) in at most three variables to the at most $3Q + 1$ polynomials of $\mathcal{P}_\Theta$. The cost and the sizes of all projection polynomials, in particular of the members of $\mathcal{U}_\Theta$, are polynomial in Q and $(B+1)Q$; enlarging C_0 and c_0 restores the stated form. The members of $\mathcal{U}_\Theta$ are kept as a list and are not multiplied together. A finite set defined by a quantifier-free formula in one variable consists of roots of its polynomials, which bounds m_c . The cell of the curve of Lemma 4.3(b) is identified from the sample point of the base cell I_Θ by counting sections in the two stacks.

(c) Substituting a rational t into a polynomial of degree at most Q and clearing denominators adds $O(Q \cdot (B + 1)Q)$ bits to each coefficient, which is again absorbed. Membership becomes a univariate formula: isolate the roots of its polynomials by (RI), keep those at which the formula holds, and number them increasingly. A test with two seeds becomes a formula in (σ, σ') , evaluated at the pair of indexed roots by a decomposition of the plane, or equivalently by sign determination at a pair of real algebraic numbers [4, Chapters 10–11]. A rational constant in a comparison is absorbed in the same way. $\square$

Proof of Theorem 1.1. Correctness is Theorem 5.1 and Lemma 5.2. Let $E \neq \emptyset$. By Lemma 3.1 a phase consists of at most $h + 1$ rounds of at most $2hn$ arc scans, each of which is one test (E), one membership lookup in a stored path, and at most one test (O) per candidate, plus n overflow tests (B) per round. A jump enumerates at most Q branches with two tests each. So a phase costs $O(h^2n + Q)$ local tests, each within $Q(B + 1)^{c_0}$ bit operations by Lemma 6.1(b); indices and stored paths have $O(h \log(n + 2))$ bits. There are $J + 1$ phases. No number field generated by earlier roots is ever constructed: by Corollary 3.4 every comparison is between two labels of bounded description.

For the output, t_1 has $H + 2 \leq (B + 1)Q + 2$ bits, and Lemma 5.2 with $\log N = H + 4$ uses $O(H)$ comparisons per species of a label at the rational margin t_1 with a dyadic rational of $O(H)$ bits, each covered by Lemma 6.1(c); the output has $O(n(B + H))$ bits. Altogether the cost is $(J + 1)nQ^{O(1)}(B + \log(n + 2) + 1)^{O(1)}$, which is (3).

By Theorem 5.1, $J \leq \sum_c m_c \leq \sigma Q$. A rooted simple cycle is either one of the $2|E| \leq 2hn$ two-arc cycles (counted with their root) or a sequence of at most $h + 1$ distinct vertices, which gives $\sigma \leq 2hn + (n + 1)^{h+1}$; and $Q \leq (n + 2)^{O(h)}$ because $h + 1 \leq n$. If the underlying graph has $O(n)$ simple cycles of length at least three, counted with their $2(h + 1)$ rootings and directions, then $\sigma = O(hn)$. $\square$

On a forest the only implication cycles are the two-arc cycles, whose fixed-point equation $F_e(x) - G_e(x) = 2t$ is quadratic, so $J \leq 4|E|$ and the cost is $n^2(h + 2)^{O(h)} \text{poly}(B)$. On a windmill W_k (Section 8) there are $6k$ rooted two-arc cycles and $6k$ rooted directed triangles. On the complete bipartite graph $K_{q,n}$, by contrast, $h = 2q$ and $\sigma = n^{\Theta(q)}$; whether J can actually be that large is open (Section 13), and Theorem 1.2 covers such families by a different algorithm.

The bound is a guarantee for the algorithm as specified, with complete projected decompositions as the exact fallback of every sign decision. An implementation that sends the $O(h)$ internal variables to a full-dimensional decomposition, or that merges the algebraic number fields of successive roots, is not covered by this analysis. We have not implemented the complete algorithm, and we make no claim about running times.

7 Fixed-parameter tractability by monotone elimination

7.1 Monotone systems and next-maps

Fix a real margin $t > 0$. Let $\mathcal{M}$ be the class of functions $f: [0, \infty] \rightarrow [0, \infty]$ that are nondecreasing, left-continuous on $[0, \infty)$, and satisfy $f(\infty) = \infty$. It contains the implication maps (9) and is closed under composition and under finite pointwise maxima. A *monotone system* on a vertex set W consists of boxes $[\ell_v, u_v]$, maps $M_{ij} \in \mathcal{M}$ for ordered pairs $i \neq j$ (absent maps are 0), and self-maps $T_v \in \mathcal{M}$ (initially 0). A state $z \in \prod_v [\ell_v, u_v]$ is feasible if $z_j \geq M_{ij}(z_i)$ for all pairs and $z_v \geq T_v(z_v)$ for all v . The original problem at margin t is the monotone system with $M_{uv} = g_{e,t}$ and $M_{vu} = k_{e,t}$ for every edge $e: u \rightarrow v$. The proof of Lemma 2.2 applies verbatim: the feasible set is compact and min-closed, since a constraint $z_j \geq f(z_i)$ with f nondecreasing and left-continuous defines a closed min-closed set.

The *admissible set* and the *next-map* of a vertex w are

$$U_w = \{x \in [\ell_w, u_w] : T_w(x) \leq x\}, \quad \nu_w(a) = \min(U_w \cap [a, \infty)), \quad (15)$$

with $\nu_w(a) = \infty$ if $U_w \cap [a, \infty)$ is empty. The set U_w is closed, because T_w is nondecreasing and left-continuous, so the minimum exists. It is in general a disconnected union of intervals; on a single unit core it is the interval between the two roots of Figure 2(c) intersected with the box. The map ν_w is the identity on U_w , is constant on every gap of U_w , and belongs to $\mathcal{M}$: if $a_k \uparrow a$ then the nondecreasing values $\nu_w(a_k) \in U_w$ converge to a point of the closed set U_w that is at least a and at most $\nu_w(a)$.

Lemma 7.1 (Vertex elimination). *Let w be a vertex of a monotone system on W . Define a system on $W \setminus \{w\}$ by*

$$\begin{aligned} M'_{ij} &= \max(M_{ij}, M_{wj} \circ \nu_w \circ M_{iw}) \quad (i \neq j), & T'_i &= \max(T_i, M_{wi} \circ \nu_w \circ M_{iw}), \\ \ell'_j &= \max(\ell_j, M_{wj}(\nu_w(\ell_w))), & u'_j &= u_j, \end{aligned} \quad (16)$$

where a composite is ∞ wherever ν_w is. A state z' on $W \setminus \{w\}$ is feasible for the new system if and only if it extends to a feasible state of the old one, and the least value of z_w among all extensions is

$$z_w = \nu_w\left(\max\{\ell_w, \max_i M_{iw}(z'_i)\}\right). \quad (17)$$

Proof. Fix z' satisfying the old constraints that do not involve w , and let a be the maximum of ℓ_w and of the numbers $M_{iw}(z'_i)$. A value x extends z' if and only if $x \in U_w$, $x \geq a$, and $M_{wj}(x) \leq z'_j$ for all j . If such an x exists then $\nu_w(a) \leq x$ is finite and $M_{wj}(\nu_w(a)) \leq M_{wj}(x) \leq z'_j$ by monotonicity; conversely, if $\nu_w(a)$ is finite and satisfies these inequalities, it is an extension, and it is the least one. Since ν_w and M_{wj} are nondecreasing, they commute with the maximum defining a , so the condition “ $M_{wj}(\nu_w(a)) \leq z'_j$ for all j ” is the conjunction over all i and j of $z'_j \geq M_{wj}(\nu_w(M_{iw}(z'_i)))$ and of $z'_j \geq M_{wj}(\nu_w(\ell_w))$. For $i \neq j$ these are the new implications, for $i = j$ the new self-constraints, and the last ones the new lower endpoints; finiteness of $\nu_w(a)$ is implied because a composite is ∞ where ν_w is. $\square$

This is the order-theoretic core of the algorithm, and it is compiled in Lean for arbitrary index types and arbitrary monotone maps (declarations `IsNextMap`, `eliminate_vertex_iff`, `eliminated_value_least`; Section 12). The self-constraint $z_i \geq T'_i(z_i)$ is where cycles of the graph go: eliminating v from one unit core $u \rightarrow v$ gives $T'_u = k_{e,t} \circ \nu_v \circ g_{e,t}$, which on the box of v is the cycle map of Example 3.5. The upper endpoint u_w needs no separate rule, because it is part of U_w and an inadmissible lower bound produces the value ∞ .

Algorithm 3 (elimination along a depth-first forest, at a fixed margin). Compute a depth-first forest of the underlying graph. Eliminate the vertices in post-order by Lemma 7.1, storing ν_w , the maps M_{iw} and ℓ_w at the time w is eliminated. For a root r there is nothing to update; if $\nu_r(\ell_r) = \infty$ return INFEASIBLE. Then assign values in reverse order: $z_r = \nu_r(\ell_r)$ for roots, and (17) with the stored data for the other vertices.

Proposition 7.2 (Correctness). *In a depth-first forest every edge joins a vertex to one of its ancestors. Hence, when w is eliminated, all nonzero maps M_{iw}, M_{wj} join w to its ancestors, every generated map joins two ancestors of w , and a vertex has at most h ancestors. Algorithm 3 returns INFEASIBLE if and only if $\mathcal{S}_t = \emptyset$, and otherwise returns the least element of $\mathcal{S}_t$.*

Proof. The structural claims follow by induction from (16). By Lemma 7.1 the system after each elimination is the exact projection of the previous one. After all non-root vertices of a tree have been eliminated, the root is feasible exactly on U_r , with least value $\nu_r(\ell_r)$; the least state projects to the least state of each projected system, and (17) is the least extension of given ancestor values, which are already least. $\square$

The script `sanity_elimination.py` of the source package runs a tabulated floating-point version of Algorithm 3 on 300 random instances on trees, triangles, diamonds, windmills and $K_{2,m}$, and compares the result with the limit of Kleene iteration, which converges to the same least state from below; all verdicts agree and all feasible states agree within the grid tolerance. This is a sanity check of the logic, not a certificate and not the exact algorithm.

7.2 Pieces

All maps that arise are piecewise of a simple kind. A *walk map* is a composition of implication maps (9) along a walk of the implication graph. Call $f \in \mathcal{M}$ *tame* if $[0, 1]$ splits into finitely many intervals on each of which f is a constant, a walk map, or ∞ .

Lemma 7.3 (Tameness and sizes). *Every map produced by Algorithm 3 is tame, its walk maps follow walks of at most 2^{h+1} arcs, and there is a function g such that the total number of pieces of all maps ever produced is at most $g(h)n \log(n+2)$.*

Proof. Tameness. Let L_d bound the walk length for maps whose deeper endpoint has at most d levels of the forest below it. A generated map $M_{wj} \circ \nu_w \circ M_{iw}$ is, on each interval of the common refinement, a constant or the composition of two walk maps with the identity piece of ν_w in between, so $L_d \leq 2L_{d-1}$ and $L_0 = 1$. A non-identity walk map τ is real analytic and $\tau(x) - x$ has at most s_h zeros on $[0, 1]$, where s_h depends only on h : the graph of τ lies on an algebraic curve of degree at most 2^{L_h} , and by Lemma 4.1 it is not the diagonal. Likewise two distinct walk maps, or a walk map and a constant, agree at most s_h times. Hence U_w has at most $(s_h + 1)$ intervals per piece of T_w , and ν_w at most $2(s_h + 1)\pi(T_w) + 1$ pieces, where π denotes the number of pieces. For a nondecreasing inner map, the breakpoints of a composition are breakpoints of the inner map or preimages of breakpoints of the outer one, so $\pi(M_{wj} \circ \nu_w \circ M_{iw}) \leq \pi(M_{iw}) + \pi(\nu_w) + \pi(M_{wj})$.

Envelopes. The final map on a pair of ancestors is the upper envelope of the original map and of one generated map for every common descendant. The pieces are partially defined continuous functions that pairwise agree at most s_h times, or coincide, in which case ties are broken by index. The upper envelope of N such functions has at most $\lambda_{s_h+2}(N)$ pieces, where λ_s is the maximal length of a Davenport–Schinzel sequence of order s [29, Chapter 1], and $\lambda_s(N) \leq C_s N \log^* N$ [31]. Write $\lambda_{s_h+2}(N) \leq \beta N$ with $\beta = C_{s_h+2} \log^* N_{\max}$, where $N_{\max}$ is the final total.

Recursion. Let $\Omega(w)$ be the number of pieces of T_w and of all maps between w and its ancestors when w is eliminated, and $S(w)$ the sum of Ω over the subtree of w . Each map counted in $\Omega(j)$ is an envelope of at most one original piece and of generated maps with at most $(2s_h + 5)\Omega(w)$ pieces for each descendant w of j , and there are at most $2h + 1$ such maps, so with $K = (2h + 1)(2s_h + 5)\beta$,

$$\Omega(j) \leq K \left(1 + \sum_{w \text{ below } j} \Omega(w) \right), \quad S(j) = \Omega(j) + \sum_{c \text{ child of } j} S(c) \leq K + (K + 1) \sum_c S(c).$$

By induction on the height d of j this gives $S(j) \leq K(K + 1)^d |\text{subtree}(j)|$, so the total is at most $K(K + 1)^{h+1}n$. Envelope sizes thus *add* along the forest, up to the factor β . Finally $\log^* N_{\max} \leq \log^* n + O_h(1)$, and $(\log^* n)^{h+2} \leq g_1(h) \log(n + 2)$. $\square$

Without the Davenport–Schinzel bound one would use the trivial quadratic bound on envelopes, the exponent would double at every level, and the result would be a bound of the form $n^{2^{O(h)}}$. With maps in up to h variables instead of monotone maps between two, one is back at the generic nested elimination (Remark 8.2).

7.3 Arithmetic and the proof of Theorem 1.2

Every number that Algorithm 3 computes is a breakpoint or a value of a tame map, and is described recursively: it is a rational endpoint, or the m -th real solution in $[0, 1]$ of an equation $\tau_1(x) = \tau_2(x)$, where each τ_i is the identity, a walk map applied to x , or a walk map applied to a number described earlier, at a vertex strictly lower in the forest. Indeed, endpoints of U_w are solutions of $\tau(x) = x$ on a piece or breakpoints of T_w ; breakpoints of envelopes are crossings of two pieces; breakpoints of compositions are preimages $\tau(x) = \beta$ of earlier breakpoints; and the values in (17) are walk maps applied to such numbers. A description is thus a binary tree of depth at most $h + 1$ whose nodes carry walks of at most 2^{h+1} arcs: it involves at most 2^{2h+3} quadratic steps (11)–(12) with unique admissible branches, at most 2^{h+1} root indices, each bounded by s_h , and $2^{O(h)}$ input rationals. This is bounded ancestry again, with a bound depending on h alone.

Proof of Theorem 1.2. A comparison of two described numbers, with the margin t as a parameter, is a first-order formula over the reals whose size, number of variables and degrees are bounded in terms of h , with $2^{O(h)}$ integer coefficients of $O(B)$ bits. Its truth value is constant on an interval $(0, d)$ with $d \geq 2^{-f_1(h)(B+1)}$, and this value, the bound d , and the truth value at a given rational t of $f_1(h)(B + 1)$ bits are computed in $f_1(h)(B + 1)^c$ bit operations by quantifier elimination and a decomposition of the t -axis, exactly as in Section 4 but with more variables [4, Chapters 11 and 14]; bit lengths enter these bounds polynomially with an absolute exponent. Run Algorithm 3 symbolically at $t = 0^+$. As in Theorem 5.1, the run coincides with the run at every real margin below the minimum of the thresholds of the comparisons performed, so by Proposition 7.2 and (8) it decides strict compatibility, and the labels of the returned state, evaluated at a dyadic margin t_1 below that minimum, are rounded by Lemma 5.2 with $O(f_1(h)(B + 1))$ comparisons per species.

The number of comparisons is governed by Lemma 7.3. Eliminating w computes ν_w by solving $\tau(x) = x$ on each piece of T_w , and for each of the at most h^2 ordered pairs of ancestors one composition, by merging sorted breakpoint lists, and one envelope update. The upper envelope of maps with N pieces in total is computed by divide and conquer with $O(\lambda_{s_h+2}(N) \log N)$ comparisons and crossing computations [29]; to keep this bound the maps generated for a pair are collected and merged once, when the deeper vertex of the pair is eliminated. Altogether there are at most $g_2(h) n \log^2(n + 2) \leq g_2(h) n^2$ operations on described numbers. The function f is computable, since s_h , C_s and the constants of the cited algorithms are. $\square$

Remark 7.4 (The two guarantees). Theorem 1.2 settles the parameterized status of the problem, but f is large: walks of length 2^{h+1} give algebraic degrees $2^{2^{h+1}}$, descriptions with $2^{O(h)}$ variables, and Davenport–Schinzel constants of order s_h . Theorem 1.1 has arithmetic $(h + 2)^{O(h)}$ in three free variables and an explicit least state with $2h + 1$ steps of ancestry, at the price of the factor J . Restricting generated maps to simple paths would shorten the walks of Algorithm 3, but then a generated map depends on which vertices were used, and the two-variable closure of Lemma 7.1 is lost. An algorithm with cost $2^{\text{poly}(h)} \text{poly}(n, B)$ is not excluded by Proposition 1.3 and is open.

8 What the parameter buys

Proposition 8.1 (Graphs without long paths). *Let the underlying undirected graph have n vertices, m edges and longest simple path of h edges.*

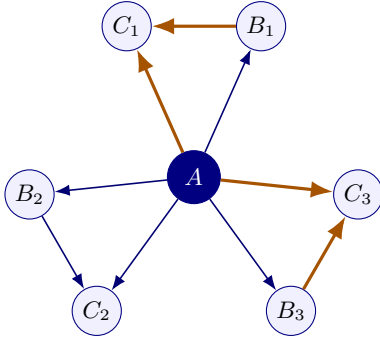
(a) $m \leq hn$.

- (b) The treedepth satisfies $\text{td} \leq h + 1$ and $h \leq 2^{\text{td}} - 2$. Bounded h and bounded treedepth thus define the same graph classes.
- (c) For every $k \geq 2$ the windmill W_k , consisting of k triangles $A \rightarrow B_i$, $B_i \rightarrow C_i$, $A \rightarrow C_i$ that share the hub A , has $2k + 1$ species, $3k$ cores, k independent undirected cycles, and $h = 4$.

Proof. (a) In a depth-first forest every edge joins a vertex to one of its ancestors, and the tree path from a root to a vertex is simple, so every vertex has at most h ancestors; charge each edge to its deeper endpoint. (b) The same forest is an elimination forest of height at most $h + 1$; conversely a path with 2^d vertices has treedepth $d + 1$ [25, Chapter 6]. (c) A simple path passes through A at most once, so it meets at most two triangles and uses at most two edges of each; B_i, C_i, A, C_j, B_j attains four. $\square$

Family	h	Theorem 1.1	Theorem 1.2
Star with any number of leaves	2	$O(n^2) \text{poly}(B)$	$O(n^2) \text{poly}(B)$
Trees of rooted height r , any branching	$\leq 2r$	$n^2(r + 1)^{O(r)} \text{poly}(B)$	$f(2r) n^2 \text{poly}(B)$
Windmill W_k (Figure 3)	4	$O(n^2) \text{poly}(B)$	$O(n^2) \text{poly}(B)$
Complete bipartite $K_{q,n}$	$2q$	$n^{O(q)} \text{poly}(B)$	$f(2q) n^2 \text{poly}(B)$
Balanced binary tree	$\Theta(\log n)$	$n^{O(\log \log n)} \text{poly}(B)$	no useful bound
Path with n vertices	$n - 1$	$2^{O(n \log n)} \text{poly}(B)$	no useful bound
Complete graph (diameter one)	$n - 1$	$2^{O(n \log n)} \text{poly}(B)$	no useful bound

Table 1: The two guarantees on some families. On forests $J \leq 4|E|$, which gives the entries for trees; on a path Proposition 1.3 applies. Small diameter does not help, and sparsity does not help; by Proposition 8.1(a), graphs with bounded h are sparse, but not conversely.



W_3 : three modules share the hub A .

Thick: a longest simple path, $h = 4$ for all $k \geq 2$.

Every C_i has in-degree two, so a junction–path representation [8] needs $k + 1$ junctions.

Figure 3: The windmill family. Each arrow is a two-channel core (1). The number of undirected cycles and the number of shared branching species grow with k , while h stays at four.

On W_k the guarantees at hand compare as follows. Interval elimination on forests does not apply, since every module is an undirected cycle. In a junction–path decomposition [8], an interior vertex of a path has in-degree and out-degree one among the selected cores, so the hub and every C_i are junctions, and the bound $\mathcal{A}^{O(k+1)}$ of that paper has an exponent that grows with k . A decision procedure for the existential theory of the reals that ignores the graph has a guarantee exponential in the number $2k + 1$ of variables [3, 27]. Theorems 1.1 and 1.2 give a fixed polynomial in k and B , for *all* rational factors and boxes on these graphs, including incompatible and zero-margin instances. These are comparisons of proved upper bounds. Windmills have a simple block structure, and nothing here shows that other methods are slow on them.

Remark 8.2 (Generic nested elimination). Proposition 8.1(b) also suggests the generic alternative: eliminate the variables one at a time from the leaves of a depth-first forest by real quantifier elimination, with the at most h ancestor variables free. This runs in polynomial time for every fixed h , so polynomiality on fixed- h classes is within the reach of general tools. But the messages are semialgebraic sets in up to h variables, each level raises the number of polynomials to a power of order $2(h+1)$, and a routine estimate gives an exponent of n of order $(2h+2)^{h+1}$; we have not tried to optimize this estimate, which is an upper bound on one strategy. Algorithm 3 is the same elimination order carried out inside the class of monotone two-variable constraints, which is what keeps the exponent of n fixed.

Proof of Proposition 1.3. Let G_P and F_P be the compositions of the lower and upper responses along the path. If z is productive then $z_0 = x_0$ and, by induction with Lemma 2.1 and monotonicity, $G_P(x_0) < z_m < F_P(x_0)$; with $z_m \geq \lambda$ this gives $F_P(x_0) > \lambda$. Conversely let $F_P(x_0) > \lambda$ and choose $y \geq \lambda$ with $G_P(x_0) < y < F_P(x_0)$, which is possible because $G_P < F_P$ on $(0, 1)$. For $\theta \in [0, 1]$ let $z_0 = x_0$ and $z_i = (1 - \theta)G_i(z_{i-1}) + \theta F_i(z_{i-1})$. The endpoint z_m depends continuously on θ and runs from $G_P(x_0)$ to $F_P(x_0)$, so $z_m = y$ for some $\theta \in (0, 1)$, and then every core is productive by (6). Activities decrease along the path, so $z_i \in [y, x_0] \subseteq [\lambda, 1]$. Each F_i squares the denominator of its argument, which gives the bit length. $\square$

9 Consequences of the order structure

9.1 Least required inventory

Corollary 9.1. *Let $t \geq 0$ and $\mathcal{S}_t \neq \emptyset$. The state $z_*(t)$ returned by Algorithm 1 or Algorithm 3 minimizes every coordinatewise nondecreasing function $C: \mathbb{R}^V \rightarrow \mathbb{R}$ over $\mathcal{S}_t$; in particular it minimizes every weighted inventory $\sum_v w_v z_v$ with $w_v \geq 0$.*

Proof. $z_*(t) \leq z$ coordinatewise for every $z \in \mathcal{S}_t$. $\square$

The statement concerns a fixed weak margin. The set of productive states is not closed and in general has no least element, and the rational witness of Lemma 5.2 spends part of the margin. Inventory here means activities of the internal species; nothing is asserted about food consumption or enzyme cost.

9.2 Capacity

For $E \neq \emptyset$ define the *capacity*

$$\delta_* = \max_{z \in \prod_v [\ell_v, u_v]} \min_{e: u \rightarrow v} \min\{z_v - G_e(z_u), F_e(z_u) - z_v\},$$

which exists by compactness and may be negative. Then $\mathcal{S}_t \neq \emptyset$ if and only if $t \leq \delta_*$, and a productive state exists if and only if $\delta_* > 0$.

Proposition 9.2 (Capacity scale). *With $\rho_e = a_e/b_e$ and $\kappa(\rho) = \rho/(8(\rho+1)(\rho+2))$,*

$$\delta_* \leq \min_{e: u \rightarrow v} \kappa(\rho_e) \max_{x \in [\ell_u, u_u]} 4x(1-x) \leq \min_e \kappa(\rho_e) \leq \kappa(\sqrt{2}) = \frac{3-2\sqrt{2}}{8} \approx 0.02145. \quad (18)$$

For a single core whose boxes contain $x = \frac{1}{2}$ and $y = \frac{1}{2}(G(\frac{1}{2}) + F(\frac{1}{2}))$, one has $\delta_ = \kappa(\rho)$ exactly; for unit factors $\delta_* = \frac{1}{48}$, attained at $(\frac{1}{2}, \frac{17}{48})$.*

Proof. The smaller of two numbers is at most half their sum, and the sum of the two normalized residuals of e is $F_e(z_u) - G_e(z_u)$, given by (6). Since $\rho/((\rho+1)(\rho+2)) = 1/(\rho+3+2/\rho)$ and $\rho+2/\rho \geq 2\sqrt{2}$, the maximum of κ is at $\rho = \sqrt{2}$; equivalently $(3-2\sqrt{2})(a+b)(a+2b) - ab = (3-2\sqrt{2})(a-\sqrt{2}b)^2$. For one core, choosing y at the midpoint of the band equalizes the two residuals at half the band width, and $x = \frac{1}{2}$ maximizes the width. $\square$

The two inequalities are compiled as `min_residual_le_capacity` and `ratio_factor_le`. The universal ceiling is approached but not attained by rational ratios. The root equation $x(1-x) = 12t$ of Example 3.5 is the capacity computation for the unit core: its discriminant $1-48t$ vanishes at $\delta_* = \frac{1}{48}$. Because $\mathcal{S}_t \neq \emptyset$ is decided for any rational $t \geq 0$ by Algorithm 1 with the local tests of Lemma 6.1(c), bisection in t brackets δ_* to any prescribed dyadic accuracy and returns a rational state whose margin is within that accuracy of optimal. Computing δ_* exactly, as an algebraic number, would require a decomposition of the margin axis at all critical values and is not done here.

9.3 Interval uncertainty in the ratios

Proposition 9.3 (Robust compatibility). *Let each ratio range independently over $[r_e^-, r_e^+] \subset (0, \infty)$. A state $x \in (0, 1]^V$ is productive for all realizations of the ratios if and only if*

$$G_{r_e^+}(x_u) < x_v < F_{r_e^-}(x_u) \quad \text{for every } e: u \rightarrow v, \quad (19)$$

where the subscript denotes the ratio. The width of this band is

$$F_{r^-}(x) - G_{r^+}(x) = \frac{(2r^- - r^+)x(1-x)}{(r^- + 1)(r^+ + 2)}, \quad (20)$$

so an edge with $r_e^+ \geq 2r_e^-$ makes robust compatibility impossible, and this threshold is sharp for a single core. For rational $r_e^\pm$, Theorems 1.1 and 1.2 hold for the system (19) with the same bounds.

Proof. On $[0, 1]$ both responses are nondecreasing in the ratio: $\partial_r F_r = x(1-x)/(r+1)^2$ and $\partial_r G_r = 2x(1-x)/(r+2)^2$. So the intersection of the bands over $r \in [r^-, r^+]$ is (19), and (20) is a direct computation. Sections 2–7 use, for each edge, only one increasing quadratic forward map of the form $G_r + t$ and one reverse map of the form $F_{r'}^{-1}(\cdot + t)$ with their analyticity, positivity at zero and derivative bounds; they never use that $r = r'$. $\square$

The monotonicity in the ratio and the obstruction are compiled as `lower_mono_ratio`, `upper_mono_ratio` and `robust_band_empty`. The quantifier order is $\exists x \forall r$: one state for all realizations, which is stronger than compatibility of every realization separately. Independent factor boxes $a_e \in [a_e^-, a_e^+]$, $b_e \in [b_e^-, b_e^+]$ give $r_e^- = a_e^-/b_e^+$ and $r_e^+ = a_e^+/b_e^-$ for positivity; guaranteed *magnitudes* of the residuals depend on the factors and not only on their ratio, and are treated at a fixed state in Section 10.

Remark 9.4 (Weighted demands). Replace the two margins of edge e in (7) by $t\omega_e^-$ and $t\omega_e^+$ with positive rational weights. The implication maps become $G_e + t\omega_e^-$ and $F_e^{-1}(\cdot + t\omega_e^+)$ and keep every property used above, so both algorithms and their bounds carry over with the weights included in B ; in Lemma 5.2 one takes $1/N \leq t \min_e \min(\omega_e^-, \omega_e^+)/4$. The choice $\omega_e^- = w_{e,u}/(a_e + 2b_e)$ and $\omega_e^+ = w_{e,v}/(a_e + b_e)$ expresses production demands $R_{e,u} \geq t w_{e,u}$ and $R_{e,v} \geq t w_{e,v}$ in the original flux units. Zero weights are not covered: they reintroduce identity cycles at positive t .

10 A certified module and an all-size family

10.1 The nominal certificate

Consider the shortcut triangle $A \rightarrow B$, $B \rightarrow C$, $A \rightarrow C$ with factors $(a, b) = (2, 1)$, $(2, 1)$ and $(1, 1)$, which is incompatible when all factors are equal to one (Section 2), and the rational state

$$(x_A, x_B, x_C) = \left(\frac{1}{10}, \frac{13}{200}, \frac{43}{1000} \right). \quad (21)$$

Table 2 lists the exact currents and residuals. All six residuals are positive; the smallest normalized residual is $\frac{209}{120000}$, on the target of BC . This is a certificate of compatibility, not an optimizer.

Core	p_e	q_e	$R_{e,u}$	$R_{e,v}$	min $R_{e,u}$ on the box	min $R_{e,v}$ on the box
AB	7/100	11/200	1/25	3/200	30351981/10 ⁹	16488021/(2 · 10 ⁹)
BC	11/250	1551/40000	671/20000	209/40000	26837781/10 ⁹	1175221/(2 · 10 ⁹)
AC	57/1000	33/1000	9/1000	3/125	2411981/10 ⁹	38368021/(2 · 10 ⁹)

Table 2: Exact currents and residuals of the module at the state (21), and the exact minima of the residuals over the uncertainty box of Section 10.2: every activity within $\pm 10^{-4}$ and every factor within $\pm 5\%$.

10.2 Tolerances

Factors. Let every factor vary independently by a relative amount at most r at the fixed state (21). Since $p_e, q_e > 0$, the worst cases are $R_{e,u} - r(2q_e + p_e)$ and $R_{e,v} - r(p_e + q_e)$. All cores stay productive exactly for

$$r < \min_e \min \left\{ \frac{R_{e,u}}{2q_e + p_e}, \frac{R_{e,v}}{p_e + q_e} \right\} = \frac{209}{3311} = \frac{19}{301} \approx 6.31\%,$$

and at $r = \frac{19}{301}$ one residual vanishes. At $r = 5\%$ the smallest residual is $\frac{869}{800000}$.

Factors and activities. Let, in addition, every activity vary independently within $\pm \varepsilon$, with $\varepsilon = 10^{-4}$ and $r = 5\%$. On these boxes $x - y > 0$ and $y - x^2 > 0$ for every core, $R_{e,u}$ decreases in x and in a and increases in y and in b , and $R_{e,v}$ does the opposite. The minima over the whole box are therefore attained at the corners

$$\begin{aligned} \min R_{e,u} &= 2b(1 - r)[y - \varepsilon - (x + \varepsilon)^2] - a(1 + r)[x - y + 2\varepsilon], \\ \min R_{e,v} &= a(1 - r)[x - y - 2\varepsilon] - b(1 + r)[y + \varepsilon - (x - \varepsilon)^2], \end{aligned}$$

listed in the last two columns of Table 2. Every state and every factor assignment in the box is productive, with residuals at least $\frac{1175221}{2 \cdot 10^9} \approx 5.876 \times 10^{-4}$. This is a statement about the continuum, proved by monotonicity; no sampling is involved.

10.3 Replication, food and scaling

In the windmill W_k of Proposition 8.1, give every module the factors above and the activities (21), with the hub shared. Every core sees the same pair of endpoint activities as in the triangle, so the certificate and both tolerance statements hold for every $k \geq 1$, on a family with $h = 4$. Food is consumed by the second channel only, at rate q_e per core, so a module consumes

$$q_{AB} + q_{BC} + q_{AC} = \frac{5071}{40000} \approx 0.1268$$

and contributes $R_{AB,A} + R_{AC,A} = \frac{49}{1000}$ to the net production of the hub. By the identity $R_{e,u} + R_{e,v} = q_e$ of Lemma 2.1, total production of internal species equals total food consumption, $k \cdot \frac{5071}{40000}$. Dividing all factors by k leaves the state compatible and keeps the aggregate food demand constant, but divides every individual residual by k . A polynomial-time certificate does not remove the linear resource cost of a growing assembly.

10.4 Units and scope of the physical model

A dimensional reading takes ideal activities $x_i = c_i/c_{\text{ref}}$, food concentration $c_F = c_{\text{ref}}$, and a flux scale J_{ref} . The first channel then has forward and reverse first-order constants $J_{\text{ref}}a/c_{\text{ref}}$, the second channel has forward and reverse second-order constants $J_{\text{ref}}b/c_{\text{ref}}^2$, and the dimensional currents are J_{ref} times (2). All displayed equilibrium constants equal one in these units; the constraints are of the loop-law type of Beard et al. [5]. For mass balance F and all X_s must share elemental composition and charge, for instance as isomers. With $c_{\text{ref}} = 1 \text{ mM}$ the state (21) is $(100, 65, 43) \mu\text{M}$ with uncertainty $\pm 0.1 \mu\text{M}$, and with the illustrative choice $J_{\text{ref}} = 1 \text{ mM min}^{-1}$ the certified worst residual is $0.5876 \mu\text{M min}^{-1}$. At a productive core $x^2 < y < x$, so the affinities $RT \log(x/y)$ and $RT \log(y/x^2)$ and both currents are positive, and the dissipation $p RT \log(x/y) + q RT \log(y/x^2)$ is positive. The quantity certified here is a *production residual*, which depends on kinetic factors; it differs from the minimal thermodynamic driving force optimized over concentration ranges by Noor et al. [26], although both are constrained pathway-design problems. These numbers specify a consistent normalized model. They are not measured parameters, and instantaneous productivity of a food-chemostatted network implies neither a steady state nor persistence.

11 Hypotheses used, and cores of higher order

The proofs use the following properties of the constraint system and nothing else about quadratics.

- (H1) Independent compact boxes, and constraints that are implications $z_j \geq f(z_i)$ with f increasing, so that feasible states dominate all path consequences, the feasible set is min-closed, and a variable can be eliminated exactly (Lemma 7.1).
- (H2) Compositions along walks that are real analytic near the compact seed intervals, so that non-identity cycle maps have finitely many fixed points and distinct walk maps cross finitely often. Any other source of this finiteness would serve equally.
- (H3) At positive margin, no identity cycles; here because every implication map is positive at 0.
- (H4) Transports whose graphs are described by polynomial equations and sign conditions of bounded degree with a *unique* admissible branch.
- (H5) A number of free seeds per comparison that is bounded (Theorem 1.1) or bounded in terms of h (Theorem 1.2).
- (H6) An explicit Lipschitz bound for the responses on the boxes, for the rational output.

Monotonicity and analyticity alone give termination of Algorithm 1 and correctness of Algorithm 3 over the reals, but no effective algebraic execution and no bit bound; conversely degree two is not essential. Rational rate laws of Michaelis–Menten type are *not* covered automatically: poles, range restrictions, non-unique inverse branches and couplings between more than two species have to be examined case by case. Constraints in three or more variables are in general not min-closed.

Theorem 11.1 (Cores of fixed order). *Fix an integer $d \geq 2$ and replace the second channel of every core by $X_v + (d-1)F \rightleftharpoons dX_u$, so that $q_e = b_e(x_v - x_u^d)$, $R_{e,u} = dq_e - p_e$ and $R_{e,v} = p_e - q_e$. Productivity of a core is equivalent to $G_d(x_u) < x_v < F_d(x_u)$ with*

$$G_d(x) = \frac{ax + dbx^d}{a + db}, \quad F_d(x) = \frac{ax + bx^d}{a + b}, \quad F_d(x) - G_d(x) = \frac{(d-1)abx(1 - x^{d-1})}{(a+b)(a+db)},$$

and Theorems 1.1 and 1.2 hold with constants C_d, c_d and a function f_d depending on d .

Proof. The band is a rearrangement, compiled as `order_d_band`. Both responses vanish at 0, fix 1, and have derivative at least $a/(a+db) > 0$ on $[0, \infty)$, so they are increasing bijections of $[0, \infty)$ with analytic inverses near every nonnegative point, by the analytic inverse function theorem [23]; this gives (H1)–(H3). The graph of a step is $(a+db)y - ax - dbx^d - (a+db)t = 0$, respectively $ay + by^d - (a+b)(x+t) = 0$, with $x, y \geq 0$ and a unique admissible y , which is (H4); (H5) is unchanged. In Lemma 6.1, (QE) is applied with degree d instead of 2, which changes Q to $(k+1)^{C_{dk}}$, and s_h in Lemma 7.3 depends on d . On $[0, 1]$ one has $G'_d \leq d$ and $F'_d \leq d$, so in Lemma 5.2 downward rounding with $1/N \leq t/(2d)$ keeps the margin $t/2$, which is (H6). $\square$

If d is part of the input, a separate analysis of degree and encoding is required. A single elementary step of molecularity d is chemically unrealistic for large d ; the theorem is stated to show that the mechanism is not tied to quadratics, and replacing the channel by a chain of bimolecular intermediates would require its own source-equivalence argument.

12 Formal verification

The Lean 4 development accompanying the paper compiles against Mathlib in a frozen environment (Lean v4.30.0, pinned Mathlib manifest), with warnings promoted to errors; it contains no `sorry`, no added axiom and no `native_decide`, and the reported axioms of the root declarations are `propext`, `Classical.choice` and `Quot.sound`. The sources are in the accompanying source repository, under `proofs/ThermoCoreCompatibility/`. Table 3 maps the claims of the paper to declarations in the namespace `ThermoCoreCompatibility.GeneralCompatibility`, except where another module is named.

The two root declarations state, for arbitrary finite index types, real factors and boxes, that if weak feasibility at margin σ is equivalent to weak feasibility at margin t for all $0 < \sigma \leq t$, then the existence of a boxed state at which every core is productive, in the literal sense of the source predicate `Productive`, is equivalent to the existence of a boxed state of margin t ; and that the absence of the latter excludes the former. The stability hypothesis is what Theorem 5.1 provides for $t = t_1$, and Corollary 5.3 for $t = t_0$, by conventional real-algebraic arguments. The elimination lemma is compiled for arbitrary index types, an arbitrary set U with a next-map, and arbitrary monotone outgoing maps; it uses no continuity and no algebra. The declarations do not construct decompositions, do not traverse graphs and do not establish any complexity statement, and the development is not an end-to-end verified decision procedure. A small rational witness checker accompanies the sources; it verifies supplied witnesses only.

13 Discussion and open questions

The structural content of the paper is twofold. First, a finite representation theorem: the least state of a system of monotone, analytic, two-variable implications with unique algebraic branches is assembled from box endpoints and roots of *fixed* simple-cycle equations, each transported along one simple path, uniformly for small positive margins. Second, a closure property: such systems are closed under exact elimination of a variable, the disconnected

Claim in the paper	Compiled declarations	Conventional part
Lemma 2.1: band, residual identities	MultiInterface.WeightedPair: productive_iff, residual_lower, residual_upper, gap; RobustExample.residual_sum	none
Implications (9), explicit inverse	inverseUpper, upper_inverseUpper, inverseUpper_le_iff, inverseUpper_monotoneOn, continuous_inverseUpper, boxed_margin_iff_implications	analyticity of compositions
Lemma 2.2: least state, tight support	compact_min_closed_has_least, margin_min, margin_isClosed, margin_has_least, active_support_coverage	none
Lemma 3.2: forced root	lifted_cycle_violation, fixed_between_violation_and_prefixed, least_cycle_root_preserves_bound, no_cycle_root_rejects_prefixed	extraction of the cycle from the label path (Lemma 3.1)
Lemmas 3.3, 4.1: finiteness	finite_cycle_roots_of_violation, finite_cycle_roots, consumed_tokens_injective, cycle_jump_count_bound, no_infinite_anchor_updates	identification of tokens with graph cycles
Remark 3.6(a)	Interaction: support_least, full_least, selected_support_is_tight_at_global, selected_least_not_global	six-species instance (exact computation)
Theorem 5.1, Corollary 5.3	feasible_of_candidate_transport, boxed_strict_iff_source, strict_iff_positive_margin, margin_downward, strict_decision_of_stable_margin, source_decision_at_stable_threshold, no_productive_source_of_no_threshold_state	Lemmas 4.2, 4.3: elimination, CAD, branches; the stability hypothesis
Lemma 5.2	round_down_margin; BeyondJunction.MarginTransfer	bisection, output size
Lemma 7.1, Proposition 7.2	IsNextMap, IsNextMap.mono, IsNextMap.fixed_of_mem, eliminate_vertex_iff, eliminated_value_least, generated_map_mono	closedness of U_w , induction over the forest
Corollary 9.1	least_minimizes_monotone, least_minimizes_inventory	none
Proposition 9.2	min_residual_le_capacity, ratio_factor_le	attainment for one core
Proposition 9.3	lower_mono_ratio, upper_mono_ratio, robust_band_empty	reduction of the algorithms
Section 10	triangle_nominal, triangle_box_corners_positive	monotonicity reduction to corners
Theorem 11.1	order_d_band	analytic and algebraic part
Theorems 1.1, 1.2: bit bounds; Lemma 7.3; Section 8	none	entirely conventional

Table 3: Claim-to-proof concordance. “Compiled” means that a strict compilation receipt exists for the current source hash of the module.

projections being recorded by next-maps, and along a forest of bounded depth the description grows additively. The algorithms are stated for the two-reaction cores of [8, 9, 22], where they turn a problem that was decided only for forests and for boundedly many junctions into one

that is fixed-parameter tractable in the longest path on all graphs.

Question 13.1. Is there an algorithm with cost $2^{\text{poly}(h)} \text{poly}(n, B)$? Theorem 1.2 has an iterated exponential f , caused by walks of length 2^{h+1} , while Theorem 1.1 has $(h+2)^{O(h)}$ arithmetic but the factor J .

Question 13.2. Is the number J of jumps of Algorithm 1 bounded by $\varphi(h) \text{poly}(n)$ on all graphs, for a suitable rule for choosing the detected cycle? The token bound counts all roots of all simple cycles, which is $n^{\Theta(q)}$ on $K_{q,n}$. There, eliminating the n leaves by Lemma 7.1 leaves a system on q hubs whose maps are envelopes with near-linearly many pieces, and a jump to a fixed point of the *attaining piece* either lands on a fixed point of the envelope cycle map or leaves the piece for good, which bounds the jumps by $\varphi(q) n^{1+o(1)}$; we do not know a rule that achieves this on general graphs without computing envelopes.

Question 13.3. Can the sign of $F_m \circ \dots \circ F_1(x_0) - \lambda$ in Proposition 1.3 be decided in time polynomial in m and B ? A positive answer is necessary for a polynomial-time algorithm on all graphs; a hardness result for this special sign problem would be of independent interest.

Question 13.4. Can the capacity δ_* be computed exactly, as an algebraic number with a certified optimizer, within the bounds of Theorems 1.1 or 1.2? This asks for the critical margins at which branches of Lemma 4.3 are born, merge or exchange test results, as in Figure 2(c).

Question 13.5. Which reversible mass-action motifs beyond (1) and Theorem 11.1 satisfy (H1)–(H6)? Cores with three or more internal species give constraints in more than two variables, for which min-closure can fail.

Practical matters are also open. A complete implementation needs certified interval filters for the overwhelming majority of sign decisions and the projected exact fallback of Lemma 6.1 for the rest; in an exploratory prototype of cycle acceleration, exact comparison of nested radicals, not the number of jumps, was the bottleneck, which is the difficulty that bounded ancestry and retained seeds are designed to remove. Treating the margin as an infinitesimal, rather than fixing an a priori threshold, keeps the working precision tied to the comparisons actually made. No runtime claim is made.

A Reproducibility

The source package contains `main.tex`, `refs.bib`, the figure script `make_figures.py`, the numerical sanity check `sanity_elimination.py` described in Section 7, and `check_paper.py`, which replays with exact rational and symbolic arithmetic every number and identity printed in the paper: the identities of Lemma 2.1 and the explicit inverse; Example 3.5, including $\alpha = (5 - \sqrt{13})/10$ and the tight least state; the three examples of Remark 3.6; the longest paths of stars, complete graphs and windmills and the inequality $m \leq hn$ on them; an instance of Proposition 1.3 with its bit growth; Proposition 9.2, including the maximizer $\rho = \sqrt{2}$ and the unit optimum; the identities of Proposition 9.3; Table 2, the radius $\frac{19}{301}$, the value $\frac{869}{800000}$, the food and hub totals; and the identities and derivative bounds of Theorem 11.1 for $d = 2, \dots, 6$. It requires Python with SymPy and ends with the line `"status": "PASS"`. The script is a certificate replay and not an implementation of the algorithms. The core of the certificate for Section 10 is short enough to state:

```
from fractions import Fraction as Q
edges = [('AB', Q(1,10), Q(13,200), Q(2), Q(1)),
         ('BC', Q(13,200), Q(43,1000), Q(2), Q(1)),
         ('AC', Q(1,10), Q(43,1000), Q(1), Q(1))]
r, eps = Q(1,20), Q(1,10000)
radii, worst, food = [], [], Q(0)
```

```

for name, x, y, a, b in edges:
    p, q = a*(x-y), b*(y-x*x)
    ru, rv = 2*q-p, p-q
    assert ru > 0 and rv > 0
    radii += [ru/(2*q+p), rv/(p+q)]
    assert x-eps > y+eps and y-eps > (x+eps)**2
    worst += [2*b*(1-r)*(y-eps-(x+eps)**2) - a*(1+r)*(x-y+2*eps),
              a*(1-r)*(x-y-2*eps) - b*(1+r)*(y+eps-(x-eps)**2)]
    food += q
assert min(radii) == Q(19,301) and food == Q(5071,40000)
assert min(worst) == Q(1175221,2000000000) > 0

```

The Lean modules named in Table 3 are in the subdirectories `GeneralCompatibility`, `BeyondJunction` and `MultiInterface` of `proofs/ThermoCoreCompatibility/`, together with their strict compilation receipts; the archive `lean_sources.zip` of the source package contains them with the toolchain pin and a short guide.

References

- [1] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009. doi:10.1137/070697926.
- [2] B. Aspvall and Y. Shiloach. A polynomial time algorithm for solving systems of linear inequalities with two variables per inequality. *SIAM Journal on Computing*, 9(4):827–845, 1980. doi:10.1137/0209063.
- [3] S. Basu. Algorithms in real algebraic geometry: a survey. arXiv preprint arXiv:1409.1534 [math.AG], 2014. arXiv:1409.1534. Theorem 2.27 is the statement used here.
- [4] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer, Berlin, second edition, 2006. doi:10.1007/3-540-33099-2.
- [5] D. A. Beard, E. Babson, E. Curtis, and H. Qian. Thermodynamic constraints for biochemical networks. *Journal of Theoretical Biology*, 228(3):327–333, 2004. doi:10.1016/j.jtbi.2004.01.008.
- [6] A. Blokhuis, D. Lacoste, and P. Nghe. Universal motifs and the diversity of autocatalytic systems. *Proceedings of the National Academy of Sciences*, 117(41):25230–25236, 2020. doi:10.1073/pnas.2013527117.
- [7] G. E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 134–183. Springer, 1975. doi:10.1007/3-540-07407-4_17.
- [8] Companion manuscript. Exact common-activity compatibility for autocatalytic junction–path assemblies: boundary reduction, a rational decision algorithm, and operating certificates, 2026. Unpublished; Lean development in the same source repository as the present paper.
- [9] Companion manuscript. Collective thermodynamic incompatibility of autocatalytic cores: minimal conflicts of unbounded order, bounded fans, and graded assemblies, 2026. Unpublished; Lean development in the same source repository as the present paper.

- [10] Companion manuscript. Thermodynamic compatibility of interacting autocatalytic cores: an exact phase diagram and a layered obstruction theory, 2026. Unpublished; Lean development in the same source repository as the present paper.
- [11] M. C. Cooper, P. G. Jeavons, and A. Z. Salamon. Generalizing constraint satisfaction on trees: hybrid tractability and variable elimination. *Artificial Intelligence*, 174(9–10): 570–584, 2010. doi:10.1016/j.artint.2010.03.002.
- [12] A. Costan, S. Gaubert, E. Goubault, M. Martel, and S. Putot. A policy iteration algorithm for computing fixed points in static analysis of programs. In *Computer Aided Verification (CAV 2005)*, volume 3576 of *Lecture Notes in Computer Science*, pages 462–475. Springer, 2005. doi:10.1007/11513988_46.
- [13] D. Dadush, Z. K. Koh, B. Natura, and L. A. Végh. An accelerated Newton–Dinkelbach method and its application to two variables per inequality systems. arXiv preprint arXiv:2004.08634 [cs.DS], 2020. arXiv:2004.08634.
- [14] J. H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5(1–2):29–35, 1988. doi:10.1016/S0747-7171(88)80004-X.
- [15] L. de Moura and S. 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.
- [16] J. Esparza, S. Kiefer, and M. Luttenberger. Newtonian program analysis. *Journal of the ACM*, 57(6):1–47, 2010. doi:10.1145/1857914.1857917.
- [17] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. *Journal of the ACM*, 56(1):1–66, 2009. doi:10.1145/1462153.1462154.
- [18] T. Gawlitza and H. Seidl. Precise fixpoint computation through strategy iteration. In *Programming Languages and Systems (ESOP 2007)*, volume 4421 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2007. doi:10.1007/978-3-540-71316-6_21.
- [19] D. Y. Grigor’ev and N. N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *Journal of Symbolic Computation*, 5(1–2):37–64, 1988. doi:10.1016/S0747-7171(88)80005-1.
- [20] D. S. Hochbaum and J. Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM Journal on Computing*, 23(6):1179–1192, 1994. doi:10.1137/S0097539793251876.
- [21] P. G. Jeavons and M. C. Cooper. Tractable constraints on ordered domains. *Artificial Intelligence*, 79(2):327–339, 1995. doi:10.1016/0004-3702(95)00107-7.
- [22] T. Kosci, D. Kuperberg, E. Rajon, and S. Charlat. Thermodynamic consistency of autocatalytic cycles. *Proceedings of the National Academy of Sciences*, 122(18):e2421274122, 2025. doi:10.1073/pnas.2421274122.
- [23] S. G. Krantz and H. R. Parks. *A Primer of Real Analytic Functions*. Birkhäuser, Boston, second edition, 2002. doi:10.1007/978-0-8176-8134-0.
- [24] M. Mignotte. *Mathematics for Computer Algebra*. Springer, New York, 1992. doi:10.1007/978-1-4613-9171-5.

- [25] J. Nešetřil and P. Ossona de Mendez. *Sparsity: Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and Combinatorics*. Springer, 2012. doi:[10.1007/978-3-642-27875-4](https://doi.org/10.1007/978-3-642-27875-4).
- [26] E. Noor, A. Bar-Even, A. Flamholz, E. Reznik, W. Liebermeister, and R. Milo. Pathway thermodynamics highlights kinetic obstacles in central metabolism. *PLoS Computational Biology*, 10(2):e1003483, 2014. doi:[10.1371/journal.pcbi.1003483](https://doi.org/10.1371/journal.pcbi.1003483).
- [27] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. Parts I–III. *Journal of Symbolic Computation*, 13(3):255–352, 1992. doi:[10.1016/S0747-7171\(10\)80003-3](https://doi.org/10.1016/S0747-7171(10)80003-3).
- [28] M. Sagraloff and K. Mehlhorn. Computing real roots of real polynomials. *Journal of Symbolic Computation*, 73:46–86, 2016. doi:[10.1016/j.jsc.2015.03.004](https://doi.org/10.1016/j.jsc.2015.03.004).
- [29] M. Sharir and P. K. Agarwal. *Davenport–Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [30] R. Shostak. Deciding linear inequalities by computing loop residues. *Journal of the ACM*, 28(4):769–779, 1981. doi:[10.1145/322276.322288](https://doi.org/10.1145/322276.322288).
- [31] E. Szemerédi. On a problem of Davenport and Schinzel. *Acta Arithmetica*, 25(2):213–224, 1974. doi:[10.4064/aa-25-2-213-224](https://doi.org/10.4064/aa-25-2-213-224).
- [32] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkeley, second edition, 1951.
- [33] 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).