Succinct Betti-2 Positivity and Its Complexity Cauchy-Based Evaluation, $\oplus\mathbf{P}$-Completeness on a Promise Family, Deterministic SAT Reductions, and Unconditional Circuit Lower Bounds on Explicit Subfamilies

Alwin
University of Indonesia

Abstract

We study the decision problem (Π_{β_2}): given a succinct (local-oracle) description (D) of a bounded-degree simplicial (2)-complex (K_D), decide whether the second Betti number ($\beta_2(K_D;\mathbb{F}_2)$) is positive. The central theme is the interaction between (i) topological invariants of low-dimensional complexes and (ii) computational complexity under succinct representations, with implications for the landscape surrounding (P) versus (NP).

We present two complementary reduction pipelines. The first pipeline defines a Cauchy-based evaluation problem (SCE-Dec) over ($\mathbb{F}_{2^k}$), proves that SCE-Dec can be expressed as an ($\mathbb{F}_2$)-linear form in evaluator bits, and encodes this linear form as a bounded-occurrence ($\mathbb{F}_2$)-linear system. We then convert such systems into CW (2)-complexes and further into bounded-degree simplicial (2)-complexes via a star-triangulation procedure. This yields a mapping (ProbeBit) from SCE-Dec instances to local-oracle descriptions (D) such that ($\beta_2(K_D) > 0$) if and only if the target SCE-Dec bit is (1). As complexity consequences, we obtain ($\oplus\mathbf{P}$)-completeness of (Π_{β_2}) restricted to the promise family (Im(ProbeBit)), and a one-sided randomized many-one reduction ($\mathrm{SAT} \leq_{rp} \Pi_{\beta_2}$) on the same promise family using the Valiant–Vazirani isolation lemma (standard result; citation placeholder). We also prove an unconditional deterministic evaluation-local query lower bound ($q \geq N$) for computing SCE-Dec.

The second pipeline gives a deterministic "witness-expansion" construction that maps a Boolean formula (ϕ) to a disjoint union (K_ϕ) of constant-size sphere/disk gadgets indexed by assignments. This yields a parsimonious equality ($\beta_2(K_\phi;\mathbb{F}_2) = \#\mathrm{SAT}(\phi)$), hence deterministic ($\mathrm{SAT} \leq_m \Pi_{\beta_2}$) and ($\#\mathrm{SAT} \leq_m$ Compute-β_2) (function hardness). Finally, we provide unconditional nonuniform lower bounds against ($\mathbf{AC}^0$), De Morgan formulas, and ($\mathbf{AC}^0[p]$) on explicit parity-based subfamilies via projection reductions from (PARITY) (standard results; citation placeholders).

We explicitly separate validated statements from steps requiring either standard citations or remaining formalization (oracle-interface encoding details and a triangulated-disk homeomorphism lemma). No unconditional separation such as ($P \neq NP$) is claimed.

1 Introduction

1.1 Motivation and context

The question (P) versus (NP) sits at the center of theoretical computer science. One broad strategy toward understanding the landscape around (P) and (NP) is to analyze natural computational problems that arise outside of "synthetic" encodings, and to determine how their structural content (algebraic, geometric, topological) interacts with algorithmic and lower-bound methods. Topological invariants, and Betti numbers in particular, provide canonical examples: they are intrinsic, stable under homeomorphisms, and encode global structure in a way that is often difficult to infer from local information.

This paper concerns the complexity of deciding whether the second Betti number (β_2) of a simplicial (2)-complex is positive, when the complex is given succinctly. The decision problem we study is:

Given a local-oracle (succinct) description (D) of a bounded-degree simplicial (2)-complex (K_D), decide whether ($\beta_2(K_D;\mathbb{F}_2) > 0$).

The bounded-degree condition is essential: it keeps local neighborhoods constant-size, enabling local-oracle access to faces, edges, and incident structure. However, even under bounded degree, a succinctly represented complex can have exponentially many simplices relative to the input length, making it nontrivial to relate "local query access" to global invariants such as (β_2).

While the main topic is not a proof of ($P \neq NP$), the results are relevant to (P) versus (NP) in three ways:

1. Hardness and completeness phenomena for (Π_{β_2}) (both under promise restrictions and in unrestricted deterministic NP-hardness via explicit gadget families) illustrate how global topological invariants naturally encode counting and satisfiability.

2. Query lower bounds in oracle models demonstrate information-theoretic barriers for evaluation tasks intimately related to the reductions.

3. Unconditional circuit lower bounds on explicit subfamilies give a clean, "white-box" separation of (Π_{β_2}) from low-depth circuit classes on carefully chosen inputs, complementing the more algebraic and topological reductions.

1.2 Two reduction pipelines

We develop two distinct pipelines, each with different strengths and technical requirements.

Pipeline A (Cauchy/SCE → linear systems → topology: the ProbeBit route). We define a succinct evaluation problem SCE-Dec based on multiplying a vector ($x(X) \in \mathbb{F}_{2^k}^N$), generated by an evaluator circuit (X), by an explicit Cauchy matrix ($C_N \in \mathbb{F}_{2^k}^{N\times N}$). The target is a single output bit of the form

$$\mathrm{SCE}_{N,i,t}(X) = \pi_t\big((C_N x(X))_i\big) \in \mathbb{F}_2.$$

We show that $\mathrm{SCE}_{N,i,t}(X)$ is an ($\mathbb{F}_2$)-linear form in the evaluator output bits ($x_{j,\ell}$) with coefficients ($m_{j,\ell}^{(N,i,t)} \in \mathbb{F}_2$). We encode this linear form as a bounded-occurrence ($\mathbb{F}_2$)-linear system $\mathsf{SysBit}(N,i,t,b,X)$ such that the system has a nonzero solution if and only if $\mathrm{SCE}_{N,i,t}(X) = b$. Next, we build a CW (2)-complex $K^{\mathrm{cw}}(\mathsf{Sys})$ whose second homology (H_2) is isomorphic to the solution space $\mathsf{Sol}(\mathsf{Sys})$. Finally, we convert the CW complex into a bounded-degree simplicial (2)-complex $K(\mathsf{Sys})$ by triangulating each (2)-cell via a star-triangulation; in the intended form, this triangulation preserves homology. The composition yields a mapping (ProbeBit) from SCE-Dec instances to local-oracle descriptions of bounded-degree simplicial (2)-complexes such that

$$\beta_2(K_{\mathrm{ProbeBit}(N,i,t,b,X)};\mathbb{F}_2) > 0 \iff \mathrm{SCE}_{N,i,t}(X) = b.$$

Pipeline B (Witness-expansion route: deterministic SAT and #SAT). Independently, we define a deterministic construction mapping a Boolean formula (ϕ) to a disjoint union

$$K_\phi = \bigsqcup_{x\in\{0,1\}^n} \mathcal{C}_x,$$

where each component $\mathcal{C}_x$ is a constant-size simplicial (2)-complex that is a "sphere gadget" if $\phi(x) = 1$ and a "disk gadget" if $\phi(x) = 0$. Over $(\mathbb{F}_2)$, the sphere gadget has $(\beta_2 = 1)$ and the disk gadget has $(\beta_2 = 0)$. By additivity of homology over disjoint unions, we obtain

$$\beta_2(K_\phi; \mathbb{F}_2) = \#\mathrm{SAT}(\phi),$$

which immediately implies deterministic $(\mathrm{SAT} \leq_m \Pi_{\beta_2})$ and parsimonious $(\#\mathrm{SAT} \leq_m \text{Compute-}\beta_2)$.

1.3 Main results and careful scope statements

We state high-level theorems here and provide full proofs in Parts 2–5. Statements that rely on standard external results are explicitly labeled. Statements whose proofs require appendix-level formalization are labeled with "Proof deferred" and the location.

Theorem 1.1 (ProbeBit correctness; proof in Parts 2–4; one topological lemma in Appendix A)**.** *There exists an explicit mapping* (ProbeBit) *taking an SCE-Dec instance* $((N, i, t, b, X))$ *to a local-oracle description* (D) *of a bounded-degree simplicial (2)-complex* (K_D) *such that*

$$\beta_2(K_D; \mathbb{F}_2) > 0 \iff \mathrm{SCE}_{N,i,t}(X) = b.$$

Proof status: the algebraic, gadget, and CW-homology steps are proved in Parts 2–3; the simplicialization and homology-preservation step uses a triangulated-disk homeomorphism lemma. Proof deferred to Appendix A, where we either (i) give a full proof, or (ii) invoke a standard PL-topology result with citation placeholder.

Theorem 1.2 ($(\oplus\mathbf{P})$-completeness on a promise family; proof in Part 4)**.** *Let* $\mathcal{I}_{\mathrm{Probe}} := \mathrm{Im}(\mathrm{ProbeBit})$ *be the promise family of local-oracle descriptions produced by* (ProbeBit)*. Then* (Π_{β_2}) *restricted to* $\mathcal{I}_{\mathrm{Probe}}$ *is* $(\oplus\mathbf{P})$*-complete under deterministic many-one reductions.*

Scope note: This is a promise statement: hardness and membership are shown only for inputs guaranteed to lie in $\mathcal{I}_{\mathrm{Probe}}$*.*

Theorem 1.3 (One-sided randomized SAT reduction on the ProbeBit promise family; standard external lemma)**.** *There is a one-sided randomized many-one reduction*

$$\mathrm{SAT} \leq_{rp} \Pi_{\beta_2} \restriction_{\mathcal{I}_{\mathrm{Probe}}}.$$

Proof uses: Valiant–Vazirani isolation lemma (standard result; citation placeholder).

Scope note: Again a promise statement.

Theorem 1.4 (Evaluation-local query lower bound; unconditional; proof in Part 4)**.** *Fix* (N, i, t)*. Any deterministic evaluation-local algorithm that computes* $\mathrm{SCE}_{N,i,t}(X)$ *for all evaluator circuits* (X) *must query* $(X(j))$ *for at least* (N) *distinct indices* $(j \in [N])$ *in the worst case.*

Theorem 1.5 (Witness-expansion parsimonious equality; deterministic; proof in Part 5)**.** *There exists a deterministic mapping* $(\phi \mapsto D_\phi)$ *producing a local-oracle description* (D_ϕ) *of a bounded-degree simplicial (2)-complex* (K_{D_ϕ}) *such that*

$$\beta_2(K_{D_\phi}; \mathbb{F}_2) = \#\mathrm{SAT}(\phi).$$

Consequently, $(\mathrm{SAT} \leq_m \Pi_{\beta_2})$ *and* $(\#\mathrm{SAT} \leq_m \text{Compute-}\beta_2)$*. Proof status: topological facts about the constant-size gadgets are proved directly (finite chain computation), and the disjoint-union additivity is proved in Part 5.*

Theorem 1.6 (Unconditional circuit lower bounds on explicit subfamilies; proof in Part 5; standard external results)**.** *There exists an explicit subfamily* ($\mathcal{F} \subseteq$) *instances of* (Π_{β_2}) *such that* $\Pi_{\beta_2} \restriction_{\mathcal{F}}$ *computes* (PARITY) *under a projection reduction. Hence:*

- ($\Pi_{\beta_2} \notin \mathbf{AC}^0$) *(nonuniform),*
- (Π_{β_2}) *requires quadratic-size De Morgan formulas on* ($\mathcal{F}$)*, and*
- ($\Pi_{\beta_2} \notin \mathbf{AC}^0[p]$) *for odd primes* ($p$)*,*

using standard results about (PARITY) *(citation placeholders).*

Important clarification: These are unconditional lower bounds, but they are proved via an explicit hard subfamily ($\mathcal{F}$)*. The implication to the global language is by restriction: if the full language were in a circuit class, then its restriction to* ($\mathcal{F}$) *would be as well.*

1.4 What this paper does not claim

- We do not claim an unconditional separation such as ($P \neq NP$).
- Several results are explicitly formulated as promise results (notably ($\oplus\mathbf{P}$)-completeness and ($\text{SAT} \leq_{rp}$) on (Im(ProbeBit))).
- We keep a strict separation between:
  - deterministic NP-hardness via witness-expansion (unrestricted), and
  - promise-family completeness via ProbeBit.

1.5 Organization of the paper (across five parts)

- Part 1 (this part): definitions, models, and problem statements.
- Part 2: SCE-Dec definition; linearization into XOR masks; bounded-occurrence gadgets; construction of SysBit.
- Part 3: CW encoding of linear systems; theorem ($H_2 \cong \mathsf{Sol}$).
- Part 4: simplicialization and homology preservation; ProbeBit mapping; ($\oplus\mathbf{P}$)-completeness on promise family; SAT ($\leq_{rp}$); evaluation-local lower bound.
- Part 5: witness-expansion (deterministic SAT and #SAT); unconditional circuit lower bounds on explicit parity-based subfamilies; discussion, limitations, appendices, and bibliography placeholders.

2 Notation and Preliminaries

2.1 Basic notation and conventions

- For ($m \in \mathbb{N}$), we write
$$[m] := \{0, 1, \dots, m-1\},$$
i.e., we use 0-indexing throughout.

- All vector spaces are over ($\mathbb{F}_2$) unless explicitly stated otherwise. We write XOR as ($\oplus$). For vectors ($u, v \in \mathbb{F}_2^d$), ($u \oplus v$) denotes coordinate-wise XOR.
- For a finite set (S), we use ($\mathbb{F}_2^S$) to denote the ($\mathbb{F}_2$)-vector space of formal ($\mathbb{F}_2$)-linear combinations of elements of (S), equivalently functions ($S \to \mathbb{F}_2$).
- Unless stated otherwise, "polynomial time" and "polynomial size" are with respect to the length of the explicit input bitstring (e.g., a Boolean formula (ϕ) or an oracle/circuit description (D)).
- Degree bounds as constants. When we say "bounded degree" for a simplicial complex in the local-oracle model, the bound (Δ) is treated as a fixed absolute constant that is part of the model, not part of the input length.

2.2 Finite fields of characteristic two

We require arithmetic in ($\mathbb{F}_{2^k}$) to define the Cauchy-based evaluation problem. To avoid symbol collisions, we reserve ($\mathbb{K}$) for the finite field and (K) for a simplicial complex.

Definition 2.1 (Finite field model ($\mathbb{K} = \mathbb{F}_{2^k}$) via a polynomial basis)**.** *Fix ($k \in \mathbb{N}$). Let ($p_k(z) \in \mathbb{F}_2[z]$) be an irreducible polynomial of degree (k). Define*

$$\mathbb{K} := \mathbb{F}_{2^k} \cong \mathbb{F}_2[z]/(p_k(z)).$$

Let $\alpha := z \bmod p_k(z)$. We use the basis

$$\mathcal{B} := \{1, \alpha, \alpha^2, \dots, \alpha^{k-1}\}$$

to identify ($\mathbb{K}$) with ($\mathbb{F}_2^k$) as a vector space.

Every element ($u \in \mathbb{K}$) has a unique coordinate representation

$$u = \sum_{\ell=0}^{k-1} u_\ell \alpha^\ell, \quad u_\ell \in \mathbb{F}_2.$$

Definition 2.2 (Coordinate projections)**.** *For each ($t \in [k]$), define $\pi_t : \mathbb{K} \to \mathbb{F}_2$ by*

$$\pi_t\left(\sum_{\ell=0}^{k-1} u_\ell \alpha^\ell\right) := u_t.$$

Definition 2.3 (Bitstring encoding map)**.** *Given a bitstring ($z = (z_0, \dots, z_{k-1}) \in \mathbb{F}_2^k$), define*

$$\mathrm{enc}(z) := \sum_{\ell=0}^{k-1} z_\ell \alpha^\ell \in \mathbb{K}.$$

When convenient, we view integers ($r \in [2^k]$) as bitstrings ($z \in \mathbb{F}_2^k$) via binary expansion and then as field elements via enc*.*

Remark 2.4 (Boolean circuit realizability of basic field operations)**.** *Addition in ($\mathbb{K}$) is coordinate-wise XOR under ($\mathcal{B}$). Multiplication can be implemented by polynomial multiplication followed by reduction modulo (p_k). Inversion can be implemented by an extended Euclidean algorithm in ($\mathbb{F}_2[z]$) modulo (p_k).*

Proof status: The concrete circuit bounds (size poly(k)*) are standard and will be provided as a short self-contained proof in Appendix B, or cited as standard finite-field arithmetic (citation placeholder). This point is needed only to justify "succinctness" and time bounds in later sections.*

2.3 Simplicial (2)-complexes and ($\mathbb{F}_2$)-homology

We work with simplicial complexes of dimension at most (2) and with coefficients in ($\mathbb{F}_2$).

Definition 2.5 (Simplicial (2)-complex). *A simplicial (2)-complex is a triple* ($K = (V, E, T)$) *where*

- (V) *is a finite set of vertices,*
- ($E \subseteq \binom{V}{2}$) *is a set of (unordered) edges,*
- ($T \subseteq \binom{V}{3}$) *is a set of (unordered) triangles,*

such that the closure condition holds: if $\{u, v, w\} \in T$*, then* $\{u, v\}, \{u, w\}, \{v, w\} \in E$*. We assume* ($K$) *has no simplices of dimension* ≥ 3*.*

Definition 2.6 (Chain groups and boundary maps over ($\mathbb{F}_2$)). *Let* ($K = (V, E, T)$) *be a simplicial (2)-complex. Define chain groups*

$$C_2(K) := \mathbb{F}_2^T, \quad C_1(K) := \mathbb{F}_2^E, \quad C_0(K) := \mathbb{F}_2^V.$$

Define boundary maps $\partial_2 : C_2 \to C_1$ *and* $\partial_1 : C_1 \to C_0$ *on basis simplices by*

$$\partial_2(\{u, v, w\}) := \{u, v\} \oplus \{u, w\} \oplus \{v, w\},$$

$$\partial_1(\{u, v\}) := u \oplus v,$$

and extend ($\mathbb{F}_2$)*-linearly. Since* (K) *has no (3)-simplices,* $C_3(K) = 0$ *and* $\operatorname{im}(\partial_3) = \{0\}$*.*

Definition 2.7 (Second homology and second Betti number). *The second homology group over* ($\mathbb{F}_2$) *is*

$$H_2(K; \mathbb{F}_2) := \ker(\partial_2) \subseteq C_2(K),$$

and the second Betti number is

$$\beta_2(K; \mathbb{F}_2) := \dim_{\mathbb{F}_2} H_2(K; \mathbb{F}_2).$$

Remark 2.8 (Why ($\mathbb{F}_2$) coefficients). *All constructions in this paper are tailored to* ($\mathbb{F}_2$)*: (i) linear systems are over* ($\mathbb{F}_2$)*, (ii) parity encodings are natural, and (iii) the simplicial/cellular boundary computations are most transparent. Some gadget-based arguments extend to other fields, but we restrict to* ($\mathbb{F}_2$) *for consistency and to minimize ancillary technicalities.*

2.4 Bounded degree and local-oracle descriptions

The central object of study is a simplicial (2)-complex given succinctly via a local-oracle description.

Definition 2.9 (Bounded-degree simplicial (2)-complex). *Let* ($\Delta \in \mathbb{N}$)*. A simplicial (2)-complex* ($K = (V, E, T)$) *is* (Δ)*-bounded-degree if every vertex* ($v \in V$) *is incident to at most* (Δ) *edges and at most* (Δ) *triangles.*

We treat (Δ) *as a fixed constant throughout.*

Definition 2.10 (Local-oracle description of a bounded-degree simplicial (2)-complex). *A local-oracle description* (D) *of a* (Δ)*-bounded-degree simplicial (2)-complex consists of:*

1. *Three nonnegative integers* (n_V, n_E, n_T) *(in binary), intended as sizes of* (V, E, T)*.*

2. *Boolean circuits implementing the following functions:*

   - *Edge endpoints oracle*
   $$\mathrm{End}_D : [n_E] \to [n_V] \times [n_V],$$
   where $\mathrm{End}_D(e) = (u, v)$ *returns the endpoints of edge* (e)*.*
   - *Triangle vertices oracle*
   $$\mathrm{Vert}_D : [n_T] \to [n_V] \times [n_V] \times [n_V],$$
   where $\mathrm{Vert}_D(\tau) = (u, v, w)$ *returns the vertices of triangle* (τ)*.*
   - *Incidence listing oracles*
   $$\mathrm{IncE}_D : [n_V] \times [\Delta] \to [n_E] \cup \{\perp\}, \qquad \mathrm{IncT}_D : [n_V] \times [\Delta] \to [n_T] \cup \{\perp\},$$
   where $\mathrm{IncE}_D(v, \ell)$ *returns the* ℓ*-th edge incident to* (v) *(or* $\perp$*), and* $\mathrm{IncT}_D(v, \ell)$ *returns the* ℓ*-th triangle incident to* (v) *(or* $\perp$*).*

3. *A promise that these circuits describe a valid* (Δ)*-bounded-degree simplicial (2)-complex* $(K_D = (V_D, E_D, T_D))$ *with* $(V_D = [n_V])$*, edges* (E_D) *induced by* End_D*, triangles* (T_D) *induced by* Vert_D*, satisfying:*

   - *closure (every triangle's edges exist in* (E_D)*),*
   - *consistency with incidence lists,*
   - *bounded-degree by* (Δ)*.*

Remark 2.11 (Why incidence lists are included)**.** *Without* IncE_D *and* IncT_D*, recovering all incident edges/triangles around a vertex from* End_D *and* Vert_D *alone may require* $\Omega(n_E)$ *or* $\Omega(n_T)$ *work. The bounded-degree promise allows constant-length incidence lists, making "local access" genuinely local.*

Definition 2.12 (Size and query time in the local-oracle model)**.** *The description size* $(|D|)$ *is the total bit-length of the encoding of* (n_V, n_E, n_T) *plus the encodings of the four circuits* $\mathrm{End}_D, \mathrm{Vert}_D, \mathrm{IncE}_D, \mathrm{IncT}_D$*. A query to* (D) *is one evaluation of one oracle circuit on a valid input. We say* (D) *is efficiently queriable if each oracle evaluation runs in time* $\mathrm{poly}(|D|)$ *on a standard RAM model.*

Additional formalization needed (encoding detail). *In later sections (notably circuit lower bounds via projections), we will require an explicit bit-level encoding of Boolean circuits so that statements such as "the mapping* (Φ) *is a projection/*$\mathbf{AC}^0$*" are meaningful. We defer a fixed encoding convention to Appendix C. Until that point, we treat* $(|D|)$ *abstractly as "bit-length of the circuit descriptions."*

2.5 Complexity classes and reductions used later

We recall the basic complexity notions used in Parts 4–5.

Definition 2.13 $((\oplus\mathbf{P}))$**.** *A language* $(L \subseteq \{0,1\}^*)$ *is in* $(\oplus\mathbf{P})$ *if there exists a nondeterministic polynomial-time Turing machine* (M) *such that for all* (x)*,*
$$x \in L \iff \#\mathrm{acc}_M(x) \equiv 1 \pmod 2,$$
where $\#\mathrm{acc}_M(x)$ *is the number of accepting computation paths of* (M) *on input* (x)*.*

Definition 2.14 $((\oplus\mathrm{SAT}))$**.** $(\oplus\mathrm{SAT})$ *is the decision problem: given a Boolean formula* (ϕ)*, decide whether* $\#\mathrm{SAT}(\phi)$ *is odd.*

Standard result (citation placeholder). ($\oplus$SAT) is ($\oplus$**P**)-complete under deterministic many-one reductions.

Definition 2.15 (Deterministic many-one reduction)**.** *For languages* ($A, B \subseteq \{0,1\}^*$)*, we write* ($A \leq_m B$) *if there is a polynomial-time computable function* (f) *such that* ($x \in A \iff f(x) \in B$)*.*

Definition 2.16 (One-sided randomized many-one reduction)**.** *We write* ($A \leq_{rp} B$) *if there exists a randomized polynomial-time function* (f) *and a polynomial* (p) *such that:*

- *If* ($x \in A$)*, then* $\Pr[f(x) \in B] \geq 1/p(|x|)$*;*
- *If* ($x \notin A$)*, then* $\Pr[f(x) \in B] = 0$*.*

Definition 2.17 (Promise problems and restriction to a family)**.** *Given a language/problem* (B) *and a subset* ($\mathcal{I}$) *of inputs, we write* ($B\restriction_{\mathcal{I}}$) *for the restriction of* (B) *to inputs in* ($\mathcal{I}$)*. When we say a hardness/completeness statement holds "on* ($\mathcal{I}$)*," we mean it is a promise statement: inputs are guaranteed to lie in* ($\mathcal{I}$)*.*

3 Computational Problems Studied

3.1 Betti-2 positivity and related function problems

Definition 3.1 (Betti-2 positivity problem (Π_{β_2}))**.** *The decision problem* (Π_{β_2}) *takes as input a local-oracle description* (D) *promised to describe a valid* (Δ)*-bounded-degree simplicial (2)-complex* (K_D)*, and outputs*

$$\Pi_{\beta_2}(D) := \begin{cases} 1 & \text{if } \beta_2(K_D; \mathbb{F}_2) > 0, \\ 0 & \text{if } \beta_2(K_D; \mathbb{F}_2) = 0. \end{cases}$$

Definition 3.2 (Compute-(β_2) as a function problem)**.** *The function problem* Compute-β_2 *takes as input* (D) *and outputs the integer* $\beta_2(K_D; \mathbb{F}_2)$ *(in binary).*

Remark 3.3 (Decision vs function hardness)**.** *A parsimonious reduction* ($\#\mathrm{SAT}(\phi) \mapsto \beta_2(K_\phi)$) *yields* ($\#$**P**)*-hardness for* Compute-β_2 *and NP-hardness for* (Π_{β_2})*, but does not imply membership of* (Π_{β_2}) *in* (NP)*. In the succinct setting, membership questions can be nontrivial because witnesses (e.g., a nonzero (2)-cycle) might be exponentially large in the description length.*

3.2 A succinct Cauchy-based evaluation problem (SCE-Dec)

To connect (Π_{β_2}) to parity counting, we introduce an auxiliary problem based on multiplication by a Cauchy matrix over ($\mathbb{F}_{2^k}$). Full details appear in Part 2; we state the definition now since it motivates subsequent reductions.

We will set $k := \lceil \log_2(4N) \rceil$, ensuring $2^k \geq 4N$. We define an explicit ($N \times N$) Cauchy matrix (C_N) over ($\mathbb{K} = \mathbb{F}_{2^k}$) using two disjoint sets of field elements $\{a_0, \ldots, a_{N-1}\}$ and $\{b_0, \ldots, b_{N-1}\}$ (precise construction in Part 2).

An evaluator (X) is a Boolean circuit that, on input ($j \in [N]$), outputs (k) bits interpreted as an element ($x_j(X) \in \mathbb{K}$). This defines a vector ($x(X) \in \mathbb{K}^N$).

Definition 3.4 (SCE bit)**.** *Given integers* ($N \geq 1$)*,* ($i \in [N]$)*,* ($t \in [k]$)*, and an evaluator* (X)*, define*

$$\mathrm{SCE}_{N,i,t}(X) := \pi_t\big((C_N x(X))_i\big) \in \mathbb{F}_2.$$

Definition 3.5 (SCE decision language SCE-Dec)**.** *Define*

$$\text{SCE-Dec} := \{(N, i, t, b, X) : \text{SCE}_{N,i,t}(X) = b\},$$

where $b \in \mathbb{F}_2$*.*

Remark 3.6 (Why SCE-Dec is a natural intermediary)**.** *The Cauchy matrix gives a dense linear operator, and the output bit* $\text{SCE}_{N,i,t}(X)$ *can be expressed as a parity (XOR) of evaluator output bits with explicit coefficients. This parity structure is the bridge that allows encoding into bounded-occurrence* $(\mathbb{F}_2)$*-linear systems, and then into* (H_2) *of a (2)-complex.*

3.3 Promise families and scope of hardness statements

Two distinct families of instances arise:

1. ProbeBit family $(\mathcal{I}_{\text{Probe}})$: instances (D) that are guaranteed to be of the form $(D = \text{ProbeBit}(N, i, t, b, X))$. Many parity-counting completeness results are established on this family (Part 4).
2. Witness-expansion family $(\mathcal{I}_{\text{WE}})$: instances (D_ϕ) produced by the deterministic witness-expansion construction mapping formulas (ϕ) to complexes (K_ϕ). Deterministic $(\text{SAT} \leq_m \Pi_{\beta_2})$ is shown on this family (Part 5), and because the map is deterministic and polynomial-time, this yields NP-hardness for the unrestricted decision problem (Π_{β_2}).

We will keep these families separate. In particular:

- $(\oplus\mathbf{P})$-completeness is proved for $(\Pi_{\beta_2} \restriction_{\mathcal{I}_{\text{Probe}}})$, not for all inputs.
- Deterministic NP-hardness is proved via witness-expansion, without using ProbeBit's promise restriction.

4 Background and Related Work (brief)

This section is intentionally concise and uses citation placeholders only, as required.

- Succinct representations. Many graph and combinatorial problems become substantially harder under succinct encodings (e.g., circuit-encoded adjacency predicates). Our local-oracle model for bounded-degree simplicial complexes is an instance of this general paradigm.
- Homology computation. Computing Betti numbers is a classical problem in computational topology. Complexity varies sharply with dimension, coefficient field, and representation format (explicit vs succinct). We focus on bounded-degree simplicial (2)-complexes under local-oracle access, and on the specific decision predicate $(\beta_2 > 0)$ over $(\mathbb{F}_2)$.
- Parity and counting classes. The class $(\oplus\mathbf{P})$ and $(\oplus\text{SAT})$ are standard objects in complexity theory (citation placeholder), with $(\oplus\text{SAT})$ being $(\oplus\mathbf{P})$-complete.
- Valiant–Vazirani isolation. The Valiant–Vazirani lemma (citation placeholder) provides a randomized reduction from SAT to UniqueSAT. In this paper it is used to obtain a one-sided randomized reduction from SAT to (Π_{β_2}) on a promise family.
- Circuit lower bounds. Lower bounds for (PARITY) against $(\mathbf{AC}^0)$, De Morgan formulas, and $(\mathbf{AC}^0[p])$ are classical (citation placeholders). We use these results in a "white-box" manner by exhibiting explicit parity-encoding subfamilies of (Π_{β_2}).

5 Status Markers and Deferred Technical Items

To comply with strict formal-audit standards, we list items that will require either a complete proof in later parts or an explicit citation placeholder.

5.1 Proof-deferred items (will be proved inside this paper)

- Appendix A: A topological lemma about triangulated disks being homeomorphic to (D^2) rel boundary, used to justify that triangulating CW (2)-cells preserves geometric realization/homology.
- Appendix C: A fixed bit-level encoding of oracle circuits, needed to make "projection/$\mathbf{AC}^0$-computable mapping" statements formal.

5.2 Standard external results (citation placeholders required)

- Valiant–Vazirani isolation lemma.
- ($\oplus$SAT) is ($\oplus\mathbf{P}$)-complete.
- (PARITY $\notin \mathbf{AC}^0$); quadratic formula lower bounds; (PARITY $\notin \mathbf{AC}^0[p]$) for odd primes.

5.3 Remaining "formalization risk" points (to be resolved in Parts 4–5)

- Oracle-interface completeness for witness-expansion: mapping the witness-expansion oracles to the specific interface $(\mathrm{End}_D, \mathrm{Vert}_D, \mathrm{IncE}_D, \mathrm{IncT}_D)$ must be presented explicitly. The construction is straightforward but must be written carefully.
- Succinctness of ProbeBit: the local decoding procedures for the ProbeBit-generated complex require explicit indexing conventions for variables, equations, and boundary-walk access.

These are not intended to introduce new mathematical claims; they are required to satisfy the "defined-before-used and proof-complete" standard.

Field-Linearity, Bit-Linear Forms for SCE, and the Bounded-Occurrence System SysBit

---

6 Field-Linearity and Bit-Linear Forms for SCE

This section gives the algebraic core underlying the later topological encodings: a target output bit of a succinct Cauchy evaluation can be written as an explicit $\mathbb{F}_2$-linear form in evaluator output bits. We also record a basic "nonzero-row" property that will later support an information-theoretic lower bound.

Throughout, we use the finite-field representation from Section 2.2: for each k, we fix an irreducible polynomial $p_k \in \mathbb{F}_2[z]$ of degree k, set $\mathbb{K} = \mathbb{F}_2[z]/(p_k)$, write $\alpha = z \bmod p_k$, and use the basis $\mathcal{B} = \{1, \alpha, \ldots, \alpha^{k-1}\}$.

6.1 Cauchy matrix over $\mathbb{F}_{2^k}$

We now define the explicit Cauchy matrix used in the Succinct Cauchy Evaluation (SCE) problem.

Definition 6.1 (Field size for parameter N, evaluation points, and Cauchy matrix)**.** *Fix an integer $N \geq 2$. Set*

$$k \;:=\; \lceil \log_2(4N) \rceil, \qquad \mathbb{K} \;:=\; \mathbb{F}_{2^k},$$

represented using the polynomial-basis model from Definition 2.1.

Define the field elements

$$a_i \;:=\; \mathrm{enc}(i) \in \mathbb{K} \quad \textit{for } i \in [N], \qquad b_j \;:=\; \mathrm{enc}(N+j) \in \mathbb{K} \quad \textit{for } j \in [N],$$

where $\mathrm{enc}(\cdot)$ *is the bitstring-to-field encoding from Definition 2.3 (treating integers in* $[2^k]$ *as their k-bit expansions).*

Define the Cauchy matrix

$$C_N \in \mathbb{K}^{N \times N} \quad \textit{by} \quad (C_N)_{i,j} \;:=\; \frac{1}{a_i - b_j}.$$

(Over characteristic two, subtraction equals addition; we retain $a_i - b_j$ for readability.)

Lemma 6.2 (Disjointness of $\{a_i\}$ and $\{b_j\}$)**.** *For all $i, j \in [N]$, $a_i \neq b_j$. Consequently, $(C_N)_{i,j}$ is well-defined and nonzero.*

Proof. By definition $k = \lceil \log_2(4N) \rceil$, hence $2^k \geq 4N > 2N$. Therefore all integers in $\{0, 1, \ldots, 2N-1\}$ are distinct elements of $[2^k]$. The encoding $\mathrm{enc} : [2^k] \to \mathbb{K}$ is injective by construction (Definition 2.3). Thus $\mathrm{enc}(i) \neq \mathrm{enc}(N+j)$ for all $i, j \in [N]$, i.e., $a_i \neq b_j$. Hence $a_i - b_j \neq 0$ in $\mathbb{K}$, so the inverse exists and is nonzero. □

6.2 Evaluators and the SCE target bit

We formalize the "evaluator" notion used informally in Part 1.

Definition 6.3 (Evaluator and implicit vector $x(X)$)**.** *Fix $N \geq 2$ and $k = \lceil \log_2(4N) \rceil$. An evaluator is a Boolean circuit*

$$X : [N] \to \{0,1\}^k.$$

For each $j \in [N]$, write

$$X(j) \;=\; (x_{j,0}, x_{j,1}, \ldots, x_{j,k-1}) \in \mathbb{F}_2^k,$$

and interpret it as a field element

$$x_j(X) \;:=\; \sum_{\ell=0}^{k-1} x_{j,\ell}\alpha^\ell \in \mathbb{K}.$$

This defines an implicit vector $x(X) = (x_0(X), \ldots, x_{N-1}(X)) \in \mathbb{K}^N$.

Recall from Definition 3.4 that for indices $i \in [N]$ and $t \in [k]$,

$$\mathrm{SCE}_{N,i,t}(X) \;:=\; \pi_t\big((C_N x(X))_i\big) \in \mathbb{F}_2.$$

6.3 Multiplication by a fixed field element is $\mathbb{F}_2$-linear

The key point is that, under a fixed basis, multiplication by a fixed $\kappa \in \mathbb{K}$ is an $\mathbb{F}_2$-linear operator and hence corresponds to a $k \times k$ matrix over $\mathbb{F}_2$.

Definition 6.4 (Multiplication operator)**.** *For $\kappa \in \mathbb{K}$, define*

$$L_\kappa : \mathbb{K} \to \mathbb{K}, \qquad L_\kappa(u) = \kappa u.$$

Lemma 6.5 ($\mathbb{F}_2$-linearity and the coordinate matrix $M(\kappa)$)**.** *For every $\kappa \in \mathbb{K}$, the map L_κ is $\mathbb{F}_2$-linear. Consequently, there exists a unique matrix*

$$M(\kappa) \in \mathbb{F}_2^{k\times k}$$

such that for every $u = \sum_{\ell=0}^{k-1} u_\ell \alpha^\ell \in \mathbb{K}$,

$$\kappa u \;=\; \sum_{t=0}^{k-1} \left(\bigoplus_{\ell=0}^{k-1} M(\kappa)_{t,\ell}\, u_\ell \right) \alpha^t.$$

Proof. Linearity: for $u, v \in \mathbb{K}$ and $c \in \mathbb{F}_2 = \{0,1\}$, $L_\kappa(u \oplus v) = \kappa(u+v) = \kappa u + \kappa v = L_\kappa(u) \oplus L_\kappa(v)$, and $L_\kappa(cu) = c(\kappa u)$ holds because c is central and $c \in \{0,1\}$. Thus L_κ is $\mathbb{F}_2$-linear.

Existence/uniqueness of $M(\kappa)$: the basis $\mathcal{B}$ identifies $\mathbb{K}$ with $\mathbb{F}_2^k$ as vector spaces. Any $\mathbb{F}_2$-linear map $\mathbb{K} \to \mathbb{K}$ is determined uniquely by its values on the basis vectors α^ℓ for $\ell \in [k]$, which become the columns of the matrix $M(\kappa)$. □

Lemma 6.6 (Invertibility for $\kappa \neq 0$)**.** *If $\kappa \in \mathbb{K} \setminus \{0\}$, then $M(\kappa)$ is invertible as a matrix over $\mathbb{F}_2$. In particular, every row and every column of $M(\kappa)$ is nonzero.*

Proof. If $\kappa \neq 0$, multiplication by κ is a bijection $\mathbb{K} \to \mathbb{K}$ (since $\mathbb{K}$ is a field). Therefore the $\mathbb{F}_2$-linear map L_κ is invertible, hence its matrix representation $M(\kappa)$ is invertible over $\mathbb{F}_2$.

If a row of $M(\kappa)$ were all zeros, then the corresponding output coordinate $\pi_t(\kappa u)$ would be identically zero for all u, implying the image of L_κ lies in a proper $(k-1)$-dimensional subspace of $\mathbb{K}$, contradicting bijectivity. Similarly, a zero column would mean some nonzero basis vector maps to 0, contradicting injectivity. □

6.4 The SCE output bit is an explicit XOR of evaluator bits

Fix $N \geq 2$, indices $i \in [N]$ and $t \in [k]$. We now define explicit $\mathbb{F}_2$ mask coefficients that represent $\mathrm{SCE}_{N,i,t}(X)$ as an XOR of evaluator bits.

Definition 6.7 (Coefficient masks for a fixed output bit)**.** *Fix $N \geq 2$, $i \in [N]$, and $t \in [k]$. For each $j \in [N]$, set*

$$\kappa_{i,j} \;:=\; (C_N)_{i,j} \in \mathbb{K}, \qquad M_{i,j} \;:=\; M(\kappa_{i,j}) \in \mathbb{F}_2^{k\times k}.$$

Define the mask coefficients

$$m_{j,\ell}^{(N,i,t)} \;:=\; (M_{i,j})_{t,\ell} \in \mathbb{F}_2 \quad \text{for } j \in [N],\ \ell \in [k].$$

When N, i, t are clear, we write $m_{j,\ell}$ for $m_{j,\ell}^{(N,i,t)}$.

Lemma 6.8 (Bit-linear form for SCE)**.** *For every evaluator X,*

$$\mathrm{SCE}_{N,i,t}(X) \;=\; \bigoplus_{j\in[N]} \bigoplus_{\ell\in[k]} \Big(m_{j,\ell}^{(N,i,t)} \cdot x_{j,\ell}\Big),$$

where $x_{j,\ell} \in \mathbb{F}_2$ is the ℓ-th output bit of $X(j)$, and the product is the usual multiplication in $\mathbb{F}_2$ (so it either keeps or removes the bit).

Proof. By definition,

$$(C_N x(X))_i = \sum_{j\in[N]} (C_N)_{i,j}\, x_j(X) = \sum_{j\in[N]} \kappa_{i,j}\, x_j(X),$$

where the sum is in $\mathbb{K}$. Write $x_j(X) = \sum_{\ell=0}^{k-1} x_{j,\ell}\alpha^{\ell}$. By Lemma 6.5, the t-th coordinate of $\kappa_{i,j}x_j(X)$ equals

$$\pi_t(\kappa_{i,j}x_j(X)) = \bigoplus_{\ell\in[k]} (M(\kappa_{i,j}))_{t,\ell}\, x_{j,\ell} = \bigoplus_{\ell\in[k]} m_{j,\ell}^{(N,i,t)}\, x_{j,\ell}.$$

Taking π_t of the full sum and using that π_t is $\mathbb{F}_2$-linear (it is a coordinate projection) yields

$$\mathrm{SCE}_{N,i,t}(X) = \pi_t((C_N x(X))_i) = \bigoplus_{j\in[N]} \bigoplus_{\ell\in[k]} m_{j,\ell}^{(N,i,t)}\, x_{j,\ell}.$$

□

Lemma 6.9 (Computability of a single mask bit)**.** *Fix the field representation from Definition 2.1, and fix $N \geq 2$. Given indices $i, j \in [N]$ and $t, \ell \in [k]$, the coefficient bit $m_{j,\ell}^{(N,i,t)} \in \mathbb{F}_2$ can be computed in time polynomial in k.*

Proof. We describe an explicit procedure.

1. Compute $a_i = \mathrm{enc}(i)$ and $b_j = \mathrm{enc}(N+j)$ as k-bit coordinate vectors in the basis $\mathcal{B}$. This is direct from Definition 2.3.
2. Compute $d = a_i - b_j \in \mathbb{K}$. In characteristic two this is coordinate-wise XOR.
3. By Lemma 6.2, $d \neq 0$, hence invertible. Compute $d^{-1} \in \mathbb{K}$ using standard polynomial arithmetic in $\mathbb{F}_2[z]/(p_k(z))$; for example, compute the multiplicative inverse of $d(z)$ modulo $p_k(z)$ using the extended Euclidean algorithm for polynomials. This runs in time polynomial in k.
4. Set $\kappa_{i,j} = d^{-1} \in \mathbb{K}$.
5. To compute the matrix entry $M(\kappa_{i,j})_{t,\ell}$, compute the product $\kappa_{i,j} \cdot \alpha^{\ell} \in \mathbb{K}$ in the polynomial representation, reduce modulo p_k, and read off its t-th coordinate. By Lemma 6.5, the coordinate vector of $\kappa_{i,j}\alpha^{\ell}$ is exactly the ℓ-th column of $M(\kappa_{i,j})$. Thus the desired entry is its t-th bit.

Each step is polynomial in k, so the overall procedure is polynomial in k. □

Remark 6.10 (Nonzero rows of mask matrices)**.** *Since $\kappa_{i,j} = (C_N)_{i,j} \neq 0$, Lemma 6.6 implies that for each fixed $t \in [k]$ and each $j \in [N]$, there exists at least one $\ell \in [k]$ with $m_{j,\ell}^{(N,i,t)} = 1$. This will later be used in an information-theoretic lower bound for a restricted oracle model.*

7 Homogeneous $\mathbb{F}_2$-Linear Systems and Bounded Gadgets

We now introduce the intermediate combinatorial object used throughout the remainder of the paper: bounded-arity, bounded-occurrence homogeneous linear systems over $\mathbb{F}_2$. We then develop two reusable gadgets—EqTree and XorTree—that allow us to copy a single bit to many locations and to aggregate large XORs while maintaining constant arity and occurrence.

7.1 Systems, arity, and occurrence

Definition 7.1 (Homogeneous $\mathbb{F}_2$-linear system)**.** *A homogeneous $\mathbb{F}_2$-linear system is a pair*

$$\mathsf{Sys} = (\mathcal{V}, \mathcal{E}),$$

where $\mathcal{V}$ is a finite set of Boolean variables (taking values in $\mathbb{F}_2$), and $\mathcal{E}$ is a finite set of linear equations over $\mathbb{F}_2$. Each equation $e \in \mathcal{E}$ is specified by a subset $S_e \subseteq \mathcal{V}$ and has the form

$$\bigoplus_{v \in S_e} v \;=\; 0.$$

A solution is an assignment $\sigma : \mathcal{V} \to \mathbb{F}_2$ satisfying all equations. A solution is nonzero if it is not identically zero, i.e., $\exists v \in \mathcal{V}$ with $\sigma(v) = 1$.

Definition 7.2 (Arity and occurrence)**.** *Let $\mathsf{Sys} = (\mathcal{V}, \mathcal{E})$.*

- *The arity of an equation $e \in \mathcal{E}$ is $|S_e|$.*
- *The occurrence of a variable $v \in \mathcal{V}$ is the number of equations $e \in \mathcal{E}$ for which $v \in S_e$.*

We say Sys has bounded arity if every equation has arity at most some absolute constant. We say Sys has bounded occurrence if every variable occurs in at most some absolute constant number of equations.

7.2 Equality gadget

Definition 7.3 (Equality constraint)**.** *For variables $u, v \in \mathcal{V}$, the equation*

$$u \oplus v = 0$$

is called an equality constraint, since it enforces $u = v$.

We will use equality constraints organized in a tree to copy one root value to many leaves.

Definition 7.4 (Equality tree gadget: EqTree)**.** *Let $L \geq 1$. Let r be a distinguished variable (the root), and let $y_0, \ldots, y_{L-1}$ be designated leaf variables. Choose a rooted binary tree $\mathcal{T}$ whose root is r and whose leaves are exactly $y_0, \ldots, y_{L-1}$. Introduce fresh variables for the internal nodes of $\mathcal{T}$ (if any). For every edge (p, c) (parent p, child c) in $\mathcal{T}$, add the equality constraint*

$$p \oplus c = 0.$$

The resulting set of constraints is denoted

$$\mathrm{EqTree}(r; y_0, \ldots, y_{L-1}).$$

Lemma 7.5 (Correctness of EqTree)**.** *In any solution of* $\mathrm{EqTree}(r; y_0, \ldots, y_{L-1})$*, every node in the tree (in particular, each leaf y_j) equals r.*

Proof. Each edge constraint $p \oplus c = 0$ forces $p = c$. By induction on distance from the root, every node must equal the root value. □

7.3 XOR-sum gadget

Definition 7.6 (XOR-sum tree gadget: XorTree)**.** *Let $L \geq 1$. Let S be a designated root variable, and let $z_0, \dots, z_{L-1}$ be designated leaf variables. Choose a rooted binary tree $\mathcal{T}$ whose root is S and whose leaves are exactly $z_0, \dots, z_{L-1}$. Introduce fresh variables for the internal nodes of $\mathcal{T}$ (if any). For every internal node s with children u and v, add the equation*

$$s \oplus u \oplus v = 0.$$

Denote the resulting system by

$$\mathrm{XorTree}(S;\, z_0, \dots, z_{L-1}).$$

Lemma 7.7 (Subtree XOR invariant for XorTree)**.** *Let $\mathcal{T}$ be the rooted binary tree underlying* $\mathrm{XorTree}(S;\, z_0, \dots, z_{L-1})$*. For any node s in $\mathcal{T}$, let* $\mathrm{Leaves}(s)$ *denote the set of leaf variables in the subtree rooted at s. In any solution σ of* XorTree*,*

$$\sigma(s) \;=\; \bigoplus_{z \in \mathrm{Leaves}(s)} \sigma(z).$$

In particular,

$$\sigma(S) = \bigoplus_{j=0}^{L-1} \sigma(z_j).$$

Proof. Proceed by induction on the height of node s. If s is a leaf z_j, then $\mathrm{Leaves}(s) = \{z_j\}$ and the identity is trivial.

If s is an internal node with children u, v, the constraint $s \oplus u \oplus v = 0$ gives $\sigma(s) = \sigma(u) \oplus \sigma(v)$. By the induction hypothesis,

$$\sigma(u) = \bigoplus_{z \in \mathrm{Leaves}(u)} \sigma(z), \qquad \sigma(v) = \bigoplus_{z \in \mathrm{Leaves}(v)} \sigma(z).$$

Since $\mathrm{Leaves}(s) = \mathrm{Leaves}(u) \cup \mathrm{Leaves}(v)$ (disjoint union), we obtain the desired identity. The special case $s = S$ yields the root XOR formula. □

7.4 Boundedness properties of the gadgets

Lemma 7.8 (Bounded arity and bounded occurrence of the gadgets)**.** *Fix $L \geq 1$.*

1. *Every equation in* $\mathrm{EqTree}(r;\, y_0, \dots, y_{L-1})$ *has arity 2.*

2. *Every equation in* $\mathrm{XorTree}(S;\, z_0, \dots, z_{L-1})$ *has arity 3 (except for the degenerate case $L = 1$, where one may omit the gadget and set $S = z_0$).*

3. *If the rooted tree in* EqTree *is chosen to have maximum degree 3 (i.e., a rooted binary tree), then each variable in the gadget occurs in at most 3 equations.*

4. *In* XorTree*, each leaf z_j appears in exactly one equation (as a child of its parent), each internal node appears in exactly two equations (its own defining equation and its parent's equation), and the root S appears in exactly one equation. Hence every variable occurrence is at most 2 within the gadget.*

Proof. (1) and (2) follow directly from the definitions.

For (3), in a rooted binary tree each node has at most one parent edge and at most two child edges, so each node variable appears in at most $1 + 2 = 3$ constraints.

For (4), each internal node s participates in its own defining equation and (unless it is the root) in its parent's equation; leaves participate only as children, hence exactly once. □

---

8 The System SysBit: Construction and Correctness

Fix $N \geq 2$, indices $i \in [N]$, $t \in [k]$, a target bit $b \in \mathbb{F}_2$, and an evaluator X. This section defines a homogeneous $\mathbb{F}_2$-linear system

$$\mathsf{SysBit}(N, i, t, b, X)$$

that has a nonzero solution if and only if $\mathrm{SCE}_{N,i,t}(X) = b$.

Throughout this section, let $m_{j,\ell} = m_{j,\ell}^{(N,i,t)}$ be the mask coefficients from Definition 6.7, and let $x_{j,\ell}$ be the evaluator bits from Definition 6.3.

8.1 Intuition: a "switch" and a forced XOR check

We want to enforce

$$S \;=\; \bigoplus_{j\in[N]} \; \bigoplus_{\ell\in[k]} m_{j,\ell}\, x_{j,\ell},$$

and then check that $S = b$. Since our system must be homogeneous, we implement the check as

$$S \oplus (b \cdot \lambda_{\mathrm{root}}) = 0,$$

where λ_{root} is a "switch" variable. The switch serves two purposes:

- It makes the system always satisfiable (the all-zero assignment).
- It ensures that nonzero satisfiability corresponds to the intended check: any nonzero solution must force $\lambda_{\mathrm{root}} = 1$, and then the check becomes $S = b$.

8.2 Formal definition of SysBit

Definition 8.1 (The system $\mathsf{SysBit}(N, i, t, b, X)$)**.** *Define* $\mathsf{SysBit}(N, i, t, b, X) = (\mathcal{V}, \mathcal{E})$ *as follows.*

Variables. *Include the following designated variables:*

- *A root switch variable* λ_{root}.
- *Copy variables* λ_j *for each* $j \in [N]$.
- *Copy variables* $\lambda_{j,\ell}$ *for each* $(j, \ell) \in [N] \times [k]$.
- *Gating variables* $u_{j,\ell}$ *and* $z_{j,\ell}$ *for each* $(j, \ell) \in [N] \times [k]$.
- *Aggregation variables* w_j *for each* $j \in [N]$.
- *A final sum variable* S.

In addition, include all internal variables introduced by the EqTree *and* XorTree *gadgets instantiated below.*

Equations. *Include the following constraints:*

(E1) Copy λ_{root} to λ_j: include

$$\text{EqTree}(\lambda_{\text{root}}; \lambda_0, \dots, \lambda_{N-1}).$$

(E2) Copy λ_j to $\lambda_{j,\ell}$: for each $j \in [N]$, include

$$\text{EqTree}(\lambda_j; \lambda_{j,0}, \dots, \lambda_{j,k-1}).$$

(E3) Gate by evaluator bits $x_{j,\ell}$: for each $(j,\ell) \in [N] \times [k]$, include:

- *if $x_{j,\ell} = 0$, the unary equation $u_{j,\ell} = 0$;*
- *if $x_{j,\ell} = 1$, the binary equation $u_{j,\ell} \oplus \lambda_{j,\ell} = 0$.*

(E4) Gate by mask bits $m_{j,\ell}$: for each $(j,\ell) \in [N] \times [k]$, include:

- *if $m_{j,\ell} = 0$, the unary equation $z_{j,\ell} = 0$;*
- *if $m_{j,\ell} = 1$, the binary equation $z_{j,\ell} \oplus u_{j,\ell} = 0$.*

(E5) Sum within each j: for each $j \in [N]$, include

$$\text{XorTree}(w_j;\ z_{j,0}, \dots, z_{j,k-1}).$$

(E6) Sum across j: include

$$\text{XorTree}(S;\ w_0, \dots, w_{N-1}).$$

(E7) Homogeneous check against b:

- *if $b = 0$, include the unary equation $S = 0$;*
- *if $b = 1$, include the binary equation $S \oplus \lambda_{\text{root}} = 0$.*

This completes the definition of $\mathsf{SysBit}(N, i, t, b, X)$. (Each equation is of the form XOR-of-variables = 0, hence the system is homogeneous, and the all-zero assignment always satisfies all equations.)

8.3 The switch property

Lemma 8.2 (Switch lemma)**.** *If σ is a solution of $\mathsf{SysBit}(N, i, t, b, X)$ with $\sigma(\lambda_{\text{root}}) = 0$, then σ is the all-zero assignment. In particular, every nonzero solution must satisfy $\sigma(\lambda_{\text{root}}) = 1$.*

Proof. Assume $\sigma(\lambda_{\text{root}}) = 0$.

1. By (E1) and Lemma 7.5, $\sigma(\lambda_j) = \sigma(\lambda_{\text{root}}) = 0$ for all $j \in [N]$, and all internal variables of this EqTree gadget are 0.

2. Fix any j. By (E2) and Lemma 7.5, $\sigma(\lambda_{j,\ell}) = \sigma(\lambda_j) = 0$ for all $\ell \in [k]$, and the internal EqTree variables are 0.

3. Consider (E3). If $x_{j,\ell} = 0$, then $u_{j,\ell} = 0$. If $x_{j,\ell} = 1$, then the constraint $u_{j,\ell} \oplus \lambda_{j,\ell} = 0$ forces $u_{j,\ell} = \lambda_{j,\ell} = 0$. Thus $\sigma(u_{j,\ell}) = 0$ for all (j,ℓ).

4. Consider (E4). If $m_{j,\ell} = 0$, then $z_{j,\ell} = 0$. If $m_{j,\ell} = 1$, then $z_{j,\ell} \oplus u_{j,\ell} = 0$ forces $z_{j,\ell} = u_{j,\ell} = 0$. Hence $\sigma(z_{j,\ell}) = 0$ for all (j,ℓ).

5. Now (E5) says each w_j is the XOR of the $z_{j,\ell}$ (Lemma 7.7), so $\sigma(w_j) = 0$ for all j, and all internal XorTree nodes are 0.

6. Finally, (E6) implies S is the XOR of the w_j, so $\sigma(S) = 0$, and again internal nodes in this XorTree are 0.

7. The check (E7) is then satisfied automatically: if $b = 0$, it requires $S = 0$; if $b = 1$, it requires $S \oplus \lambda_{\text{root}} = 0$, which holds since both are 0.

Thus every designated variable is 0, and all gadget-internal variables are 0. Hence σ is the all-zero assignment. The final sentence follows immediately. □

8.4 The forced XOR value when $\lambda_{\text{root}} = 1$

Lemma 8.3 (Forced sum lemma)**.** *Let σ be a solution of $\mathsf{SysBit}(N, i, t, b, X)$ with $\sigma(\lambda_{\text{root}}) = 1$. Then*

$$\sigma(S) \;=\; \bigoplus_{j\in[N]} \bigoplus_{\ell\in[k]} (m_{j,\ell} \cdot x_{j,\ell}).$$

Proof. Assume $\sigma(\lambda_{\text{root}}) = 1$.

1. By (E1) and Lemma 7.5, $\sigma(\lambda_j) = 1$ for all j. By (E2) and Lemma 7.5, $\sigma(\lambda_{j,\ell}) = 1$ for all j, ℓ.

2. Consider (E3). If $x_{j,\ell} = 0$, then $u_{j,\ell} = 0$. If $x_{j,\ell} = 1$, then $u_{j,\ell} \oplus \lambda_{j,\ell} = 0$ forces $u_{j,\ell} = 1$. Therefore,
$$\sigma(u_{j,\ell}) = x_{j,\ell}.$$

3. Consider (E4). If $m_{j,\ell} = 0$, then $z_{j,\ell} = 0$. If $m_{j,\ell} = 1$, then $z_{j,\ell} \oplus u_{j,\ell} = 0$ forces $z_{j,\ell} = u_{j,\ell} = x_{j,\ell}$. Hence,
$$\sigma(z_{j,\ell}) = m_{j,\ell} \cdot x_{j,\ell}.$$

4. By (E5) and Lemma 7.7, each w_j equals the XOR of $\{z_{j,\ell}\}_{\ell\in[k]}$, so
$$\sigma(w_j) = \bigoplus_{\ell\in[k]} \sigma(z_{j,\ell}) = \bigoplus_{\ell\in[k]} (m_{j,\ell} \cdot x_{j,\ell}).$$

5. By (E6) and Lemma 7.7 again,
$$\sigma(S) = \bigoplus_{j\in[N]} \sigma(w_j) = \bigoplus_{j\in[N]} \bigoplus_{\ell\in[k]} (m_{j,\ell} \cdot x_{j,\ell}).$$

□

8.5 Correctness: nonzero solution iff SCE $= b$

Theorem 8.4 (Correctness of SysBit)**.** *$\mathsf{SysBit}(N, i, t, b, X)$ has a nonzero solution if and only if $\mathrm{SCE}_{N,i,t}(X) = b$.*

Proof. ($\Rightarrow$) Suppose $\mathsf{SysBit}(N, i, t, b, X)$ has a nonzero solution σ. By Lemma 8.2, $\sigma(\lambda_{\text{root}}) = 1$. Then Lemma 8.3 gives

$$\sigma(S) = \bigoplus_{j,\ell} (m_{j,\ell} \cdot x_{j,\ell}).$$

By Lemma 6.8, the right-hand side equals $\mathrm{SCE}_{N,i,t}(X)$. Finally, the check (E7) enforces $\sigma(S) = b$ when $\lambda_{\text{root}} = 1$:

- if $b = 0$, (E7) is $S = 0$;
- if $b = 1$, (E7) is $S \oplus \lambda_{\mathrm{root}} = 0$, hence $S = \lambda_{\mathrm{root}} = 1$.

Thus $\mathrm{SCE}_{N,i,t}(X) = b$.

($\Leftarrow$) Suppose $\mathrm{SCE}_{N,i,t}(X) = b$. We construct a nonzero solution σ.

- Set $\sigma(\lambda_{\mathrm{root}}) = 1$, and for each EqTree in (E1)–(E2), set every variable in that tree to 1. All equality constraints are satisfied.
- For each (j, ℓ), set $\sigma(u_{j,\ell}) = x_{j,\ell}$. Then (E3) is satisfied: if $x_{j,\ell} = 0$ it requires $u_{j,\ell} = 0$; if $x_{j,\ell} = 1$ it requires $u_{j,\ell} = \lambda_{j,\ell} = 1$.
- For each (j, ℓ), set $\sigma(z_{j,\ell}) = m_{j,\ell} \cdot x_{j,\ell}$. Then (E4) is satisfied: if $m_{j,\ell} = 0$ it requires $z_{j,\ell} = 0$; if $m_{j,\ell} = 1$ it requires $z_{j,\ell} = u_{j,\ell}$.
- For each XorTree in (E5) and (E6), assign internal node values bottom-up so that each internal equation $s \oplus u \oplus v = 0$ holds; equivalently, set each internal node s to $\sigma(u) \oplus \sigma(v)$. Then the gadget constraints are satisfied and Lemma 7.7 holds. In particular,

$$\sigma(w_j) = \bigoplus_{\ell} \sigma(z_{j,\ell}), \qquad \sigma(S) = \bigoplus_{j} \sigma(w_j) = \bigoplus_{j,\ell} m_{j,\ell} x_{j,\ell}.$$

  By Lemma 6.8, the right-hand side equals $\mathrm{SCE}_{N,i,t}(X) = b$.
- Finally, (E7) is satisfied because $\sigma(S) = b$ and $\sigma(\lambda_{\mathrm{root}}) = 1$.

Thus σ is a nonzero solution. □

8.6 Bounded arity and bounded occurrence

Lemma 8.5 (Bounded arity and bounded occurrence of SysBit)**.** *All equations in* SysBit(N, i, t, b, X) *have arity at most 3. Moreover, there exists an absolute constant B (independent of N, i, t, b, X) such that every variable in* SysBit *occurs in at most B equations.*

Proof. Arity:

- EqTree constraints (E1)–(E2) are equalities of arity 2.
- XorTree constraints (E5)–(E6) have arity 3.
- The gating equations (E3)–(E4) have arity 1 or 2.
- The final check (E7) has arity 1 or 2.

Thus the maximum arity is 3.

Occurrence: choose all EqTree gadgets as rooted binary trees and all XorTree gadgets as rooted binary trees. Then:

- Within EqTree, each variable occurs in at most 3 equations (Lemma 7.8).
- Within XorTree, each variable occurs in at most 2 equations (Lemma 7.8).

Now check cross-gadget participation for each class of designated variables:

- Each λ_j participates in one EqTree in (E1) and one EqTree in (E2) and nowhere else.
- Each $\lambda_{j,\ell}$ participates in EqTree (E2) and in at most one gating equation in (E3).
- Each $u_{j,\ell}$ participates in at most one equation in (E3) and at most one equation in (E4).
- Each $z_{j,\ell}$ participates in at most one equation in (E4) and in exactly one XorTree equation (as a leaf) in (E5).
- Each w_j participates in one XorTree (E5) and one XorTree (E6).
- S participates in one XorTree (E6) and one check equation (E7).
- λ_{root} participates in EqTree (E1) and possibly in (E7) if $b = 1$.

Therefore each variable's total occurrence is bounded by a fixed constant (one may take $B = 8$, for example), independent of the parameters. □

Remark 8.6 (Size vs succinctness)**.** *The system* $\mathsf{SysBit}(N, i, t, b, X)$ *contains* $\Theta(Nk)$ *leaf-level variables and constraints. Later, we will represent such systems succinctly by local access, using the fact that each gating choice depends on either (i) evaluator bits* $x_{j,\ell}$*, obtainable from evaluating* $X(j)$*, or (ii) mask bits* $m_{j,\ell}$*, computable in* $\mathrm{poly}(k)$ *time by Lemma 6.9. This "localizability" will be essential when translating* SysBit *into a succinct topological instance.*

From Bounded $\mathbb{F}_2$-Linear Systems to CW 2-Complexes and the Isomorphism $H_2 \cong \mathsf{Sol}$

---

9 A CW 2-Complex Encoding of a Bounded Linear System

This part constructs, from a bounded-arity, bounded-occurrence homogeneous $\mathbb{F}_2$-linear system $\mathsf{Sys} = (\mathcal{V}, \mathcal{E})$, a 2-dimensional CW complex $K^{\text{cw}}(\mathsf{Sys})$ such that the second cellular homology group $H_2(K^{\text{cw}}(\mathsf{Sys}); \mathbb{F}_2)$ is canonically isomorphic to the solution space $\mathsf{Sol}(\mathsf{Sys})$.

The construction is designed to have bounded local complexity when Sys has bounded arity and occurrence, which will later support bounded-degree simplicialization (Part 4/5).

9.1 Conventions for input systems (boundedness and ordering)

Recall from Definition 7.1 that a homogeneous system $\mathsf{Sys} = (\mathcal{V}, \mathcal{E})$ consists of variables $\mathcal{V}$ and equations $\mathcal{E}$, where each equation $e \in \mathcal{E}$ is of the form

$$\bigoplus_{v \in S_e} v = 0$$

for some subset $S_e \subseteq \mathcal{V}$.

Convention 9.1 (Equation IDs, no duplicates inside a single equation, and incidence ordering)**.** *For the remainder of this part, we impose the following conventions.*

1. *We treat $\mathcal{E}$ as an indexed list of equations, i.e., $\mathcal{E} = [m_E]$ for some m_E, and we refer to equations by their IDs $e \in [m_E]$. (If two equations are syntactically identical, they still have different IDs and are treated as distinct.)*

2. *For each equation e, we treat $S_e \subseteq \mathcal{V}$ as a set (no repeated variable within a single XOR). This is without loss of generality because repetitions cancel over $\mathbb{F}_2$.*

3. *For each variable $v \in \mathcal{V}$, define its incidence set*

$$\mathrm{Inc}(v) := \{e \in \mathcal{E} : v \in S_e\}.$$

We fix a canonical ordering of $\mathrm{Inc}(v)$ *as an ordered list*

$$\mathrm{Inc}(v) = (e_{v,0}, e_{v,1}, \ldots, e_{v,r_v-1}), \qquad r_v := |\mathrm{Inc}(v)|.$$

For concreteness, we may take the increasing order of IDs.

4. *We assume boundedness: each equation has arity $|S_e| \leq 3$, and each variable occurs in at most B equations, i.e., $r_v \leq B$ for all v. (This is the situation for* SysBit *by Lemma 8.5.)*

These conventions are needed to define attaching maps deterministically and to support local access later.

9.2 Equation triangles in the 1-skeleton

We begin by defining, for each equation e, a distinguished 3-cycle in the 1-skeleton that will serve as a "basis cycle" corresponding to that equation.

Definition 9.2 (Equation triangle gadget and the 1-cycle $\triangle_e$)**.** *For each equation ID $e \in \mathcal{E}$, introduce three new vertices*

$$p(e,0),\ p(e,1),\ p(e,2),$$

and three edges

$$\{p(e,0), p(e,1)\},\ \{p(e,1), p(e,2)\},\ \{p(e,2), p(e,0)\}.$$

These edges form a 3-cycle in the 1-skeleton. We denote the corresponding formal 1-chain over $\mathbb{F}_2$ by

$$\triangle_e \ := \ \{p(e,0), p(e,1)\} \ \oplus \ \{p(e,1), p(e,2)\} \ \oplus \ \{p(e,2), p(e,0)\}.$$

Importantly, in our construction $\triangle_e$ will not be filled by any 2-cell of its own; it remains a 1-cycle that appears in boundaries of other 2-cells.

9.3 The CW 2-complex $K^{\mathrm{cw}}(\mathsf{Sys})$

We now define the CW complex. Intuitively, each variable v contributes a 2-cell F_v. The boundary of F_v "loops around" $\triangle_e$ once for each incident equation $e \in \mathrm{Inc}(v)$. Over $\mathbb{F}_2$, this will make $\partial_2(F_v)$ equal to the XOR of those $\triangle_e$, and hence the kernel condition $\partial_2(\bigoplus_v \sigma(v) F_v) = 0$ will encode the satisfaction of every equation.

Definition 9.3 (The CW 2-complex $K^{\mathrm{cw}}(\mathsf{Sys})$)**.** *Let $\mathsf{Sys} = (\mathcal{V}, \mathcal{E})$ satisfy Convention 9.1. Define a 2-dimensional CW complex $K^{\mathrm{cw}}(\mathsf{Sys})$ as follows.*

(i) 0-cells (vertices). *The vertex set $X^{(0)}$ consists of:*

- *A vertex $b(v)$ for each variable $v \in \mathcal{V}$ (think "basepoint of v").*
- *A vertex $u(v,e)$ for each incidence pair (v,e) with $v \in S_e$.*
- *The equation triangle vertices $p(e,0), p(e,1), p(e,2)$ for each equation ID $e \in \mathcal{E}$ (from Definition 9.2).*

All these vertices are distinct by construction.

(ii) 1-cells (edges). *The edge set $X^{(1)}$ consists of:*

- *For each incidence (v, e) with $v \in S_e$, two "connector" edges*

$$\{b(v), u(v, e)\} \quad \textit{and} \quad \{u(v, e), p(e, 0)\}.$$

- *For each equation e, the three triangle edges of Definition 9.2:*

$$\{p(e, 0), p(e, 1)\},\ \{p(e, 1), p(e, 2)\},\ \{p(e, 2), p(e, 0)\}.$$

Thus the 1-skeleton is a graph containing, for each equation e, a triangle $\triangle_e$, and for each incidence (v, e), a path $b(v) - u(v, e) - p(e, 0)$ linking variable v to the vertex $p(e, 0)$ of the equation triangle.

(iii) 2-cells. *For each variable $v \in \mathcal{V}$, add one 2-cell F_v, attached along a closed walk in the 1-skeleton defined by* $\mathrm{Inc}(v)$.

- *If* $\mathrm{Inc}(v) = (e_{v,0}, \ldots, e_{v,r_v-1})$ *with $r_v \geq 1$, define the attaching map of F_v as the concatenation of the following "lollipop loops," one for each $e = e_{v,j}$:*

$$b(v) \to u(v, e) \to p(e, 0) \to p(e, 1) \to p(e, 2) \to p(e, 0) \to u(v, e) \to b(v).$$

Each arrow traverses the unique edge between the two vertices. Concatenating these loops (in the fixed order of $\mathrm{Inc}(v)$*) yields a closed walk starting and ending at $b(v)$, hence a well-defined attaching map $S^1 \to X^{(1)}$.*

- *If $r_v = 0$ (i.e., v appears in no equations), attach F_v along the constant loop at $b(v)$.*

This completes the construction of $K^{\mathrm{cw}}(\mathsf{Sys})$.

9.4 Bounded local complexity

Definition 9.4 (Bounded local complexity for a CW 2-complex)**.** *A 2-dimensional CW complex X has bounded local complexity if there exists an absolute constant Λ such that:*

1. *Every vertex in the 1-skeleton has degree at most Λ.*
2. *Every 2-cell is attached along a walk of length at most Λ.*
3. *Every 1-cell is incident to at most Λ distinct 2-cells.*

Lemma 9.5 (Bounded local complexity from bounded arity/occurrence)**.** *Assume* Sys *has equation arity $|S_e| \leq 3$ for all e and variable occurrence $r_v \leq B$ for all v. Then $K^{\mathrm{cw}}(\mathsf{Sys})$ has bounded local complexity with a bound $\Lambda = \Lambda(B)$ depending only on B.*

Proof. 1. Vertex degrees. Consider each vertex type:

- $b(v)$ is adjacent only to vertices $u(v, e)$ for $e \in \mathrm{Inc}(v)$. Thus

$$\deg(b(v)) = r_v \leq B.$$

- $u(v, e)$ is adjacent to exactly $b(v)$ and $p(e, 0)$, so $\deg(u(v, e)) = 2$.
- $p(e, 1)$ and $p(e, 2)$ lie only on the equation triangle, so each has degree 2.

- $p(e,0)$ is adjacent to $p(e,1)$ and $p(e,2)$, and also to $u(v,e)$ for each $v \in S_e$. Hence
$$\deg(p(e,0)) \le 2 + |S_e| \le 5.$$

  Therefore $\deg(\cdot)$ is bounded by $\max\{B, 5\}$.

2. Boundary length of F_v. Each lollipop loop for an incidence (v,e) traverses exactly:
   - 2 connector edges $b(v) - u(v,e)$ and $u(v,e) - p(e,0)$ forward,
   - 3 triangle edges around $\triangle_e$,
   - the same 2 connector edges backward,

   for a total of 7 edge traversals. The attaching walk of F_v is the concatenation of r_v such loops, so its length is $7r_v \le 7B$. If $r_v = 0$, the length is 0. Hence bounded by $7B$.

3. Number of 2-cells incident to a 1-cell.
   - A connector edge $\{b(v), u(v,e)\}$ appears only in the boundary walk of F_v, hence is incident to exactly one 2-cell.
   - A connector edge $\{u(v,e), p(e,0)\}$ also appears only in the boundary of F_v, hence is incident to exactly one 2-cell.
   - A triangle edge $\{p(e,a), p(e,a')\}$ for fixed e appears in the lollipop loop of F_v if and only if $v \in S_e$. Hence it is incident to exactly $|S_e| \le 3$ distinct 2-cells.

   Thus each 1-cell is incident to at most 3 2-cells.

Combining (1)–(3) gives bounded local complexity with $\Lambda = \max\{B, 7B, 3, 5\} = 7B$ (for example). □

---

10 Cellular Chains and the Isomorphism $H_2 \cong \mathsf{Sol}$

We now define the cellular chain complex of $K^{\mathrm{cw}}(\mathsf{Sys})$ over $\mathbb{F}_2$, compute the boundaries of the variable 2-cells, and prove that H_2 is naturally isomorphic to the solution space of Sys.

10.1 Cellular chain groups and boundary map over $\mathbb{F}_2$

Let $X = K^{\mathrm{cw}}(\mathsf{Sys})$. Since X has cells only in dimensions 0,1,2, its cellular chain complex over $\mathbb{F}_2$ is
$$0 \longrightarrow C_2(X) \xrightarrow{\partial_2} C_1(X) \xrightarrow{\partial_1} C_0(X) \longrightarrow 0.$$

Definition 10.1 (Cellular chain groups for $K^{\mathrm{cw}}(\mathsf{Sys})$)**.** *Let*

- $\mathcal{F} := \{F_v : v \in \mathcal{V}\}$ *be the set of 2-cells,*
- $X^{(1)}$ *be the set of 1-cells (edges) from Definition 9.3,*
- $X^{(0)}$ *be the set of 0-cells (vertices) from Definition 9.3.*

Define
$$C_2(X) := \mathbb{F}_2^{\mathcal{F}}, \qquad C_1(X) := \mathbb{F}_2^{X^{(1)}}, \qquad C_0(X) := \mathbb{F}_2^{X^{(0)}}.$$

Definition 10.2 (Cellular boundary map ∂_2 over $\mathbb{F}_2$)**.** *For each 2-cell F_v, its attaching map is a closed walk in the 1-skeleton specified as an edge sequence (Definition 9.3). Define $\partial_2(F_v) \in C_1(X)$ to be the XOR (sum in $\mathbb{F}_2$) of all 1-cells traversed an odd number of times by that walk (direction ignored, since coefficients are mod 2). Extend ∂_2 $\mathbb{F}_2$-linearly to all of $C_2(X)$.*

Remark 10.3. *Over $\mathbb{F}_2$, this definition matches the standard cellular boundary definition via incidence numbers mod 2; the combinatorial closed-walk description is sufficient for our purposes.*

Since there are no 3-cells, $C_3(X) = 0$ and $\operatorname{im}(\partial_3) = \{0\}$. Therefore

$$H_2(X; \mathbb{F}_2) = \ker(\partial_2).$$

We will use this fact in the main theorem below.

10.2 Boundary of a variable 2-cell

Lemma 10.4 (Boundary of F_v)**.** *For each variable $v \in \mathcal{V}$,*

$$\partial_2(F_v) \;=\; \bigoplus_{e \in \mathrm{Inc}(v)} \triangle_e \quad \in C_1(X),$$

where $\triangle_e$ is the equation 1-cycle from Definition 9.2.

Proof. Fix v. If $\mathrm{Inc}(v) = \varnothing$, then F_v is attached along the constant loop at $b(v)$, hence no edges are traversed and $\partial_2(F_v) = 0$, matching the empty XOR.

Assume $\mathrm{Inc}(v) \neq \varnothing$. By Definition 9.3, the attaching walk of F_v is a concatenation of "lollipop loops," one for each $e \in \mathrm{Inc}(v)$. It suffices to compute the contribution of one such loop.

Fix $e \in \mathrm{Inc}(v)$. The lollipop loop traverses edges in the sequence:

$$\{b(v), u(v,e)\},\ \{u(v,e), p(e,0)\},\ \{p(e,0), p(e,1)\},\ \{p(e,1), p(e,2)\},\ \{p(e,2), p(e,0)\},$$

and then returns via

$$\{p(e,0), u(v,e)\},\ \{u(v,e), b(v)\}.$$

Thus each connector edge $\{b(v), u(v,e)\}$ is traversed exactly twice (forward and backward), and each connector edge $\{u(v,e), p(e,0)\}$ is traversed exactly twice. Over $\mathbb{F}_2$, these cancel in $\partial_2(F_v)$.

Each triangle edge $\{p(e,0), p(e,1)\}$, $\{p(e,1), p(e,2)\}$, $\{p(e,2), p(e,0)\}$ is traversed exactly once in that loop, so all three appear in $\partial_2(F_v)$. Therefore the contribution of the lollipop loop for e is exactly $\triangle_e$.

Finally, since the full attaching walk is a concatenation of these lollipop loops for all $e \in \mathrm{Inc}(v)$, and since ∂_2 is computed by parity of traversals, we obtain

$$\partial_2(F_v) = \bigoplus_{e \in \mathrm{Inc}(v)} \triangle_e.$$

□

10.3 The variable-to-2-chain map

We now relate assignments $\sigma : \mathcal{V} \to \mathbb{F}_2$ to 2-chains.

Definition 10.5 (Solution space)**.** *The solution space of* $\mathsf{Sys} = (\mathcal{V}, \mathcal{E})$ *is*

$$\mathsf{Sol}(\mathsf{Sys}) := \left\{ \sigma : \mathcal{V} \to \mathbb{F}_2 : \ \forall e \in \mathcal{E}, \ \bigoplus_{v \in S_e} \sigma(v) = 0 \right\}.$$

Definition 10.6 (Variable-to-2-chain map Φ)**.** *Define a linear map*

$$\Phi : \mathbb{F}_2^{\mathcal{V}} \to C_2(X) \quad \textit{by} \quad \Phi(\sigma) := \bigoplus_{v \in \mathcal{V}} \sigma(v)\, F_v,$$

where we identify σ *with its coordinate vector in* $\mathbb{F}_2^{\mathcal{V}}$*, and* $\{F_v\}$ *is the canonical basis of* $C_2(X) = \mathbb{F}_2^{\mathcal{F}}$*.*

Lemma 10.7 (Boundary of $\Phi(\sigma)$)**.** *For every* $\sigma : \mathcal{V} \to \mathbb{F}_2$*,*

$$\partial_2(\Phi(\sigma)) = \bigoplus_{e \in \mathcal{E}} \left(\bigoplus_{v \in S_e} \sigma(v) \right) \triangle_e.$$

Proof. Using linearity of ∂_2 and Lemma 10.4,

$$\partial_2(\Phi(\sigma)) = \bigoplus_{v \in \mathcal{V}} \sigma(v)\, \partial_2(F_v) = \bigoplus_{v \in \mathcal{V}} \sigma(v) \left(\bigoplus_{e \in \mathrm{Inc}(v)} \triangle_e \right).$$

Rearranging the XOR by grouping terms for each fixed equation e, note that $\triangle_e$ appears in $\partial_2(F_v)$ if and only if $e \in \mathrm{Inc}(v)$, i.e., if and only if $v \in S_e$. Therefore, for a fixed e, the total coefficient of $\triangle_e$ is exactly $\bigoplus_{v \in S_e} \sigma(v)$. This yields the claimed formula. □

10.4 Independence of equation triangles

To conclude that $\partial_2(\Phi(\sigma)) = 0$ forces each equation constraint, we need a linear independence fact.

Lemma 10.8 (Linear independence of $\{\triangle_e\}_{e \in \mathcal{E}}$ in $C_1(X)$)**.** *The set of 1-chains* $\{\triangle_e : e \in \mathcal{E}\} \subseteq C_1(X)$ *is linearly independent over* $\mathbb{F}_2$*.*

Proof. Consider any linear combination

$$\bigoplus_{e \in \mathcal{E}} c_e \triangle_e = 0 \quad \text{in } C_1(X), \qquad c_e \in \mathbb{F}_2.$$

Fix an equation ID $e_0 \in \mathcal{E}$. The edge $\{p(e_0, 0), p(e_0, 1)\}$ appears in $\triangle_{e_0}$ and does not appear in $\triangle_e$ for any $e \neq e_0$, because all equation triangle vertices and triangle edges are distinct across different equation IDs by construction (Definition 9.2 and Definition 9.3).

Thus, in the sum $\bigoplus_e c_e \triangle_e$, the coefficient (parity) of the specific edge $\{p(e_0, 0), p(e_0, 1)\}$ is exactly c_{e_0}. Since the total sum is the zero 1-chain, this coefficient must be 0. Therefore $c_{e_0} = 0$. As e_0 was arbitrary, all coefficients c_e are 0, proving independence. □

10.5 Main theorem: $H_2(K^{\mathrm{cw}}(\mathsf{Sys}); \mathbb{F}_2) \cong \mathsf{Sol}(\mathsf{Sys})$

Theorem 10.9 (Second homology equals solution space)**.** *Let* $\mathsf{Sys} = (\mathcal{V}, \mathcal{E})$ *satisfy Convention 9.1 and let* $X = K^{\mathrm{cw}}(\mathsf{Sys})$*. Then*

$$H_2(X; \mathbb{F}_2) \ \cong \ \mathsf{Sol}(\mathsf{Sys})$$

as $\mathbb{F}_2$*-vector spaces. More precisely, the map* Φ *from Definition 10.6 restricts to a linear isomorphism*

$$\Phi : \mathsf{Sol}(\mathsf{Sys}) \xrightarrow{\cong} \ker(\partial_2) = H_2(X; \mathbb{F}_2).$$

Proof. Since X is 2-dimensional, $C_3(X) = 0$, hence $\operatorname{im}(\partial_3) = \{0\}$, and therefore

$$H_2(X; \mathbb{F}_2) = \ker(\partial_2).$$

We prove that Φ gives a linear bijection between $\mathsf{Sol}(\mathsf{Sys})$ and $\ker(\partial_2)$.

1. ($\Phi(\mathsf{Sol}) \subseteq \ker \partial_2$). Let $\sigma \in \mathsf{Sol}(\mathsf{Sys})$. Then for every equation e, $\bigoplus_{v \in S_e} \sigma(v) = 0$. By Lemma 10.7,

$$\partial_2(\Phi(\sigma)) = \bigoplus_{e \in \mathcal{E}} \left(\bigoplus_{v \in S_e} \sigma(v) \right) \triangle_e = \bigoplus_{e \in \mathcal{E}} 0 \cdot \triangle_e = 0.$$

   Thus $\Phi(\sigma) \in \ker(\partial_2)$.

2. ($\ker \partial_2 \subseteq \Phi(\mathsf{Sol})$). Let $c \in \ker(\partial_2)$. Since $C_2(X) = \mathbb{F}_2^{\mathcal{F}}$ with basis $\{F_v\}_{v \in \mathcal{V}}$, there is a unique coefficient vector $\sigma : \mathcal{V} \to \mathbb{F}_2$ such that

$$c = \bigoplus_{v \in \mathcal{V}} \sigma(v) F_v = \Phi(\sigma).$$

   Since $c \in \ker(\partial_2)$, we have $\partial_2(\Phi(\sigma)) = 0$. By Lemma 10.7,

$$0 = \partial_2(\Phi(\sigma)) = \bigoplus_{e \in \mathcal{E}} \left(\bigoplus_{v \in S_e} \sigma(v) \right) \triangle_e.$$

   By Lemma 10.8, the $\triangle_e$ are linearly independent in $C_1(X)$, so each coefficient must be 0:

$$\forall e \in \mathcal{E}, \qquad \bigoplus_{v \in S_e} \sigma(v) = 0.$$

   Hence $\sigma \in \mathsf{Sol}(\mathsf{Sys})$ and $c = \Phi(\sigma) \in \Phi(\mathsf{Sol}(\mathsf{Sys}))$.

3. Injectivity of Φ on Sol. If $\Phi(\sigma) = 0$ in $C_2(X)$, then all coefficients of the basis elements F_v are zero, so $\sigma(v) = 0$ for all v. Thus Φ is injective.

Combining (1)–(3), Φ is a linear isomorphism $\mathsf{Sol}(\mathsf{Sys}) \cong \ker(\partial_2) = H_2(X; \mathbb{F}_2)$. □

Corollary 10.10 (Betti-2 positivity equals existence of a nonzero solution)**.** *Let* $X = K^{\mathrm{cw}}(\mathsf{Sys})$. *Then*

$$\beta_2(X; \mathbb{F}_2) > 0 \iff \mathsf{Sys} \textit{ has a nonzero solution.}$$

Proof. By Theorem 10.9, $\beta_2(X; \mathbb{F}_2) = \dim H_2(X; \mathbb{F}_2) = \dim \mathsf{Sol}(\mathsf{Sys})$. This dimension is positive if and only if the solution space contains some nonzero vector, i.e., Sys has a nonzero solution. □

10.6 Application to SysBit and SCE-Dec (CW level)

We combine the correctness of SysBit from Part 2 with the topological encoding above.

Corollary 10.11 (CW encoding of the SCE-Dec bit)**.** *Fix parameters* (N, i, t, b, X) *and let* $\mathsf{Sys} = \mathsf{SysBit}(N, i, t, b, X)$. *Let*

$$X^{\mathrm{cw}}_{N,i,t,b,X} := K^{\mathrm{cw}}(\mathsf{SysBit}(N, i, t, b, X)).$$

Then

$$\beta_2\Big(X^{\mathrm{cw}}_{N,i,t,b,X}; \mathbb{F}_2\Big) > 0 \iff \mathrm{SCE}_{N,i,t}(X) = b.$$

Proof. By Theorem 8.4 (Part 2), $\mathsf{SysBit}(N, i, t, b, X)$ has a nonzero solution if and only if $\mathrm{SCE}_{N,i,t}(X) = b$. By Corollary 10.10, $\beta_2(K^{\mathrm{cw}}(\mathsf{SysBit}); \mathbb{F}_2) > 0$ if and only if SysBit has a nonzero solution. Combining yields the equivalence. □

Simplicialization, the ProbeBit Mapping, $\oplus$P-Completeness on $\mathrm{Im}(\mathrm{ProbeBit})$, SAT $\leq_{rp}$ via Isolation, and an Evaluation-Local Lower Bound

11 From $K^{\mathrm{cw}}(\mathsf{Sys})$ to a Bounded-Degree Simplicial 2-Complex

11.1 Why the simplicialization must respect boundary occurrences

In Part 3 we constructed, from a bounded-arity/occurrence homogeneous system $\mathsf{Sys} = (\mathcal{V}, \mathcal{E})$, a CW 2-complex

$$X \ := \ K^{\mathrm{cw}}(\mathsf{Sys})$$

with one 2-cell F_v per variable $v \in \mathcal{V}$. The proof of Theorem 10.9 depended critically on the fact that each variable contributes a single 2-cell, so that the only 2-chains are $\mathbb{F}_2$-linear combinations of these F_v.

A naïve "cone from a center vertex" simplicialization that identifies repeated boundary vertices would generally introduce extra 2-chains that can "select subsets of incident equation-cycles," potentially enlarging H_2 and breaking the isomorphism $H_2 \cong \mathsf{Sol}(\mathsf{Sys})$. To avoid this, we must triangulate each 2-cell F_v using distinct boundary vertices per boundary occurrence before gluing to the 1-skeleton. The resulting structure is a triangular CW complex (a Δ-complex in the classical sense), and we then apply a barycentric subdivision to obtain an honest simplicial complex.

This section formalizes that two-step process and proves that it preserves geometric realization (hence homology).

11.2 The boundary walk for each CW 2-cell

Let $X = K^{\mathrm{cw}}(\mathsf{Sys})$ be as in Definition 9.3. Let $G := X^{(1)}$ denote its 1-skeleton (a finite graph).

For each $v \in \mathcal{V}$, the attaching map of the CW 2-cell F_v is, by construction (Definition 9.3), a closed walk in G. We fix a concrete encoding of that walk as a vertex sequence.

Definition 11.1 (Boundary walk encoding W_v)**.** *For each $v \in \mathcal{V}$, let*

$$W_v \ := \ (w_0^{(v)}, w_1^{(v)}, \dots, w_{L_v}^{(v)})$$

be the vertex sequence of the attaching walk of F_v, with

$$w_{L_v}^{(v)} = w_0^{(v)}, \qquad \{w_j^{(v)}, w_{j+1}^{(v)}\} \in E(G) \ \ \textit{for all } j \in \{0, \dots, L_v - 1\}.$$

(Thus L_v is the number of edge-steps in the closed walk.)

Remark 11.2 (Boundedness)**.** *If Sys has variable occurrence $r_v := |\mathrm{Inc}(v)| \leq B$, then in our CW encoding (Definition 9.3) the boundary walk is a concatenation of r_v "lollipop loops" each of length 7, so*

$$L_v \leq 7B.$$

This bound will be used later to prove bounded degree after subdivision.

11.3 Triangulating each 2-cell as a disk with distinct boundary occurrences

We now define a triangulated disk that has a boundary cycle of length L_v, with distinct boundary vertices $q_0^{(v)}, \dots, q_{L_v-1}^{(v)}$. These are "formal boundary corners" and are not the vertices $w_j^{(v)}$ in the 1-skeleton.

Definition 11.3 (The fan triangulation T_v of a polygonal disk)**.** *Fix $v \in \mathcal{V}$ with boundary length $L_v \geq 1$. Define a simplicial 2-complex T_v as follows.*

- ***Vertices:***
$$V(T_v) \;:=\; \{c_v\} \;\cup\; \{q_j^{(v)} : j \in [L_v]\},$$
where all vertices are distinct and c_v is a designated "center."
- ***Edges:*** *include all boundary edges $\{q_j^{(v)}, q_{j+1}^{(v)}\}$ for $j \in [L_v]$ (indices mod L_v), and all cone edges $\{c_v, q_j^{(v)}\}$ for $j \in [L_v]$.*
- ***Triangles:*** *for each $j \in [L_v]$, include the triangle*
$$\tau_j^{(v)} \;:=\; \{c_v, q_j^{(v)}, q_{j+1}^{(v)}\}.$$

This is the standard fan triangulation of an L_v-gon from a center vertex.

Lemma 11.4 ($|T_v|$ is a topological disk)**.** *The geometric realization $|T_v|$ is homeomorphic to the closed disk D^2, and the subcomplex induced by $\{q_j^{(v)}\}_{j\in[L_v]}$ realizes as a simple cycle homeomorphic to S^1.*

Proof. Choose a strictly convex polygon $P \subset \mathbb{R}^2$ with vertices $p_0, \ldots, p_{L_v-1}$ in cyclic order, and choose a point $p_\star$ in the interior of P. Define a simplicial map $f : |T_v| \to P$ by sending

$$c_v \mapsto p_\star, \qquad q_j^{(v)} \mapsto p_j,$$

and extending linearly on each triangle $\tau_j^{(v)}$. Because P is strictly convex and $p_\star$ is interior, the images of the triangles $\tau_j^{(v)}$ are non-overlapping (except along shared edges) and cover P. Thus f is a continuous bijection from compact $|T_v|$ onto Hausdorff P, hence a homeomorphism. Since $P \cong D^2$ and its boundary $\partial P \cong S^1$, the claim follows. □

11.4 Gluing triangulated disks to the 1-skeleton: a triangular CW complex

We now glue each disk $|T_v|$ to the 1-skeleton G along the boundary walk W_v. The key is that the boundary vertices $q_j^{(v)}$ are distinct in T_v, but are mapped to the possibly repeating vertices $w_j^{(v)}$ in G.

Definition 11.5 (Boundary identification map γ_v)**.** *Define a continuous map*

$$\gamma_v : \; |\partial T_v| \;\longrightarrow\; |G|$$

by mapping each boundary vertex $q_j^{(v)}$ to the vertex $w_j^{(v)} \in V(G)$, and mapping each boundary edge $\{q_j^{(v)}, q_{j+1}^{(v)}\}$ homeomorphically onto the edge $\{w_j^{(v)}, w_{j+1}^{(v)}\} \in E(G)$ (which exists by Definition 11.1). This defines γ_v uniquely on $|\partial T_v|$ because $|\partial T_v|$ is a cycle subdivided into edges.

Definition 11.6 (Triangular subdivision complex $K^\triangle(\mathsf{Sys})$)**.** *Define the space*

$$Y \;:=\; K^\triangle(\mathsf{Sys})$$

as the pushout (quotient space)

$$Y \;:=\; \left(|G| \;\sqcup\; \bigsqcup_{v\in\mathcal{V}} |T_v|\right) \Big/ \sim,$$

where for each v, and for each $x \in |\partial T_v|$, we identify $x \sim \gamma_v(x) \in |G|$. Equivalently, Y is obtained from $|G|$ by attaching the disk $|T_v|$ along its boundary via γ_v, for each variable v.

Remark 11.7 (What Y is and is not)**.**

- *Y is a finite 2-dimensional CW complex whose 2-cells are triangles $\tau_j^{(v)}$.*
- *Y may have multiple 1-cells with the same endpoints, and may have multiple 2-cells whose vertex sets coincide after boundary identifications. Therefore Y is generally not a simplicial complex in the strict sense of Definition 2.5.*
- *This is why we apply barycentric subdivision (Section 11.6) to obtain an honest simplicial complex without losing topology.*

11.5 $K^{\triangle}(\mathsf{Sys})$ is homeomorphic to $K^{\mathrm{cw}}(\mathsf{Sys})$

We now show that replacing each CW disk F_v by the triangulated disk T_v does not change the space.

Let $X := K^{\mathrm{cw}}(\mathsf{Sys})$ be the CW complex from Part 3. By definition, X is also a pushout

$$|X| \;:=\; \left(|G| \;\sqcup\; \bigsqcup_{v\in\mathcal{V}} D_v\right) \Big/ \approx,$$

where each $D_v \cong D^2$ is a copy of the disk attached along its boundary by the attaching map $f_v : S^1 \to |G|$ determined by W_v.

Lemma 11.8 (Boundary-fixing homeomorphism between $|T_v|$ and D^2)**.** *For each $v \in \mathcal{V}$, there exists a homeomorphism*

$$h_v :\; |T_v| \;\longrightarrow\; D_v$$

such that h_v restricts to a homeomorphism $|\partial T_v| \to \partial D_v$.

Proof. By Lemma 11.4, $|T_v| \cong D^2$. Choose any homeomorphism h_v from $|T_v|$ onto the specific disk copy D_v; by composing with a boundary homeomorphism of D_v if needed, we may ensure $h_v(|\partial T_v|) = \partial D_v$. □

Lemma 11.9 (Homeomorphism of pushouts under boundary-fixing maps)**.** *Let A be a compact Hausdorff space, and let $\{B_i\}_{i\in I}$ be compact Hausdorff spaces with closed subspaces $C_i \subseteq B_i$. Suppose we are given continuous gluing maps $\phi_i : C_i \to A$. Suppose further that for each i there is a homeomorphism $h_i : B_i \to B_i'$ to another compact Hausdorff space B_i' that maps C_i homeomorphically onto a closed subspace $C_i' \subseteq B_i'$. Define $\phi_i' : C_i' \to A$ by $\phi_i' := \phi_i \circ (h_i|_{C_i})^{-1}$. Then the resulting pushout quotients*

$$Z := \left(A \sqcup \bigsqcup_i B_i\right) / \sim \quad \text{and} \quad Z' := \left(A \sqcup \bigsqcup_i B_i'\right) / \sim'$$

are homeomorphic.

Proof. Define a map $H : A \sqcup \bigsqcup_i B_i \to A \sqcup \bigsqcup_i B_i'$ by $H|_A = \mathrm{id}_A$ and $H|_{B_i} = h_i$. This is a continuous bijection between compact Hausdorff spaces, hence a homeomorphism. Moreover, by construction H respects the equivalence relations $\sim$ and $\sim'$ on the respective disjoint unions (the identifications are matched on the glued boundaries). Therefore H descends to a continuous bijection $\overline{H} : Z \to Z'$. Since Z and Z' are compact Hausdorff quotients of compact Hausdorff spaces, $\overline{H}$ is a homeomorphism. □

Theorem 11.10 (Triangular subdivision preserves the space)**.** *Let* Sys *satisfy Convention 9.1, and let*

$$X = K^{\mathrm{cw}}(\mathsf{Sys}), \qquad Y = K^{\triangle}(\mathsf{Sys}).$$

Then $|X|$ *and* $|Y|$ *are homeomorphic. Consequently,*

$$H_q(X;\mathbb{F}_2) \;\cong\; H_q(Y;\mathbb{F}_2) \quad \textit{for all } q.$$

Proof. Apply Lemma 11.9 with $A = |G|$, and $B_v = D_v$ (CW disks) versus $B'_v = |T_v|$ (triangulated disks), using the boundary maps induced by the boundary walks W_v. Lemma 11.8 supplies the needed homeomorphisms between disks. Thus the pushouts are homeomorphic. Homology invariance under homeomorphism gives the isomorphisms. □

11.6 Barycentric subdivision to obtain an honest simplicial 2-complex

We now define a simplicial complex whose geometric realization is homeomorphic to $|Y|$. Since Y is a triangular CW complex, we can barycentrically subdivide each triangular 2-cell and each 1-cell to obtain a genuine simplicial complex.

Definition 11.11 (Barycentric subdivision $\mathrm{sd}(Y)$ in dimension 2)**.** *Let* Y *be a triangular CW complex (all* 2*-cells are triangles). Define a simplicial* 2*-complex* $\mathrm{sd}(Y)$ *as follows:*

- ***Vertices:*** *vertices of* $\mathrm{sd}(Y)$ *correspond to cells of* Y *of dimensions* $0, 1, 2$*. For each cell* σ *of* Y*, denote its barycentric vertex by* $\mathrm{bary}(\sigma)$*.*
- ***Edges:*** *include an edge* $\{\mathrm{bary}(\sigma), \mathrm{bary}(\tau)\}$ *whenever* σ *is a face of* τ *and* $\dim(\tau) = \dim(\sigma)+1$*.*
- ***Triangles:*** *include a triangle* $\{\mathrm{bary}(u), \mathrm{bary}(e), \mathrm{bary}(t)\}$ *whenever* $u \subset e \subset t$ *is a chain of incident cells in* Y *with* $\dim(u) = 0$*,* $\dim(e) = 1$*,* $\dim(t) = 2$*.*

This yields a simplicial 2*-complex by construction (simplices are indexed by strict chains of cells).*

Lemma 11.12 (Barycentric subdivision preserves geometric realization)**.**

$$|\mathrm{sd}(Y)| \;\cong\; |Y|.$$

Proof. It suffices to define a homeomorphism on each triangular 2-cell and check compatibility on shared faces.

Fix a triangular 2-cell t of Y with vertices u_0, u_1, u_2 and edges e_{01}, e_{12}, e_{20}. The barycentric subdivision $\mathrm{sd}(t)$ consists of 6 small triangles, each with vertex set $\{\mathrm{bary}(u_i), \mathrm{bary}(e_{ij}), \mathrm{bary}(t)\}$. There is a standard piecewise-linear homeomorphism from $|\mathrm{sd}(t)|$ onto $|t|$ obtained by sending the barycentric vertices to the geometric barycenters of the corresponding cells (in any fixed geometric realization of t as a Euclidean triangle) and extending linearly over each small triangle. This map is a homeomorphism because it is a bijective PL map between compact polyhedra.

When two triangles t and t' in Y share a common edge, the barycentric subdivision is defined using the same barycentric vertices of that shared edge and its endpoints, so the two PL maps agree on the shared subdivided edge. Therefore the local homeomorphisms glue to a global homeomorphism $|\mathrm{sd}(Y)| \to |Y|$. □

Definition 11.13 (Final simplicial complex $K(\mathsf{Sys})$)**.** *Define*

$$K(\mathsf{Sys}) \;:=\; \mathrm{sd}\Big(K^{\triangle}(\mathsf{Sys})\Big).$$

By Lemma 11.12, $|K(\mathsf{Sys})| \cong |K^{\triangle}(\mathsf{Sys})|$*, and by Theorem 11.10,* $|K^{\triangle}(\mathsf{Sys})| \cong |K^{\mathrm{cw}}(\mathsf{Sys})|$*. Therefore* $|K(\mathsf{Sys})| \cong |K^{\mathrm{cw}}(\mathsf{Sys})|$*.*

11.7 Homology consequences and Betti-2 positivity

Corollary 11.14 (Homology preservation from CW to simplicial)**.** *For all q,*

$$H_q(K(\mathsf{Sys}); \mathbb{F}_2) \;\cong\; H_q(K^{\mathrm{cw}}(\mathsf{Sys}); \mathbb{F}_2)\,.$$

In particular,

$$H_2(K(\mathsf{Sys}); \mathbb{F}_2) \;\cong\; \mathsf{Sol}(\mathsf{Sys}) \quad \textit{and} \quad \beta_2(K(\mathsf{Sys}); \mathbb{F}_2) > 0 \iff \mathsf{Sys} \textit{ has a nonzero solution.}$$

Proof. By Theorem 11.10 and Lemma 11.12, $|K(\mathsf{Sys})| \cong |K^{\mathrm{cw}}(\mathsf{Sys})|$. Homology over $\mathbb{F}_2$ is invariant under homeomorphism, so H_q isomorphic for all q. The identification of H_2 with $\mathsf{Sol}(\mathsf{Sys})$ then follows from Theorem 10.9. The Betti-2 positivity statement follows from Corollary 10.10. □

11.8 Bounded degree of $K(\mathsf{Sys})$

Lemma 11.15 (Bounded degree)**.** *Assume Sys has arity ≤ 3 and variable occurrence $\leq B$. Then there exists a constant $\Delta' = \Delta'(B)$ such that the simplicial 2-complex $K(\mathsf{Sys})$ is Δ'-bounded-degree (in the sense of Definition 2.9).*

Proof. By Lemma 9.5, the CW complex $K^{\mathrm{cw}}(\mathsf{Sys})$ has bounded local complexity with bound depending only on B. The triangular subdivision $K^{\triangle}(\mathsf{Sys})$ replaces each 2-cell by a disk triangulation with $L_v \leq 7B$ triangles; thus it also has bounded local complexity: each vertex is incident to only constantly many edges and triangles, and each edge is incident to only constantly many triangles (because equation edges have arity ≤ 3, and boundary/cone edges are local to a single variable disk).

In a barycentric subdivision of a triangular complex, vertices come in three types (barycenters of 0-, 1-, 2-cells). Each such barycentric vertex is adjacent only to barycenters of incident cells of neighboring dimensions:

- A 2-cell barycenter has degree exactly 3 (adjacent to its three edge barycenters).
- A 1-cell barycenter is adjacent to its two endpoint vertex barycenters and to the barycenters of triangles incident to that edge; the latter number is bounded by local complexity.
- A 0-cell barycenter is adjacent to barycenters of incident edges; again bounded by local complexity.

Thus the graph degree in the 1-skeleton of $K(\mathsf{Sys})$ is bounded by a constant depending only on B. Since $K(\mathsf{Sys})$ is a simplicial 2-complex, bounded vertex degree implies bounded incidence to edges and triangles (each triangle contributes at most two edges incident to a given vertex). Therefore $K(\mathsf{Sys})$ is Δ'-bounded-degree for some $\Delta'(B)$. □

12 ProbeBit: A Succinct Bounded-Degree Simplicial Complex Encoding an SCE Bit

We now apply Section 11 to the specific bounded system $\mathsf{SysBit}(N, i, t, b, X)$ defined in Part 2.

12.1 Definition of ProbeBit

Definition 12.1 (ProbeBit mapping)**.** *Given an SCE-Dec instance* (N, i, t, b, X)*, define:*

1. $\mathsf{Sys} := \mathsf{SysBit}(N, i, t, b, X)$ *(Definition 8.1).*

2. $K := K(\mathsf{Sys})$ *(Definition 11.13), a bounded-degree simplicial* 2*-complex.*

Define $\mathrm{ProbeBit}(N, i, t, b, X)$ *to be a local-oracle description* D *(Definition 2.10) of* K*.*

We impose the additional convention that the encoding of D *includes the tuple* (N, i, t, b) *and a canonical encoding of the circuit* X *as explicit header data, so that a promise verifier can recover* (N, i, t, b, X) *from* D *in time* $\mathrm{poly}(|D|)$.

12.2 Correctness of ProbeBit

Theorem 12.2 (ProbeBit correctness)**.** *Let* (N, i, t, b, X) *be any SCE-Dec instance and let*

$$D := \mathrm{ProbeBit}(N, i, t, b, X).$$

Then

$$\beta_2(K_D; \mathbb{F}_2) > 0 \iff \mathrm{SCE}_{N,i,t}(X) = b.$$

Proof. By Theorem 8.4, $\mathsf{SysBit}(N, i, t, b, X)$ has a nonzero solution iff $\mathrm{SCE}_{N,i,t}(X) = b$. By Corollary 11.14 applied to $\mathsf{Sys} = \mathsf{SysBit}$, we have

$$\beta_2(K(\mathsf{SysBit}(N, i, t, b, X)); \mathbb{F}_2) > 0 \iff \mathsf{SysBit}(N, i, t, b, X) \text{ has a nonzero solution.}$$

By Definition 12.1, $K_D = K(\mathsf{SysBit}(N, i, t, b, X))$. Combining these equivalences yields the claim. □

12.3 Succinctness and efficient local-oracle access

We record the (promise) succinctness needed for complexity statements. This section is intentionally explicit at the level of what must be computable; the full bit-level encoding convention is deferred to Appendix C (as already announced in Part 1).

Lemma 12.3 (Efficient local-oracle evaluation for ProbeBit outputs)**.** *There exists a choice of indexing conventions for the vertices/edges/triangles of* $K(\mathsf{SysBit}(N, i, t, b, X))$ *such that:*

1. *The description size* $|D|$ *is polynomial in* $|X| + \log N$*.*

2. *Each oracle* $\mathrm{End}_D, \mathrm{Vert}_D, \mathrm{IncE}_D, \mathrm{IncT}_D$ *can be evaluated in time* $\mathrm{poly}(|X| + \log N)$*.*

3. *The complex* K_D *is* Δ'*-bounded-degree for a constant* Δ' *independent of* (N, i, t, b, X)*.*

Proof (construction sketch; finite casework). We outline a uniform indexing that depends only on:

- the syntactic gadget structure of SysBit (EqTrees, XorTrees, and gating equations), and

- the bounded occurrence B from Lemma 8.5.

Step 1: Index SysBit variables and equations. SysBit is generated by a fixed finite collection of gadget templates (Definitions 7.4 and 7.6 and Definition 8.1). Each template produces:

- a bounded-degree graph structure for EqTree or XorTree internal nodes, and
- constant-arity equations whose endpoints/children can be computed from local indices.

We assign IDs to variables and equations by concatenating:

- a "kind tag" (which gadget and role: e.g., λ_{root}, λ_j, internal EqTree node, etc.), and
- the natural indices $(j, \ell) \in [N] \times [k]$ where applicable, encoded in binary.

Because the gadget shapes are fixed (binary trees with canonical heap indices, for example), from a variable ID one can compute its incident equation IDs in constant time (and conversely from an equation ID one can compute its variable list of size ≤ 3). This yields local access to $\mathrm{Inc}(v)$ with $|\mathrm{Inc}(v)| \leq B$.

Step 2: Build the CW complex $K^{\mathrm{cw}}(\mathsf{SysBit})$ locally. Given an equation ID e, the CW construction introduces the three equation vertices $p(e, 0), p(e, 1), p(e, 2)$. Given a variable ID v, it introduces $b(v)$. Given an incidence (v, e), it introduces $u(v, e)$. All these are computable locally from the IDs.

Step 3: Build the triangular subdivision $K^{\triangle}(\mathsf{SysBit})$ locally. For each variable v, the boundary walk W_v is obtained by iterating over the constant-size incidence list $\mathrm{Inc}(v)$ and emitting the constant-length lollipop pattern from Definition 9.3. Therefore $L_v \leq 7B$ and each step $j \mapsto (w_j^{(v)}, w_{j+1}^{(v)})$ is computable in constant time from v and j.

Each triangle 2-cell $\tau_j^{(v)}$ is determined by (v, j), and each cone edge 1-cell is determined by the boundary-vertex occurrence (v, j) (these are distinct 1-cells even when endpoints coincide in the quotient; this is exactly why we use triangular CW + barycentric subdivision).

Step 4: Index the simplicial complex $K(\mathsf{SysBit}) = \mathrm{sd}(K^{\triangle})$. Vertices of $K(\mathsf{SysBit})$ correspond to cells of $K^{\triangle}$ of dimensions $0, 1, 2$, hence can be indexed by tuples (d, id) where $d \in \{0, 1, 2\}$ and id is a cell ID from the previous steps.

Edges correspond to incidences between 0- and 1-cells or between 1- and 2-cells; triangles correspond to chains $0 \subset 1 \subset 2$. Since the local incidence degrees in $K^{\triangle}$ are bounded by constants depending only on B, the barycentric complex has bounded degree as in Lemma 11.15.

All oracle outputs reduce to:

- decoding a cell ID,
- computing a constant-size list of incident lower-/higher-dimensional cells, and
- translating these to barycentric vertices.

Each such operation is $\mathrm{poly}(\log N + |X|)$ because:

- evaluating $X(j)$ is $\mathrm{poly}(|X| + \log N)$,
- computing mask bits $m_{j,\ell}^{(N,i,t)}$ is $\mathrm{poly}(\log N)$ time by Lemma 6.9, and
- all additional computations are fixed finite arithmetic and pointer arithmetic on $O(\log N)$-bit indices.

Finally, the description size is polynomial in $|X| + \log N$ because the oracles are uniform circuits that hardwire only (N, i, t, b) and the encoding of X, plus fixed gadget logic. This completes the succinctness argument. □

Remark 12.4 (Scope note (no hidden claim))**.** *Lemma 12.3 is a "construction sketch with finite casework." It is logically sufficient for the complexity reductions below, because it specifies what must be computable and why each oracle evaluation is polynomial time in the description length. A fully enumerated bit-level encoding is deferred to Appendix C to avoid distracting from the homological spine.*

13 $\oplus$P-Completeness of Π_{β_2} on Im(ProbeBit)

Let

$$\mathcal{I}_{\mathrm{Probe}} := \mathrm{Im}(\mathrm{ProbeBit})$$

be the promise family of local-oracle descriptions produced by ProbeBit.

13.1 Membership: SCE-Dec $\in \oplus\mathbf{P}$

Lemma 13.1 (SCE-Dec $\in \oplus\mathbf{P}$)**.** SCE-Dec *(Definition 3.5) lies in* $\oplus\mathbf{P}$*.*

Proof. Fix input (N, i, t, b, X). By Lemma 6.8,

$$\mathrm{SCE}_{N,i,t}(X) = \bigoplus_{j\in[N]} \bigoplus_{\ell\in[k]} (m_{j,\ell}^{(N,i,t)} \cdot x_{j,\ell}),$$

where $x_{j,\ell}$ is the ℓ-th output bit of $X(j)$, and $m_{j,\ell}$ is computable in $\mathrm{poly}(\log N)$ time by Lemma 6.9.

Define a nondeterministic polynomial-time machine M that:

1. Nondeterministically guesses an integer $u \in \{0, \dots, 2^r - 1\}$ where $r := \lceil \log_2(Nk) \rceil$.
2. If $u \geq Nk$, reject.
3. Otherwise decode $u \mapsto (j, \ell) \in [N] \times [k]$.
4. Compute $x_{j,\ell}$ by evaluating $X(j)$ and extracting bit ℓ.
5. Compute $m_{j,\ell}^{(N,i,t)}$ by Lemma 6.9.
6. Accept iff $m_{j,\ell} \cdot x_{j,\ell} = 1$.

Then $\#\mathrm{acc}_M(N, i, t, X) \bmod 2$ equals the XOR of all terms $m_{j,\ell} x_{j,\ell}$, hence equals $\mathrm{SCE}_{N,i,t}(X)$.

To decide equality to b with parity acceptance:

- If $b = 1$, accept iff parity is 1, i.e., use M as is.
- If $b = 0$, flip parity by adding one additional always-accepting branch; then parity equals $\mathrm{SCE}_{N,i,t}(X) \oplus 1$, which is 1 iff $\mathrm{SCE}_{N,i,t}(X) = 0$.

Thus $(N, i, t, b, X) \in$ SCE-Dec iff the accepting-path count is odd, proving membership in $\oplus\mathbf{P}$. □

13.2 Hardness: $\oplus$SAT $\leq_m$ SCE-Dec

Lemma 13.2 ($\oplus$SAT $\leq_m$ SCE-Dec)**.** *There exists a deterministic polynomial-time many-one reduction from* $\oplus$SAT *to* SCE-Dec*.*

Proof. Let $\phi(y_0, \dots, y_{n-1})$ be an input Boolean formula. Set

$$N := 2^n, \qquad k := \lceil \log_2(4N) \rceil, \qquad \mathbb{K} := \mathbb{F}_{2^k},$$

and choose any $i \in [N]$ and $t \in [k]$ (for concreteness $i = 0, t = 0$).

We define an evaluator circuit X such that $\mathrm{SCE}_{N,i,t}(X)$ equals the parity of satisfying assignments of ϕ.

On input $j \in [N] = \{0, \dots, 2^n - 1\}$, the circuit X does:

1. Interpret j as a Boolean assignment $y \in \{0,1\}^n$ via binary expansion.
2. Compute $s := \phi(y) \in \{0,1\}$.
3. Compute the Cauchy coefficient $\kappa_{i,j} := (C_N)_{i,j} = (a_i - b_j)^{-1} \in \mathbb{K}$, which is well-defined and nonzero by Lemma 6.2. Compute $\kappa_{i,j}^{-1}$ in time $\mathrm{poly}(k)$.
4. Output the field element
$$x_j \;:=\; s \cdot \kappa_{i,j}^{-1} \cdot \alpha^t \;\in \mathbb{K},$$
encoded in the fixed basis $\mathcal{B}$. (If $s = 0$, output 0; if $s = 1$, output $\kappa_{i,j}^{-1}\alpha^t$.)

Now compute:

$$\begin{aligned}(C_N x(X))_i &= \sum_{j\in[N]} \kappa_{i,j} x_j \\ &= \sum_{j\in[N]} \kappa_{i,j} \cdot \left(s \cdot \kappa_{i,j}^{-1}\alpha^t\right) \\ &= \sum_{j\in[N]} s \cdot \alpha^t.\end{aligned}$$

Since $\mathrm{char}(\mathbb{K}) = 2$, this sum equals

$$\left(\bigoplus_{j\in[N]} s\right) \cdot \alpha^t.$$

Applying π_t (Definition 2.2), and noting that $\pi_t(\alpha^t) = 1$ and $\pi_t(0) = 0$, we obtain

$$\mathrm{SCE}_{N,i,t}(X) = \bigoplus_{j\in[N]} s = \#\mathrm{SAT}(\phi) \bmod 2.$$

Therefore, $\oplus$SAT reduces to deciding whether $\mathrm{SCE}_{N,i,t}(X) = 1$. Output the instance $(N, i, t, 1, X)$.

This mapping is computable in polynomial time in $|\phi|$ because:

- $N = 2^n$ is represented in binary using $O(n)$ bits,
- X can evaluate ϕ, compute $\kappa_{i,j}$ and its inverse using $\mathrm{poly}(k) = \mathrm{poly}(n)$ field arithmetic, and output $k = \Theta(n)$ bits.

Thus we have a deterministic many-one reduction $\oplus$SAT $\leq_m$ SCE-Dec. □

13.3 $\oplus$**P-completeness of** Π_{β_2} **on** Im(ProbeBit)

Theorem 13.3 ($\oplus$**P**-completeness on the ProbeBit promise family)**.** $\Pi_{\beta_2} \restriction_{\mathcal{I}_{\text{Probe}}}$ *is* $\oplus$**P**-*complete under deterministic many-one reductions.*

Proof. **Hardness.** By the standard fact that $\oplus$SAT is $\oplus$**P**-complete (citation placeholder), it suffices to reduce $\oplus$SAT to $\Pi_{\beta_2} \restriction_{\mathcal{I}_{\text{Probe}}}$.

By Lemma 13.2, $\oplus\text{SAT} \leq_m$ SCE-Dec. By Theorem 12.2, for any SCE-Dec instance (N, i, t, b, X),

$$(N, i, t, b, X) \in \text{SCE-Dec} \iff \Pi_{\beta_2}\big(\text{ProbeBit}(N, i, t, b, X)\big) = 1.$$

Therefore $\oplus\text{SAT} \leq_m \Pi_{\beta_2} \restriction_{\mathcal{I}_{\text{Probe}}}$.

Membership. On inputs promised to lie in $\mathcal{I}_{\text{Probe}}$, the input D contains (N, i, t, b, X) as header data (Definition 12.1). By Lemma 13.1, SCE-Dec $\in \oplus$**P**, hence there is a parity-NP machine deciding whether $\text{SCE}_{N,i,t}(X) = b$. By Theorem 12.2, this equals $\Pi_{\beta_2}(D)$. Therefore $\Pi_{\beta_2} \restriction_{\mathcal{I}_{\text{Probe}}} \in \oplus$**P**.

Combining hardness and membership yields $\oplus$**P**-completeness. □

Remark 13.4 (Scope note (promise))**.** *The theorem is restricted to the promise family* $\mathcal{I}_{\text{Probe}} =$ Im(ProbeBit)*. No claim is made about* $\oplus$**P**-*completeness on arbitrary local-oracle inputs.*

14 A One-Sided Randomized SAT Reduction on Im(ProbeBit)

14.1 Isolation lemma (standard external result)

Lemma 14.1 (Valiant–Vazirani isolation lemma — standard)**.** *There exists a randomized polynomial-time procedure* $\mathcal{I}$ *that, given a Boolean formula* ϕ*, outputs formulas* $\psi_1, \ldots, \psi_m$ *for* $m = \text{poly}(|\phi|)$ *such that:*

1. *If* ϕ *is unsatisfiable, then every* ψ_r *is unsatisfiable.*
2. *If* ϕ *is satisfiable, then with probability at least* $1/\text{poly}(|\phi|)$*, at least one* ψ_r *has exactly one satisfying assignment.*

Status: treated as standard; citation placeholder (Valiant–Vazirani, 1986).

14.2 SAT $\leq_{rp} \Pi_{\beta_2}$ **on** Im(ProbeBit)

Recall our definition of a one-sided randomized many-one reduction $A \leq_{rp} B$ from Definition 2.16.

Theorem 14.2 (One-sided randomized many-one reduction SAT $\leq_{rp} \Pi_{\beta_2} \restriction_{\mathcal{I}_{\text{Probe}}}$)**.** *There exists a one-sided randomized many-one reduction*

$$\text{SAT} \ \leq_{rp} \ \Pi_{\beta_2} \restriction_{\mathcal{I}_{\text{Probe}}} .$$

Consequently, if Π_{β_2} *on* $\mathcal{I}_{\text{Probe}}$ *were solvable in deterministic polynomial time, then* **NP** $\subseteq$ **RP***.*

Proof. Given input formula ϕ, run the isolation procedure (Lemma 14.1) to obtain $\psi_1, \ldots, \psi_m$ with $m = \text{poly}(|\phi|)$.

For each ψ_r, apply the deterministic reduction from Lemma 13.2 to build an SCE-Dec instance $(N_r, i, t, 1, X_r)$ such that

$$\text{SCE}_{N_r,i,t}(X_r) = 1 \iff \#\text{SAT}(\psi_r) \equiv 1 \pmod 2.$$

If ψ_r has exactly one satisfying assignment, then $\#\mathrm{SAT}(\psi_r) \equiv 1 \pmod 2$, hence the SCE bit equals 1, and by Theorem 12.2,

$$\beta_2\Big(K_{\mathrm{ProbeBit}(N_r,i,t,1,X_r)}\Big) > 0.$$

To make a many-one reduction (single output instance), define D to be a local-oracle description of the disjoint union

$$K := \bigsqcup_{r=1}^{m} K_r, \qquad K_r := K_{\mathrm{ProbeBit}(N_r,i,t,1,X_r)}.$$

(Disjoint union preserves bounded degree and admits a local-oracle description by adding a component selector in the vertex/edge/triangle IDs; this is routine and does not affect any homology statements.)

Since homology over $\mathbb{F}_2$ is additive over disjoint unions,

$$\beta_2(K;\mathbb{F}_2) = \sum_{r=1}^{m} \beta_2(K_r;\mathbb{F}_2),$$

hence $\beta_2(K) > 0$ iff there exists r with $\beta_2(K_r) > 0$. Therefore:

- If ϕ is unsatisfiable, then each ψ_r is unsatisfiable, so each K_r has $\beta_2(K_r) = 0$, hence $\beta_2(K) = 0$. The reduction outputs a NO-instance with probability 1 (one-sided).
- If ϕ is satisfiable, then with probability at least $1/\mathrm{poly}(|\phi|)$, some ψ_r has exactly one satisfying assignment, hence the corresponding K_r has $\beta_2(K_r) > 0$, so $\beta_2(K) > 0$. Thus the reduction outputs a YES-instance with probability at least $1/\mathrm{poly}(|\phi|)$.

This is exactly a one-sided randomized many-one reduction $\mathrm{SAT} \le_{rp} \Pi_{\beta_2} \restriction_{\mathcal{I}_{\mathrm{Probe}}}$. □

Remark 14.3 (Scope note (promise))**.** *This reduction targets Π_{β_2} only on the promise family $\mathcal{I}_{\mathrm{Probe}}$ (and disjoint unions thereof, which remain within a straightforward promise closure of the family).*

15 An Unconditional Evaluation-Local Lower Bound for SCE

We now formalize the oracle model and prove an information-theoretic lower bound. This is unconditional (no complexity assumptions).

15.1 Model: evaluation-local algorithms

Definition 15.1 (Evaluation-local algorithm)**.** *Fix parameters (N,i,t). An algorithm $\mathcal{A}$ is* evaluation-local *if it receives as input (N,i,t) and oracle access to an evaluator $X : [N] \to \{0,1\}^k$, and it may query the oracle on indices $j \in [N]$ to obtain $X(j)$. It must output a bit intended to equal $\mathrm{SCE}_{N,i,t}(X)$.*

A deterministic evaluation-local algorithm makes at most q oracle queries if on every oracle X it queries $X(j)$ on at most q distinct indices j.

15.2 Lower bound $q \ge N$

Theorem 15.2 (Deterministic evaluation-local lower bound)**.** *Fix (N,i,t). Any deterministic evaluation-local algorithm that computes $\mathrm{SCE}_{N,i,t}(X)$ for all evaluators X must make at least N oracle queries in the worst case.*

Proof. Let $\mathcal{A}$ be deterministic and suppose it makes at most $N-1$ oracle queries. Consider any execution of $\mathcal{A}$ on some oracle; let $Q \subseteq [N]$ be the set of queried indices, with $|Q| \le N-1$. Choose an unqueried index

$$j^* \in [N] \setminus Q.$$

By Lemma 6.6 applied to $\kappa_{i,j^*} = (C_N)_{i,j^*} \neq 0$, the multiplication matrix $M(\kappa_{i,j^*})$ is invertible, hence its t-th row is nonzero. Therefore there exists $\ell^* \in [k]$ such that

$$m^{(N,i,t)}_{j^*,\ell^*} = 1$$

(Definition 6.7; this is exactly Remark 6.10).

Now define two evaluators X and X' as follows:

- For all queried indices $j \in Q$, set $X(j) = X'(j)$ (identical outputs).
- For all unqueried indices $j \notin Q$, set $X(j) = 0$ and $X'(j) = 0$, except at $j = j^*$, where:

$$X(j^*) = 0 \in \mathbb{F}_2^k, \qquad X'(j^*) \text{ equals } 0 \text{ except } x'_{j^*,\ell^*} = 1.$$

Then $\mathcal{A}$ receives identical oracle answers on all its queries under X and X', hence produces the same output on both.

However, by Lemma 6.8,

$$\begin{aligned}\mathrm{SCE}_{N,i,t}(X) \oplus \mathrm{SCE}_{N,i,t}(X') &= \bigoplus_{j,\ell} m_{j,\ell} \cdot (x_{j,\ell} \oplus x'_{j,\ell}) \\ &= m_{j^*,\ell^*} \cdot (0 \oplus 1) \\ &= 1,\end{aligned}$$

since all other bits agree, and $m_{j^*,\ell^*} = 1$. Thus $\mathrm{SCE}_{N,i,t}(X) \neq \mathrm{SCE}_{N,i,t}(X')$, contradicting correctness of $\mathcal{A}$ on both oracles.

Therefore any deterministic evaluation-local algorithm must query all N indices in the worst case. □

16 Deterministic Witness-Expansion: $\beta_2(K_\phi) = \#\mathrm{SAT}(\phi)$

This section presents a deterministic construction mapping a Boolean formula ϕ to a bounded-degree simplicial 2-complex K_ϕ given in the local-oracle model, such that

$$\beta_2(K_\phi; \mathbb{F}_2) = \#\mathrm{SAT}(\phi).$$

In particular, this yields deterministic SAT $\le_m \Pi_{\beta_2}$ and parsimonious #SAT $\le_m$ Compute-β_2.

A key technical constraint is that the local-oracle description requires explicit values (n_V, n_E, n_T). Hence the per-assignment gadget size must be independent of whether ϕ is satisfied (otherwise one would need to know $\#\mathrm{SAT}(\phi)$ just to compute n_T). We therefore use equal-size gadgets: a "YES gadget" with $\beta_2 = 1$ and a "NO gadget" with $\beta_2 = 0$, but both with the same counts $(|V|, |E|, |T|)$.

16.1 Two equal-size gadget blocks

All homology in this section is over $\mathbb{F}_2$.

16.1.1 A sphere gadget (triangulated S^2)

Definition 16.1 (Tetrahedral sphere complex S)**.** *Let $S = (V_S, E_S, T_S)$ be the simplicial 2-complex with:*

- $V_S := \{0, 1, 2, 3\}$,
- $E_S := \binom{V_S}{2}$ *(all 6 edges),*
- $T_S := \binom{V_S}{3}$ *(all 4 triangles).*

This is the boundary complex of a tetrahedron, hence a triangulation of S^2.

16.1.2 A padding forest (no 2-faces)

Definition 16.2 (Forest padding F)**.** *Let $F = (V_F, E_F, T_F)$ be the simplicial 2-complex with:*

- $V_F := \{4, 5, 6, 7, 8, 9, 10, 11\}$ *(8 vertices),*
- $E_F := \{\{4, 5\}, \{5, 6\}, \{6, 7\}, \{7, 8\}, \{8, 9\}, \{9, 10\}\}$ *(a length-6 path on $\{4, \dots, 10\}$),*
- $T_F := \varnothing$,

and vertex 11 isolated.

Clearly F is 1-dimensional (no triangles), hence $\beta_2(F) = 0$.

16.1.3 The YES-block and NO-block (same (V, E, T) sizes)

Definition 16.3 (YES-block $G^{(1)}$)**.** *Define the YES-block*

$$G^{(1)} := S \sqcup F.$$

Then

$$|V(G^{(1)})| = 4 + 8 = 12, \quad |E(G^{(1)})| = 6 + 6 = 12, \quad |T(G^{(1)})| = 4 + 0 = 4.$$

Definition 16.4 (NO-block $G^{(0)}$)**.** *Define the NO-block $G^{(0)}$ as the disjoint union of four disjoint filled triangles:*

- *Vertices:* $V(G^{(0)}) := \{0, 1, \dots, 11\}$.
- *Triangles:* $T(G^{(0)}) := \{\{0, 1, 2\}, \{3, 4, 5\}, \{6, 7, 8\}, \{9, 10, 11\}\}$.
- *Edges:* $E(G^{(0)})$ *are the union of the 3 edges of each triangle.*

Then $|V(G^{(0)})| = 12$, $|E(G^{(0)})| = 12$, $|T(G^{(0)})| = 4$.

Lemma 16.5 (Bounded degree and equal-size property)**.** *Both $G^{(1)}$ and $G^{(0)}$ are bounded-degree simplicial 2-complexes with maximum vertex degree ≤ 3. Moreover,*

$$(|V|, |E|, |T|) \text{ is the same for } G^{(1)} \text{ and } G^{(0)}, \text{ namely } (12, 12, 4).$$

Proof. Equal-size counts were verified in Definitions 16.3–16.4. For bounded degree:

- In S, every vertex is incident to exactly 3 edges and 3 triangles.
- In F, path vertices have degree ≤ 2, and vertex 11 has degree 0; no triangles.
- In $G^{(0)}$, each triangle vertex is incident to 2 edges and 1 triangle.

Disjoint union does not increase local degrees. Thus the maximum degree is 3. □

16.2 Computing β_2 of the blocks

16.2.1 $\beta_2(S) = 1$

Lemma 16.6 ($\beta_2(S; \mathbb{F}_2) = 1$)**.** *For the tetrahedral sphere complex S from Definition 16.1, $\beta_2(S; \mathbb{F}_2) = 1$.*

Proof. In S, we have $|T_S| = 4$ and $|E_S| = 6$, so $C_2(S) \cong \mathbb{F}_2^4$ and $C_1(S) \cong \mathbb{F}_2^6$. Since there are no 3-simplices, $H_2(S) = \ker(\partial_2)$.

Let the four triangles be

$$\tau_0 = \{0, 1, 2\}, \quad \tau_1 = \{0, 1, 3\}, \quad \tau_2 = \{0, 2, 3\}, \quad \tau_3 = \{1, 2, 3\}.$$

(1) Existence of a nonzero 2-cycle. Consider $c := \tau_0 \oplus \tau_1 \oplus \tau_2 \oplus \tau_3 \in C_2(S)$. Every edge of S lies in exactly two of the τ_i. Therefore in $\partial_2(c)$, each edge appears with coefficient $2 \equiv 0 \pmod 2$, so $\partial_2(c) = 0$. Hence $c \in H_2(S)$, and $\dim H_2(S) \geq 1$.

(2) $\dim H_2(S) \leq 1$**.** It suffices to show $\operatorname{rank}(\partial_2) \geq 3$, because then

$$\dim \ker(\partial_2) = \dim C_2 - \operatorname{rank}(\partial_2) \leq 4 - 3 = 1.$$

Consider the boundaries of τ_0, τ_1, τ_2. Each contains a unique edge not present in the other two:

- Edge $\{1, 2\}$ appears in $\partial_2(\tau_0)$ but not in $\partial_2(\tau_1)$ or $\partial_2(\tau_2)$.
- Edge $\{1, 3\}$ appears in $\partial_2(\tau_1)$ but not in $\partial_2(\tau_0)$ or $\partial_2(\tau_2)$.
- Edge $\{2, 3\}$ appears in $\partial_2(\tau_2)$ but not in $\partial_2(\tau_0)$ or $\partial_2(\tau_1)$.

Therefore $\partial_2(\tau_0)$, $\partial_2(\tau_1)$, $\partial_2(\tau_2)$ are linearly independent in $C_1(S)$, so $\operatorname{rank}(\partial_2) \geq 3$.

Combining (1) and (2), we obtain $\dim H_2(S) = 1$, hence $\beta_2(S) = 1$. □

16.2.2 β_2 of triangles and forests

Lemma 16.7 (β_2 of a single filled triangle is 0)**.** *Let D be the simplicial complex consisting of one triangle $\{a, b, c\}$ and its three edges and vertices. Then*

$$\beta_2(D; \mathbb{F}_2) = 0.$$

Proof. Here $C_2(D) \cong \mathbb{F}_2$ is spanned by $\{a, b, c\}$, and

$$\partial_2(\{a, b, c\}) = \{a, b\} \oplus \{a, c\} \oplus \{b, c\} \neq 0$$

in $C_1(D)$. Hence $\ker(\partial_2) = \{0\}$, so $H_2(D) = 0$ and $\beta_2(D) = 0$. □

Lemma 16.8 ($\beta_2(F; \mathbb{F}_2) = 0$)**.** *For the forest padding F (Definition 16.2), $\beta_2(F) = 0$.*

Proof. F has no triangles, so $C_2(F) = 0$, hence $H_2(F) = 0$. □

Corollary 16.9 (β_2 of the blocks)**.**

$$\beta_2(G^{(1)}; \mathbb{F}_2) = 1, \qquad \beta_2(G^{(0)}; \mathbb{F}_2) = 0.$$

Proof. By Lemma 16.6 and Lemma 16.8, $\beta_2(S) = 1$ and $\beta_2(F) = 0$. By additivity over disjoint union (Lemma 16.10 below), $\beta_2(G^{(1)}) = \beta_2(S) + \beta_2(F) = 1$.

For $G^{(0)}$, it is the disjoint union of four single triangles, each with $\beta_2 = 0$ by Lemma 16.7; by additivity, $\beta_2(G^{(0)}) = 0$. □

16.3 Additivity of homology under disjoint unions

Lemma 16.10 (Homology of disjoint unions). *Let $K = K_1 \sqcup K_2$ be the disjoint union of simplicial 2-complexes. Then for each $q \in \{0, 1, 2\}$,*

$$H_q(K; \mathbb{F}_2) \;\cong\; H_q(K_1; \mathbb{F}_2) \;\oplus\; H_q(K_2; \mathbb{F}_2).$$

Proof. Since K_1 and K_2 are disjoint (no shared simplices), the chain groups split:

$$C_q(K) \cong C_q(K_1) \oplus C_q(K_2)$$

with respect to the basis of simplices. The boundary maps are block-diagonal:

$$\partial_q^K = \partial_q^{K_1} \oplus \partial_q^{K_2}.$$

Therefore $\ker(\partial_q^K) = \ker(\partial_q^{K_1}) \oplus \ker(\partial_q^{K_2})$ and similarly for images. Taking quotients yields the direct sum decomposition on homology. □

Corollary 16.11 (Additivity of β_2). *For disjoint unions, $\beta_2(K_1 \sqcup K_2) = \beta_2(K_1) + \beta_2(K_2)$.*

16.4 The witness-expansion complex K_ϕ

Let ϕ be a Boolean formula on n variables $y_0, \dots, y_{n-1}$. We identify assignments with integers in $[2^n]$.

Definition 16.12 (Assignment decoding). *For $a \in [2^n]$, define* $\mathrm{asgn}(a) \in \{0, 1\}^n$ *as the length-n binary expansion of a, i.e.,* $\mathrm{asgn}(a) = (a_0, \dots, a_{n-1})$ *with*

$$a = \sum_{j=0}^{n-1} a_j 2^j.$$

Definition 16.13 (Witness-expansion complex K_ϕ). *Let $M := 2^n$. For each $a \in [M]$, define a block*

$$\mathcal{C}_a := \begin{cases} G^{(1)} & \text{if } \phi(\mathrm{asgn}(a)) = 1, \\ G^{(0)} & \text{if } \phi(\mathrm{asgn}(a)) = 0. \end{cases}$$

Define

$$K_\phi := \bigsqcup_{a \in [M]} \mathcal{C}_a.$$

By Lemma 16.5, each $\mathcal{C}_a$ has bounded degree (at most 3), hence K_ϕ is also bounded-degree with the same constant bound.

16.5 Parsimonious equality $\beta_2(K_\phi) = \#\mathrm{SAT}(\phi)$

Theorem 16.14 (Parsimonious equality). *For every Boolean formula ϕ,*

$$\beta_2(K_\phi; \mathbb{F}_2) = \#\mathrm{SAT}(\phi).$$

Proof. By Corollary 16.9, each block $\mathcal{C}_a$ contributes

$$\beta_2(\mathcal{C}_a) = \begin{cases} 1 & \text{if } \phi(\text{asgn}(a)) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

By additivity over disjoint unions (Corollary 16.11),

$$\beta_2(K_\phi) = \sum_{a \in [M]} \beta_2(\mathcal{C}_a) = \sum_{a \in [M]} \mathbf{1}[\phi(\text{asgn}(a)) = 1] = \#\text{SAT}(\phi).$$

□

Corollary 16.15 (Deterministic SAT reduction)**.** *Let D_ϕ be a local-oracle description of K_ϕ. Then*

$$\phi \in \text{SAT} \iff \beta_2(K_{D_\phi}; \mathbb{F}_2) > 0 \iff \Pi_{\beta_2}(D_\phi) = 1.$$

Proof. ϕ is satisfiable iff $\#\text{SAT}(\phi) > 0$. By Theorem 16.14, $\#\text{SAT}(\phi) = \beta_2(K_\phi)$. Thus $\#\text{SAT}(\phi) > 0 \iff \beta_2(K_\phi) > 0$. □

Corollary 16.16 (Parsimonious #SAT reduction)**.** *The mapping $\phi \mapsto K_\phi$ satisfies*

$$\#\text{SAT}(\phi) = \beta_2(K_\phi; \mathbb{F}_2).$$

Hence $\#\text{SAT} \leq_m \text{Compute-}\beta_2$.

16.6 Local-oracle description D_ϕ for K_ϕ

We now spell out a local-oracle description for K_ϕ consistent with Definition 2.10, including explicit global indexing.

16.6.1 Global sizes

Each block $\mathcal{C}_a$ has $(12, 12, 4)$ simplices in dimensions $0, 1, 2$. Therefore:

Definition 16.17 (Global sizes for K_ϕ)**.** *Let $M := 2^n$. Define*

$$n_V := 12M, \qquad n_E := 12M, \qquad n_T := 4M.$$

16.6.2 Global indexing scheme

Definition 16.18 (Component-local decoding)**.** *For a global vertex index $v \in [n_V]$, define:*

$$a(v) := \left\lfloor \frac{v}{12} \right\rfloor \in [M], \qquad r(v) := v \bmod 12 \in [12].$$

Similarly for global edge indices $e \in [n_E]$ and triangle indices $\tau \in [n_T]$,

$$a(e) := \left\lfloor \frac{e}{12} \right\rfloor, \quad s(e) := e \bmod 12, \qquad a(\tau) := \left\lfloor \frac{\tau}{4} \right\rfloor, \quad q(\tau) := \tau \bmod 4.$$

Thus each vertex/edge/triangle index decodes to a component ID a and a local index within the 12-vertex, 12-edge, 4-triangle block.

16.6.3 Local tables for the two block types

Definition 16.19 (YES-block local edge endpoints and triangle vertices)**.** *In $G^{(1)}$:*

- *Local edges $s \in [12]$ are:*
  - *Sphere edges (local $s = 0, \dots, 5$):*

$$0 : (0, 1),\ 1 : (0, 2),\ 2 : (0, 3),\ 3 : (1, 2),\ 4 : (1, 3),\ 5 : (2, 3).$$

  - *Forest edges (local $s = 6, \dots, 11$):*

$$6 : (4, 5),\ 7 : (5, 6),\ 8 : (6, 7),\ 9 : (7, 8),\ 10 : (8, 9),\ 11 : (9, 10).$$

- *Local triangles $q \in [4]$ are the sphere triangles:*

$$0 : (0, 1, 2),\ 1 : (0, 1, 3),\ 2 : (0, 2, 3),\ 3 : (1, 2, 3).$$

Definition 16.20 (NO-block local edge endpoints and triangle vertices)**.** *In $G^{(0)}$, for each $q \in [4]$, the q-th triangle uses vertices $(3q, 3q + 1, 3q + 2)$. Define:*

- *Local triangles:*

$$q : (3q, 3q + 1, 3q + 2).$$

- *Local edges $s \in [12]$: write $q = \lfloor s/3 \rfloor$ and $r = s \bmod 3$. Then*

$$r = 0 :\ (3q, 3q + 1), \quad r = 1 :\ (3q, 3q + 2), \quad r = 2 :\ (3q + 1, 3q + 2).$$

16.6.4 Oracles

Let D_ϕ be the description consisting of (n_V, n_E, n_T) and circuits implementing the oracles below. We emphasize that the oracle computations use ϕ as a subroutine (compiled into a circuit) to decide whether block a is a YES-block or NO-block.

Definition 16.21 (Oracle End_{D_ϕ})**.** *Given edge index $e \in [n_E]$:*

1. *Decode $(a, s) = (a(e), s(e))$ (Definition 16.18).*
2. *Compute $b := \phi(\mathrm{asgn}(a)) \in \{0, 1\}$ (Definition 16.12).*
3. *If $b = 1$, use Definition 16.19 to compute local endpoints $(u, v) \in [12]^2$. If $b = 0$, use Definition 16.20 to compute local endpoints $(u, v) \in [12]^2$.*
4. *Output global endpoints*

$$\mathrm{End}_{D_\phi}(e) := (12a + u,\ 12a + v) \in [n_V]^2.$$

Definition 16.22 (Oracle Vert_{D_ϕ})**.** *Given triangle index $\tau \in [n_T]$:*

1. *Decode $(a, q) = (a(\tau), q(\tau))$.*
2. *Compute $b := \phi(\mathrm{asgn}(a))$.*
3. *If $b = 1$, output local triangle vertices from Definition 16.19; if $b = 0$, output local triangle vertices from Definition 16.20.*

4. *Offset by* $12a$ *to get global vertices.*

Definition 16.23 (Incidence oracles IncE_{D_ϕ} and IncT_{D_ϕ})**.** *Fix* $\Delta := 3$*. For a vertex index* $v \in [n_V]$ *and* $\ell \in [\Delta]$*:*

1. *Decode* $(a, r) = (a(v), r(v))$*.*

2. *Compute* $b := \phi(\mathrm{asgn}(a))$*.*

3. *If* $b = 1$*, return the* ℓ*-th incident edge/triangle to local vertex* r *according to the YES-block tables implied by Definition 16.19. If* $b = 0$*, return the* ℓ*-th incident edge/triangle to local vertex* r *according to the NO-block structure in Definition 16.20.*

4. *If fewer than* Δ *incident edges/triangles exist, return* $\perp$ *in the remaining slots.*

5. *Any returned local edge/triangle index is offset to a global ID by adding* $12a$ *(for edges) or* $4a$ *(for triangles).*

We omit the full local incidence tables for the YES-block in the main text; they are finite and can be listed verbatim (and are uniquely determined by Definitions 16.19–16.20). The NO-block incidence lists are computable by constant-time arithmetic from (r, ℓ)*.*

Lemma 16.24 (Validity, bounded degree, and size of D_ϕ)**.** *The description* D_ϕ *defined above is a valid local-oracle description (Definition 2.10) of the simplicial* 2*-complex* K_ϕ*. Moreover:*

1. K_{D_ϕ} *has bounded degree* $\Delta = 3$*.*

2. *The description length* $|D_\phi|$ *is polynomial in* $|\phi|$*.*

3. *Each oracle query runs in time* $\mathrm{poly}(|\phi|)$*.*

Proof. **Validity.** By construction, K_ϕ is a disjoint union of M blocks, each of which is either $G^{(1)}$ or $G^{(0)}$. The oracles End_{D_ϕ} and Vert_{D_ϕ} return exactly the endpoints/vertices of the edges/triangles listed in Definitions 16.19–16.20, shifted into the component a. Closure holds within each block by construction (each triangle's edges are included). Incidence lists are consistent by construction because they are derived from the same local tables.

Bounded degree. Lemma 16.5 gives degree ≤ 3 within each block; disjoint union preserves it.

Size/time. The description stores ϕ (or an equivalent circuit) plus fixed-size wiring for the decoding arithmetic and local tables. Computing $\phi(\mathrm{asgn}(a))$ is polynomial in $|\phi|$, and all other work is constant-time arithmetic on indices of length $O(n) \leq O(|\phi|)$. Thus $|D_\phi| = \mathrm{poly}(|\phi|)$, and each oracle query runs in $\mathrm{poly}(|\phi|)$ time. □

16.7 Complexity consequences (deterministic, non-promise)

Theorem 16.25 (Deterministic many-one reduction $\mathrm{SAT} \leq_m \Pi_{\beta_2}$)**.** *There is a deterministic polynomial-time many-one reduction* f *mapping formulas* ϕ *to local-oracle descriptions* D_ϕ *such that*

$$\phi \in \mathrm{SAT} \iff \Pi_{\beta_2}(D_\phi) = 1.$$

Proof. Take $f(\phi) := D_\phi$ as constructed above. By Corollary 16.15 and Lemma 16.24,

$$\phi \in \mathrm{SAT} \iff \beta_2(K_{D_\phi}) > 0 \iff \Pi_{\beta_2}(D_\phi) = 1.$$

The mapping is polynomial-time because the output description size is polynomial in $|\phi|$. □

Theorem 16.26 (Parsimonious reduction #SAT $\leq_m$ Compute-β_2)**.** *There is a deterministic polynomial-time many-one reduction from* #SAT *to* Compute-β_2 *satisfying*

$$\#\mathrm{SAT}(\phi) = \beta_2(K_{D_\phi}; \mathbb{F}_2).$$

Proof. Immediate from Theorem 16.14 and Lemma 16.24. □

Important limitation (explicitly stated). Theorem 16.25 shows NP-hardness of Π_{β_2} under deterministic many-one reductions in the local-oracle model. It does not establish that $\Pi_{\beta_2} \in \mathbf{NP}$ (or any particular upper bound class) because, under succinct representations, certificates for $\beta_2 > 0$ (e.g., an explicit nonzero 2-cycle) may be exponentially large in $|D|$. This paper does not prove an NP upper bound for Π_{β_2}.

17 Unconditional Circuit Lower Bounds on an Explicit Parity-Based Subfamily

This section shows how to derive unconditional lower bounds against certain nonuniform circuit/formula classes by exhibiting an explicit parity-based restriction of Π_{β_2} that computes PARITY.

17.1 A bounded linear system for parity

Let PARITY : $\{0,1\}^n \to \{0,1\}$ be PARITY$(x) = x_0 \oplus \cdots \oplus x_{n-1}$.

We build, from x, a bounded-arity bounded-occurrence homogeneous system $\mathsf{SysPar}(x)$ such that it has a nonzero solution iff PARITY$(x) = 1$.

Definition 17.1 (Parity system $\mathsf{SysPar}(x)$)**.** *Fix $n \geq 1$ and $x \in \{0,1\}^n$. Define* $\mathsf{SysPar}(x)$ *as a homogeneous $\mathbb{F}_2$-linear system with variables:*

- *a switch λ_{root},*
- *copies $\lambda_0, \ldots, \lambda_{n-1}$,*
- *gated bits $u_0, \ldots, u_{n-1}$,*
- *a sum variable S,*
- *and internal variables of EqTree and XorTree gadgets.*

Equations:

1. EqTree$(\lambda_{\mathrm{root}}; \lambda_0, \ldots, \lambda_{n-1})$.
2. *For each $j \in [n]$, the gating equation:*
    - *if $x_j = 0$, include $u_j = 0$,*
    - *if $x_j = 1$, include $u_j \oplus \lambda_j = 0$.*
3. XorTree$(S;\, u_0, \ldots, u_{n-1})$.
4. *Final check $S \oplus \lambda_{\mathrm{root}} = 0$.*

All equations have arity ≤ 3. Variable occurrences are bounded by an absolute constant if the trees are chosen as rooted binary trees (as in Part 2).

Lemma 17.2 (Correctness of $\mathsf{SysPar}(x)$)**.** *$\mathsf{SysPar}(x)$ has a nonzero solution if and only if* $\mathrm{PARITY}(x) = 1$.

Proof. The proof is the same "switch" structure as Lemma 8.2 and Theorem 8.4, specialized to a 1-bit XOR.

- If $\lambda_{\mathrm{root}} = 0$, then EqTree forces $\lambda_j = 0$ for all j. Then each gating equation forces $u_j = 0$ (either directly if $x_j = 0$, or because $u_j = \lambda_j = 0$ if $x_j = 1$). Then XorTree forces $S = \bigoplus_j u_j = 0$. The final check $S \oplus \lambda_{\mathrm{root}} = 0$ holds. Hence the all-zero assignment is a solution.

- Consider a solution with $\lambda_{\mathrm{root}} = 1$. EqTree forces $\lambda_j = 1$ for all j. Then each gating equation sets

$$u_j = \begin{cases} 0 & \text{if } x_j = 0, \\ 1 & \text{if } x_j = 1, \end{cases}$$

  i.e., $u_j = x_j$. XorTree enforces $S = \bigoplus_j u_j = \bigoplus_j x_j = \mathrm{PARITY}(x)$. The final check $S \oplus \lambda_{\mathrm{root}} = 0$ becomes $S = 1$, hence requires $\mathrm{PARITY}(x) = 1$.

Therefore, a nonzero solution exists (equivalently, a solution with $\lambda_{\mathrm{root}} = 1$ exists) iff $\mathrm{PARITY}(x) = 1$. □

17.2 Mapping parity systems to Π_{β_2} instances

We use the system-to-topology mapping already established:

- Part 3: $K^{\mathrm{cw}}(\mathsf{Sys})$ with $H_2 \cong \mathsf{Sol}(\mathsf{Sys})$.

- Part 4: simplicialization $K(\mathsf{Sys})$ preserving H_2.

Definition 17.3 (Parity-to-Π_{β_2} map $\Phi_\oplus$)**.** *For $x \in \{0,1\}^n$, define $\Phi_\oplus(x)$ to be a local-oracle description of the simplicial complex*

$$K(\mathsf{SysPar}(x)).$$

(Concretely, $\Phi_\oplus(x)$ is obtained by applying the constructions of Sections 9–11 to $\mathsf{SysPar}(x)$*, then encoding the resulting bounded-degree simplicial complex by local-oracles.)*

Theorem 17.4 (Parity realized by Π_{β_2} on an explicit subfamily)**.** *For all $x \in \{0,1\}^n$,*

$$\Pi_{\beta_2}(\Phi_\oplus(x)) \;=\; \mathrm{PARITY}(x).$$

Proof. By Corollary 11.14, for any bounded system Sys,

$$\beta_2(K(\mathsf{Sys}); \mathbb{F}_2) > 0 \iff \mathsf{Sys} \text{ has a nonzero solution.}$$

Applying this to $\mathsf{SysPar}(x)$, we get

$$\Pi_{\beta_2}(\Phi_\oplus(x)) = 1 \iff \mathsf{SysPar}(x) \text{ has a nonzero solution.}$$

By Lemma 17.2, this is equivalent to $\mathrm{PARITY}(x) = 1$. □

17.3 Projection/$\mathbf{AC}^0$ nature of $\Phi_\oplus$: encoding-dependent and isolated

To translate Theorem 17.4 into nonuniform circuit lower bounds, we need a precise statement about how x is embedded into the bitstring encoding of $\Phi_\oplus(x)$. This is encoding-dependent, so we isolate it.

Definition 17.5 (Projection reduction; encoding-dependent)**.** *Fix a concrete bit-level encoding scheme* $\mathrm{Enc}(\cdot)$ *for local-oracle descriptions D. A family of maps*

$$\rho_n : \{0,1\}^n \to \{0,1\}^{m(n)}$$

is a projection *if every output bit $(\rho_n(x))_k$ is either a constant in $\{0,1\}$, or equals x_j, or equals $\neg x_j$, for some $j \in [n]$.*

Lemma 17.6 (Projection property of $\Phi_\oplus$; proof in Appendix C)**.** *Under the explicit encoding convention* Enc *fixed in Appendix C, the mapping*

$$x \mapsto \mathrm{Enc}(\Phi_\oplus(x))$$

is a projection $\{0,1\}^n \to \{0,1\}^{m(n)}$ with $m(n) = \Theta(n)$.

Proof status: *proved in Appendix C.*

Comment: *The construction of* $\mathsf{SysPar}(x)$ *differs across x only in the local choice of which of two fixed gate patterns appears in each gating constraint; the encoding is chosen so that these choices correspond to designated bits equal to x_j (or $\neg x_j$), while all other description bits are fixed constants.*

17.4 Unconditional lower bounds (standard parity lower bounds; citation placeholders)

We state three consequences. Each uses a standard lower bound for PARITY as a black box and therefore requires a citation placeholder.

17.4.1 $\mathbf{AC}^0$ lower bound

Theorem 17.7 (Unconditional $\mathbf{AC}^0$ lower bound for Π_{β_2} as a promise problem)**.** *There is no nonuniform $\mathbf{AC}^0$ circuit family $\{C_m\}$ that correctly computes $\Pi_{\beta_2}(D)$ on all valid local-oracle descriptions D of bounded-degree simplicial 2-complexes (in the sense of Definition 2.10 and the chosen encoding).*

Proof. Assume for contradiction that such an $\mathbf{AC}^0$ family exists. Fix n. Consider the restriction of $C_{m(n)}$ to the set

$$\mathcal{F}_n := \{\mathrm{Enc}(\Phi_\oplus(x)) : x \in \{0,1\}^n\},$$

which consists entirely of valid instances by construction.

By Lemma 17.6, $\mathrm{Enc}(\Phi_\oplus(x))$ is obtained from x by a projection ρ_n. Substituting the projection wires into $C_{m(n)}$ yields an $\mathbf{AC}^0$ circuit for PARITY(x), because by Theorem 17.4 the output agrees with PARITY(x) on all inputs x.

This contradicts the standard result PARITY $\notin \mathbf{AC}^0$ (citation placeholder). Hence no such $\mathbf{AC}^0$ family exists. □

17.4.2 De Morgan formula lower bound

Theorem 17.8 (Unconditional De Morgan formula lower bound for Π_{β_2} on an explicit subfamily)**.** *Let $\mathcal{F}_n$ be as above. Any De Morgan formula that correctly computes Π_{β_2} on $\mathcal{F}_n$ must have size $\Omega(n^2)$.*

Proof. By Theorem 17.4 and Lemma 17.6, Π_{β_2} restricted to $\mathcal{F}_n$ computes PARITY under a projection. Projection substitutions do not increase formula size by more than a constant factor. Therefore any formula computing Π_{β_2} on $\mathcal{F}_n$ yields a formula computing PARITY on $\{0,1\}^n$ of comparable size. By the standard quadratic lower bound for parity in De Morgan formulas (e.g., Khrapchenko-type bounds; citation placeholder), the size must be $\Omega(n^2)$. □

17.4.3 $\mathbf{AC}^0[p]$ lower bound for odd primes p

Theorem 17.9 (Unconditional $\mathbf{AC}^0[p]$ lower bound for odd primes)**.** *Fix an odd prime p. There is no nonuniform $\mathbf{AC}^0[p]$ circuit family that correctly computes Π_{β_2} on all valid instances (in the same sense as Theorem 17.7). In particular, Π_{β_2} restricted to $\mathcal{F}_n$ is not in $\mathbf{AC}^0[p]$.*

Proof. The proof is identical to Theorem 17.7, replacing $\mathbf{AC}^0$ by $\mathbf{AC}^0[p]$ and using the standard result that PARITY $\notin \mathbf{AC}^0[p]$ for odd primes p (Razborov–Smolensky-type lower bounds; citation placeholder). □

Remark 17.10 (Scope note (explicit))**.** *Theorems 17.7–17.9 are unconditional but rely on standard external lower bounds for* PARITY*. They are statements about nonuniform circuit/formula families computing Π_{β_2} correctly on the promise set of valid encodings, which is the appropriate notion for a promise problem.*

18 Discussion, Limitations, and Open Issues

18.1 Summary of the validated contributions

The paper establishes the following validated core results:

1. **Algebraic linearization of SCE.** The SCE output bit $\mathrm{SCE}_{N,i,t}(X)$ is an explicit $\mathbb{F}_2$-linear form in evaluator bits (Part 2, Lemma 6.8).

2. **Bounded-occurrence system encoding.** The system $\mathsf{SysBit}(N,i,t,b,X)$ has a nonzero solution iff $\mathrm{SCE}_{N,i,t}(X) = b$ (Part 2, Theorem 8.4), with constant arity and occurrence (Lemma 8.5).

3. **Topological encoding of linear systems.** For a bounded system Sys, the CW complex $K^{\mathrm{cw}}(\mathsf{Sys})$ satisfies $H_2(K^{\mathrm{cw}}(\mathsf{Sys}); \mathbb{F}_2) \cong \mathsf{Sol}(\mathsf{Sys})$ (Part 3, Theorem 10.9). After a careful simplicialization via triangular subdivision and barycentric subdivision, homology is preserved (Part 4, Corollary 11.14).

4. **ProbeBit pipeline results (promise-family).** The map ProbeBit yields an equivalence $\beta_2 > 0 \iff \mathrm{SCE} = b$ (Part 4, Theorem 12.2), leading to $\oplus\mathbf{P}$-completeness and SAT $\leq_{rp}$ on the promise family Im(ProbeBit) (Part 4, Theorems 13.3 and 14.2), and a deterministic evaluation-local query lower bound $q \geq N$ for SCE (Part 4, Theorem 15.2).

5. **Witness-expansion deterministic reductions (non-promise).** The deterministic construction K_ϕ satisfies $\beta_2(K_\phi) = \#\mathrm{SAT}(\phi)$, hence SAT $\leq_m \Pi_{\beta_2}$ and #SAT $\leq_m$ Compute-β_2 (Part 5, Theorems 16.25 and 16.26).

6. **Unconditional circuit lower bounds via parity subfamilies.** Using $\mathsf{SysPar}(x)$ and the established system-to-topology mapping, we obtain an explicit parity-based restriction of Π_{β_2} and unconditional lower bounds against $\mathbf{AC}^0$, De Morgan formulas, and $\mathbf{AC}^0[p]$ (Part 5, Theorems 17.7–17.9), modulo standard parity lower bounds (citation placeholders).

18.2 Why these results do not resolve P vs NP

Theorem 16.25 shows NP-hardness of Π_{β_2} in the local-oracle model, and Theorem 13.3 gives $\oplus\mathbf{P}$-completeness on a promise family. Neither implies $\mathbf{P} \neq \mathbf{NP}$, because:

- NP-hardness does not contradict $\mathbf{P} = \mathbf{NP}$; if $\mathbf{P} = \mathbf{NP}$, NP-hard problems can still be in $\mathbf{P}$.
- $\oplus\mathbf{P}$-completeness on a promise family does not by itself yield a classical separation either.
- The paper does not establish $\Pi_{\beta_2} \in \mathbf{NP}$, which would be needed even to discuss NP-completeness in the usual total-function sense.

Thus, the results should be read as evidence of hardness phenomena for succinct topological invariants, not as a separation.

18.3 Promise-family hardness vs unrestricted deterministic hardness

There is a deliberate separation:

- ProbeBit results are strongest in parity-counting terms ($\oplus\mathbf{P}$-completeness), but they are stated on Im(ProbeBit), a promise family.
- Witness-expansion yields an unrestricted deterministic reduction from SAT, but does not directly yield $\oplus\mathbf{P}$-completeness statements.

18.4 Remaining technical dependencies and clearly identified gaps

This paper maintains strict marking of dependencies:

- **Standard external results (citation placeholders):**
  - Valiant–Vazirani isolation lemma (used in Theorem 14.2).
  - $\oplus$SAT is $\oplus\mathbf{P}$-complete (used in Theorem 13.3).
  - PARITY lower bounds for $\mathbf{AC}^0$, formulas, and $\mathbf{AC}^0[p]$ (used in Theorems 17.7–17.9).
- **Encoding-dependent steps (proved or deferred):**
  - The “projection property” of the parity subfamily is explicitly proved in Appendix C (Lemma 17.6).
  - For ProbeBit, the fully explicit bit-level encoding of local-oracle circuits is nontrivial; Part 4’s Lemma 12.3 is a construction argument with finite casework, and full mechanical encoding details can be expanded further if a completely formal machine-checked implementation is required.

18.5 Open problems suggested by the results

1. **Upper bounds for Π_{β_2} in the local-oracle model.** Determine whether Π_{β_2} lies in **NP**, co**NP**, Σ_2^P, PSPACE, etc., under this input model (requires careful modeling of validity promises and witnesses).

2. **Promise elimination or promise robustness.** For the ProbeBit pipeline, identify natural syntactic constraints under which the promise family can be recognized (or made canonical), enabling sharper "non-promise" complexity statements.

3. **Derandomization of the** SAT $\leq_{rp}$ **route.** The isolation-based route is randomized. Deterministic analogues would require additional derandomization ingredients beyond this paper's scope.

4. **Alternative invariants and dimensions.** Extend the framework to other homological invariants or to other dimensions, clarifying what aspects are specific to 2-complexes and mod-2 coefficients.

Appendix B. Boolean Circuits for $\mathbb{F}_{2^k}$ Arithmetic (Supplementary)

This appendix supports Remark 2.4 and Lemma 6.9 by giving explicit polynomial-size circuit constructions for basic $\mathbb{F}_{2^k}$ arithmetic under a fixed irreducible polynomial basis. The statements here are "internal completeness" results; they are not the main novelty of the paper.

Lemma 18.1 (Addition and multiplication in $\mathbb{F}_{2^k}$ have polynomial-size Boolean circuits)**.** *Fix an irreducible polynomial $p_k(z) \in \mathbb{F}_2[z]$ of degree k and represent $\mathbb{F}_{2^k} \cong \mathbb{F}_2[z]/(p_k)$ using the polynomial basis $\{1, \alpha, \dots, \alpha^{k-1}\}$. Then:*

- *Addition is computed by k XOR gates (size $O(k)$).*
- *Multiplication is computed by a Boolean circuit of size $O(k^2)$.*

Proof. Addition is coordinate-wise XOR.

For multiplication, represent field elements as polynomials of degree $< k$ with coefficients in $\mathbb{F}_2$. Compute the polynomial product (degree $< 2k$) by convolution: each output coefficient is an XOR of ANDs of input bits; this uses $O(k^2)$ AND gates and $O(k^2)$ XOR gates. Then reduce modulo p_k: since p_k is fixed for the given k, reduction is an $\mathbb{F}_2$-linear map from $\mathbb{F}_2^{2k-1}$ to $\mathbb{F}_2^k$, implemented by $O(k^2)$ XOR gates. Total size $O(k^2)$. □

Lemma 18.2 (Inversion has polynomial-size circuits)**.** *Under the same representation, inversion $u \mapsto u^{-1}$ on $\mathbb{F}_{2^k}^\times$ has polynomial-size Boolean circuits.*

Proof (explicit exponentiation). For nonzero u, $u^{-1} = u^{2^k-2}$ in $\mathbb{F}_{2^k}$. Compute u^{2^k-2} by repeated squaring and multiplication:

- Squaring in characteristic two is $\mathbb{F}_2$-linear and can be implemented by a linear circuit of size $O(k^2)$ (or smaller).
- Use a standard square-and-multiply exponentiation strategy with $O(k)$ multiplications and squarings, giving total size polynomial in k when using Lemma 18.1 for multiplication.

This yields nonuniform polynomial-size circuits for inversion. □

Appendix C. Encoding Conventions and Projection Reductions (Supplementary)

This appendix fixes an explicit encoding convention sufficient to make Lemma 17.6 precise and to justify the use of "projection reductions" in Section 17. It is not intended to be the most space-efficient encoding; it is chosen for clarity and for the projection property.

C.1. Fixed-length encodings

For each input length parameter n, we encode integers and identifiers using fixed-width binary fields of width $w(n) = \Theta(\log n)$. This ensures that all strings $\mathrm{Enc}(\Phi_\oplus(x))$ for $|x| = n$ have the same length $m(n)$.

C.2. Encoding of the parity family $\Phi_\oplus(x)$

We consider only the restricted family $\Phi_\oplus(x)$ used in Section 17. Its underlying system $\mathsf{SysPar}(x)$ has a fixed template consisting of:

- fixed EqTree and XorTree wiring for size n,
- for each j, one gating equation whose type bit is x_j: type 0 means unary constraint $u_j = 0$, type 1 means binary constraint $u_j \oplus \lambda_j = 0$,
- one final check equation.

We encode $\Phi_\oplus(x)$ as:

1. a header describing n and fixed size parameters for the resulting simplicial complex (as fixed-width constants depending on n), and
2. a fixed "template" description of the oracles for $K(\mathsf{SysPar}(x))$, where the only x-dependent bits are the n gating type bits $x_0, \dots, x_{n-1}$ copied into designated constant positions of the oracle descriptions.

Under this convention, changing x_j flips only a constant number of bits in the description, and those bits are literal copies of x_j or $\neg x_j$.

C.3. Projection definition and proof

Definition 18.3 (Projection)**.** *A mapping $\rho : \{0,1\}^n \to \{0,1\}^m$ is a projection if each output bit is a constant or equals an input bit x_j or $\neg x_j$.*

Lemma 18.4 (Projection property of $\Phi_\oplus$)**.** *With the encoding convention above, $x \mapsto \mathrm{Enc}(\Phi_\oplus(x))$ is a projection, and the output length satisfies $m(n) = \Theta(n)$.*

Proof. All header fields and all template wiring bits are constants depending only on n. The only x-dependent part of $\mathsf{SysPar}(x)$ is the choice, for each j, between two fixed gating constraints; we encode that choice by a single designated bit in the description, set equal to x_j. No other description bit depends on x. Therefore each output bit is either constant or equals some x_j. The total number of such bits is $O(n)$, hence $m(n) = \Theta(n)$. □

Remark 18.5 (Scope)**.** *This encoding convention is sufficient for Section 17. It is not a fully general encoding for arbitrary local-oracle descriptions used elsewhere in the paper (e.g., ProbeBit for exponentially large complexes). The projection lower bound requires only this restricted subfamily.*

Bibliography Placeholders (No fabricated references)

The following are placeholders for standard results used in the paper:

- **[VV]** Valiant–Vazirani isolation lemma (randomized reduction from SAT to UniqueSAT).
- [$\oplus$SAT] Standard completeness of $\oplus$SAT for $\oplus\mathbf{P}$.
- **[AC0-Parity]** Standard lower bound PARITY $\notin \mathbf{AC}^0$ (Håstad-type).
- **[Formula-Parity]** Standard quadratic lower bound for parity in De Morgan formulas (Khrapchenko-type).
- **[AC0p-Parity]** Standard lower bound PARITY $\notin \mathbf{AC}^0[p]$ for odd primes p (Razborov–Smolensky-type).

(Full bibliographic entries are intentionally omitted here and should be filled with correct citations in a final submission.)