hexagon:2611.00375v1

Local optimality of rank-47 schemes for 4×4 matrix multiplication over $\mathbb{F}_2$

Tejasvi Singh Tomar*

Abstract

The tensor $\langle 4,4,4\rangle$ of 4×4 matrix multiplication has rank at most 47 over $\mathbb{F}_2$, and as it is not known whether 47 is optimal, we ask how far the known decompositions of rank 47 are from one of rank 46. A decomposition is locally optimal to depth D if no D or fewer of its rank-one terms can be replaced by fewer rank-one terms. We prove that five pairwise inequivalent decompositions of rank 47 over $\mathbb{F}_2$ are locally optimal to depth 18, as is every decomposition equivalent to one of them, so a decomposition of rank 46 over $\mathbb{F}_2$ shares at most 28 terms with any decomposition in these five classes. The five classes include the scheme of Fawzi et al. and every public decomposition of rank 47 over $\mathbb{F}_2$ known to us; one of them, found in our flip-graph searches, is new. The proof rests on a peeling lemma and, for four sets of terms that resist the other bounds, on a refinement of the substitution method, which we call substitution over subspaces with lifting. Local optimality to depth 18 is also proved in Lean 4.

Keywords: matrix multiplication; tensor rank; local optimality; substitution method; flip graphs; SAT solving; Lean 4.

MSC2020: 68Q17, 15A69 (primary); 68V05, 68V15, 68V20 (secondary).

1 Introduction

Let $\langle n,m,p\rangle$ denote the tensor of the bilinear map that multiplies an $n \times m$ matrix by an $m \times p$ matrix. Its rank over a field $\mathbb{F}$ is the least number of multiplications in a bilinear algorithm for this product, and the ranks of the square formats determine the exponent of matrix multiplication [9, 7]. Strassen's algorithm [35] gives $\mathrm{R}(\langle 2,2,2\rangle) \le 7$ and hence $\mathrm{R}(\langle 4,4,4\rangle) \le 49$ over any field. Fawzi et al. [14] found a decomposition (or scheme) of $\langle 4,4,4\rangle$ of rank 47 that is valid in characteristic 2. Kauers and Moosbauer [20] introduce flip graphs and report more than 100,000 decompositions of rank 47 over $\mathbb{F}_2$, none of rank 46; the web collection of Kauers [18] contains only four distinct decompositions up to the order of the terms (Appendix B.1). Later flip-graph searches, including adaptive and symmetric variants [3, 31], found no decomposition of $\langle 4,4,4\rangle$ over $\mathbb{F}_2$ of rank below 47. Over rings that contain $1/2$ the best known rank is 48 [13], and AlphaEvolve found a decomposition of rank 48 over $\mathbb{C}$ [32]; these decompositions use the coefficient $1/2$, so they do not apply over $\mathbb{F}_2$. In the other direction, Bläser's bound $\mathrm{R}(\langle n,m,n\rangle) \ge 2mn + 2n - m - 2$ for $m \ge n \ge 3$ [6] gives $\mathrm{R}(\langle 4,4,4\rangle) \ge 34$ over every field, with $m = n = 4$, and to our knowledge no better lower bound over $\mathbb{F}_2$ has been published; the recent result $\mathrm{R}_{\mathbb{F}_2}(\langle 3,3,3\rangle) = 23$ [37] gives the same value (Remark 2.5). Whether $\mathrm{R}_{\mathbb{F}_2}(\langle 4,4,4\rangle) = 47$ is open.

This paper asks how far the known decompositions of rank 47 are from one of rank 46. Let S be a decomposition of rank 47, viewed as a list of rank-one terms. If a decomposition of rank 46 shares

*ORCID: 0000-0003-4668-3639.

m terms with S, then the other $47-m$ terms of S sum to a tensor of rank at most $46-m$. We call S *locally optimal to depth* D if every set of at most D terms of S has *full rank*, that is, sums to a tensor whose rank over $\mathbb{F}_2$ equals the number of terms. A decomposition of rank 46 then shares at most $46-D$ terms with S (Lemma 2.2). Local optimality to depth D excludes any move that replaces at most D terms of S by fewer terms, such as the reductions used in flip-graph search, and it excludes such moves from every decomposition equivalent to S as well (Section 2.3). Here two decompositions are *equivalent* if they are related by the symmetries of $\langle 4,4,4\rangle$ listed in Section 2.2, up to the order of the terms.

We study five decompositions of $\langle 4,4,4\rangle$ of rank 47 over $\mathbb{F}_2$. AT47 is the decomposition of Fawzi et al. [14], and KM47 is a decomposition from the data of Kauers and Moosbauer [20, 29]. C3 is equivalent to a decomposition in another web collection of Kauers [19], and C5 to the second decomposition of rank 47 of Zaru [43]. C4 was found by our flip-graph searches. By Theorem 3.1, the five are pairwise inequivalent, and none of them admits a flip or a reduction. Every public decomposition of rank 47 over $\mathbb{F}_2$ known to us is equivalent to AT47, KM47, C3 or C5, and none is equivalent to C4 (Appendix B.1). We mark with a dagger (†) the numbered statements whose proofs rely on computer calculation (Section 8).

Theorem 1.1 (†)**.** *Let S be a decomposition of $\langle 4,4,4\rangle$ over $\mathbb{F}_2$ that is equivalent to one of AT47, KM47, C3, C4 and C5. Then S is locally optimal to depth* 18*.*

Corollary 1.2 (†)**.** *A decomposition of $\langle 4,4,4\rangle$ of rank* 46 *over $\mathbb{F}_2$ shares at most* 28 *terms with any decomposition equivalent to one of the five.*

The proof rests on a *peeling lemma* (Lemma 4.2): if one term of a sum of rank-one terms has factors in two of the three positions outside the spans of the other terms' factors in those positions, then removing this term lowers the rank by exactly one. Repeated removal ends in a unique *core*, so local optimality to depth D holds if and only if all subsets of size at most D without such a term, the *non-peelable* subsets, have full rank (Corollary 4.4). We enumerate these subsets up to the automorphism group of the decomposition and bound their ranks from below with flattenings, the Koszul flattenings of Landsberg and Ottaviani [25, 26] and the substitution method [2], mostly with the peeling lemma applied again in every branch of the substitution. Four subsets Y of C3 resist these bounds. For them we eliminate first factors along arbitrary linear forms and record the subspaces spanned by the eliminated first factors, so the proof becomes a directed acyclic graph (DAG) of subspaces whose nodes carry statements about the minimal decompositions of the subset's sum. At some leaves SAT instances on that sum, with the first factors prescribed only modulo the subspace, settle the node; hence the name substitution over subspaces with lifting (Section 7). The method is stated there for any tensor over $\mathbb{F}_2$ that is concise in one position.

The independent checks of the computations and the Lean 4 proofs of Theorems 1.1, 3.1(a) and 7.10, in which most computations are evaluated as compiled code, are described in Section 8.

Related work. Decompositions of small matrix multiplication tensors over $\mathbb{F}_2$ are found with SAT solvers [15, 16], with flip graphs [20] and their variants [3, 31, 21], and with GPU search [28]. Kauers and Moosbauer call a scheme reducible if, among terms with a common factor in one position, the factors in another position are linearly dependent [20, Def. 2]. Arai, Ichikawa and Hukushima reduce a set of terms with a common factor whose sum has rank below the size of the set [3, Def. 2.6]. Local optimality to depth D excludes such a rank drop for all subsets of size at most D, with or without a common factor. So a scheme that is locally optimal to depth D has no reduction of either kind on D or fewer terms. The converse fails: the eight terms of the standard algorithm for $\langle 2,2,2\rangle$ admit no reduction of either kind, although their sum has rank at most 7 [35]. Moosbauer [30, Sec. 4.8]

studies tensor identities, which exchange a set of terms for another set of terms. Zaru [43] searches with a SAT solver for decompositions of $\langle 4,4,4\rangle$ of rank 47 over $\mathbb{F}_2$ that are invariant under the symmetry group of AT47. The search finds a second such decomposition, and Zaru shows that both are flip-isolated and that no decomposition of rank 46 invariant under this group keeps 8 or more of the 10 orbits of terms of either. Agarwal [1] studies the question of this paper for a decomposition of $\langle 4,4,4\rangle$ with 48 terms over $\mathbb{C}$, and shows that every shorter decomposition differs from it in at least 13 terms. On the side of lower bounds, the substitution method goes back to Pan [33], and we use the form of Alexeev, Forbes and Tsimerman [2]. Substitution over subspaces with lifting is compared in Section 7.5 with automated frameworks for certified rank lower bounds over finite fields and with capacity inequalities.

Organization. Section 2 recalls the lower-bound methods, and Section 3 presents the five decompositions. Sections 4 and 5 give the peeling lemma, the enumeration and the certification. Section 6 proves Theorem 1.1, with Section 7 for the four subsets of C3. Section 8 describes the independent checks and the formal proof, and Section 9 closes with open questions. Appendix A lists the decompositions, and Appendix B collects the details of the computations.

2 Preliminaries

We fix notation and recall the two lower-bound methods that we use: flattenings and the substitution method.

2.1 Tensors and decompositions

Let A, B, C be finite-dimensional vector spaces over a field $\mathbb{F}$. A tensor $T \in A \otimes B \otimes C$ has rank one if $T = a \otimes b \otimes c$ with a, b, c nonzero. The *rank* $\mathrm{R}(T)$ is the least r such that T is a sum of r tensors of rank one. The *border rank* $\underline{\mathrm{R}}(T)$ is the least r such that T lies in the Zariski closure, over the algebraic closure $\overline{\mathbb{F}}$, of the set of tensors of rank at most r. Thus $\underline{\mathrm{R}}(T) \le \mathrm{R}(T)$. We follow [9, 22].

We index rows, columns and basis vectors from 0. We write E_{ij} for the matrix units and use the cyclic convention

$$\langle n,m,p\rangle = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\sum_{k=0}^{p-1} E_{ij}\otimes E_{jk}\otimes E_{ki} \in \mathbb{F}^{n\times m}\otimes\mathbb{F}^{m\times p}\otimes\mathbb{F}^{p\times n}.$$

A *decomposition* (or scheme) of rank r is a list S of r rank-one tensors $a_t \otimes b_t \otimes c_t$ whose sum is $\langle n,m,p\rangle$. We call a_t, b_t, c_t the factors of the term t in the first, second and third position. For $X \subseteq S$ we write

$$\Sigma_X = \sum_{t\in X} a_t \otimes b_t \otimes c_t.$$

2.2 Symmetry

For $P, Q, R \in \mathrm{GL}_4(\mathbb{F}_2)$ the map

$$a \otimes b \otimes c \;\longmapsto\; PaQ^{-1} \otimes QbR^{-1} \otimes RcP^{-1}$$

fixes $\langle 4,4,4\rangle$, and so do the cyclic shift $a\otimes b\otimes c \mapsto b\otimes c\otimes a$ and the map $a\otimes b\otimes c \mapsto c^{\mathsf{T}}\otimes b^{\mathsf{T}}\otimes a^{\mathsf{T}}$, as one checks directly. They generate a group $G \cong \mathrm{GL}_4(\mathbb{F}_2)^3 \rtimes S_3$ of order $6\cdot 20160^3 \approx 4.9\cdot 10^{13}$, which acts on decompositions of $\langle 4,4,4\rangle$ term by term. Under this action exactly the scalar triples

$(\alpha I, \beta I, \gamma I)$ act trivially. Over $\mathbb{F}_2$ the only such triple is the identity, so the map from $\mathrm{GL}_4(\mathbb{F}_2)^3$ to G is injective. Two decompositions are *equivalent* if one is the image of the other under G, up to the order of the terms. The automorphism group $\mathrm{Aut}(S)$ is the stabilizer of S in G. It permutes the terms of S.

2.3 Local optimality

Definition 2.1. A list S of rank-one tensors is *locally optimal to depth D*, written $\mathrm{LO}(D)$, if $\mathrm{R}(\Sigma_X) = |X|$ for all $X \subseteq S$ with $|X| \le D$.

We say that a set X of terms has *full rank* if $\mathrm{R}(\Sigma_X) = |X|$. Invertible linear maps on each factor and permutations of the three positions preserve rank. Therefore $\mathrm{LO}(D)$ is invariant under these maps, and in particular under equivalence.

Lemma 2.2. *Let S and S' be decompositions of the same tensor, of ranks r and $r-1$. If S is $\mathrm{LO}(D)$, then S and S' have at most $r-1-D$ terms in common.*

Proof. Let Y be the common part of S and S', counted with multiplicity. Then $\Sigma_{S\setminus Y} = \Sigma_{S'\setminus Y}$, so $\mathrm{R}(\Sigma_{S\setminus Y}) \le r - 1 - |Y| < |S \setminus Y|$. Hence $|S \setminus Y| > D$, that is, $|Y| \le r - 1 - D$. □

2.4 Flattening bounds

Let $T \in A \otimes B \otimes C$. The flattening $T_A \colon A^* \to B \otimes C$ has rank at most $\mathrm{R}(T)$, and the same holds for T_B and T_C. We also use the Koszul flattenings of Landsberg and Ottaviani [25, 26], a special case of their Young flattenings. Fix a linear map $\pi \colon A \to V = \mathbb{F}^{2p+1}$ with $p \ge 1$, and let $M_0, \dots, M_{2p}$ be the slices of $(\pi \otimes 1 \otimes 1)T$, viewed as maps $B^* \to C$. Define

$$T_\pi^{\wedge p} \colon \Lambda^p V \otimes B^* \to \Lambda^{p+1} V \otimes C, \qquad \omega \otimes \beta \mapsto \sum_{k=0}^{2p} (e_k \wedge \omega) \otimes M_k(\beta).$$

A term $a \otimes b \otimes c$ with $\pi(a) \neq 0$ contributes $(\pi(a) \wedge \cdot) \otimes (\beta \mapsto \beta(b)c)$. The map $\omega \mapsto \pi(a) \wedge \omega$ from $\Lambda^p V$ to $\Lambda^{p+1} V$ has rank $\binom{2p}{p}$ over any field. Hence

$$\mathrm{R}(T) \;\ge\; \left\lceil \mathrm{rk}\, T_\pi^{\wedge p} \Big/ \tbinom{2p}{p} \right\rceil. \tag{1}$$

We use $p = 1$ (divisor 2), $p = 2$ (divisor 6) and $p = 3$ (divisor 20). Landsberg and Ottaviani call all of these Koszul flattenings, and so do we. For $p = 1$ and $\mathbb{F} = \mathbb{F}_2$, the matrix of $T_\pi^{\wedge 1}$ is the block matrix

$$K = \begin{pmatrix} 0 & M_2 & M_1 \\ M_2 & 0 & M_0 \\ M_1 & M_0 & 0 \end{pmatrix},$$

since signs vanish in characteristic 2. A term with $\pi(a) = 0$ contributes nothing. So if π kills a term of a list X, the bound (1) for Σ_X is at most $|X| - 1$.

Matrix ranks do not change under field extension, and the tensors for which one of these maps has rank at most m form a Zariski-closed set, defined by minors. Hence the bounds above are lower bounds for $\underline{\mathrm{R}}(T)$ as well as for $\mathrm{R}(T)$. None of them, however, can prove $\mathrm{R}(T) \ge r$ when $\underline{\mathrm{R}}(T) < r$. We call this the *border-rank barrier*.

2.5 The substitution method

The substitution method goes beyond the border-rank barrier. We use the following form, due to Alexeev, Forbes and Tsimerman [2].

Lemma 2.3 ([2])**.** *Let $T = \sum_{i=0}^{n-1} e_i \otimes M_i \in \mathbb{F}^n \otimes B \otimes C$, and suppose that $M_0, \dots, M_{k-1}$ are linearly independent. Then*

$$\mathrm{R}(T) \;\geq\; k + \min_{c} \;\mathrm{R}\Big(\sum_{i=k}^{n-1} e_i \otimes \Big(M_i + \sum_{j=0}^{k-1} c_{ij} M_j\Big)\Big),$$

where c runs over all $(n-k) \times k$ matrices over $\mathbb{F}$, with rows indexed by $i \in \{k, \dots, n-1\}$ and columns by $j \in \{0, \dots, k-1\}$.

The proof is short. Take an optimal decomposition $T = \sum_{l=1}^{r} \alpha_l \otimes \beta_l \otimes \gamma_l$. Since $M_0, \dots, M_{k-1}$ are independent, the first k coordinates of the α_l span $\mathbb{F}^k$. A projection of $\mathbb{F}^n$ onto $\mathrm{span}(e_k, \dots, e_{n-1})$ can therefore kill k of the α_l. It maps T to one of the tensors on the right, which therefore has rank at most $r - k$.

We apply the lemma in any position, after permuting the positions. For $k = 1$ and a pivot j with $M_j \neq 0$, it reads

$$\mathrm{R}(T) \;\geq\; 1 + \min_{u} \;\mathrm{R}\Big(\sum_{i \neq j} e_i \otimes (M_i + u_i M_j)\Big), \tag{2}$$

with u running over $\mathbb{F}^{n-1}$. The hypothesis $M_j \neq 0$ is essential. If $M_j = 0$, every branch is T with a zero slice removed, and (2) would spuriously raise the bound by one. Over $\mathbb{F}_2$ the minimum has 2^{n-1} branches. Applying (2) recursively, in any position and with any pivot, gives a tree of tensors with bounds from Section 2.4 at the leaves. We call this *plain substitution*.

Example 2.4. The tensor $W = e_0 \otimes e_0 \otimes e_1 + e_0 \otimes e_1 \otimes e_0 + e_1 \otimes e_0 \otimes e_0$ has border rank 2. It is the value at $\varepsilon = 0$ of the curve $\varepsilon^{-1}((e_0 + \varepsilon e_1)^{\otimes 3} - e_0^{\otimes 3})$, whose other values have rank at most 2. Its first-position slices are $M_0 = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ and $M_1 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$. With pivot 1, the bound (2) gives $\mathrm{R}(W) \geq 1 + \min_u \mathrm{rk}(M_0 + uM_1) = 3$.

Remark 2.5. Over $\mathbb{F}_2$ the rank of $\langle 3,3,3\rangle$ is 23 [37], and this value also gives $\mathrm{R}_{\mathbb{F}_2}(\langle 4,4,4\rangle) \geq 34$. Lemma 2.3, applied to the m independent slices of the last column of the second matrix, and setting the last row of the third matrix to zero give $\mathrm{R}(\langle n,m,p\rangle) \geq \mathrm{R}(\langle n,m,p-1\rangle) + m$. With the symmetries of matrix multiplication, the bound 23 then improves to 26, 30 and 34 in turn. By the same step, Wang's automated bound $\mathrm{R}_{\mathbb{F}_2}(\langle 3,4,4\rangle) \geq 29$ [38] yields only 33.

3 Five decompositions of rank 47

We encode a matrix $(m_{ij}) \in \mathbb{F}_2^{4\times 4}$ by the integer $\sum_{i,j} m_{ij} 2^{4i+j}$. We list the five decompositions in this encoding in Appendix A, with terms numbered from 0. AT47 and KM47 are named after their sources, and the further classes are numbered C3, C4 and C5 in the order in which our searches found them. For AT47 we read the decomposition of [14] over $\mathbb{F}_2$ in its index convention, which is the cyclic convention.

Theorem 3.1 (†)**.**

(a) *Each of AT47, KM47, C3, C4 and C5 is a decomposition of $\langle 4,4,4\rangle$ over $\mathbb{F}_2$ with 47 terms.*
(b) *The five decompositions are pairwise inequivalent.*
(c) *Their automorphism groups have orders 12, 2, 4, 12 and 12.*

(d) *In each of them, the* 47 *factors in each position are pairwise distinct.*

Proof. Part (a) is a direct expansion. For (b), let $\rho(S) = \sum_{t\in S}(\operatorname{rk} a_t + \operatorname{rk} b_t + \operatorname{rk} c_t)$ be the sum of the matrix ranks of all 141 factors. An element of G maps every factor to a matrix PaQ^{-1} or to a transpose, of the same rank, and permutes the three positions. Thus ρ is constant on equivalence classes. The five values of ρ in Table 1 are distinct. We computed the automorphism groups in (c) by an exhaustive search over G. For each of the six permutations of the positions and each pair $(P, Q) \in \mathrm{GL}_4(\mathbb{F}_2)^2$, the search keeps the pair only if the transformed first-position factors are, as a multiset, the first-position factors of S. For each surviving pair it tries every R. Part (d) can be read off from Appendix A. □

By part (d), in each position the 47 factors of each of the five are pairwise distinct, and a flip or a reduction needs two terms with equal factors in one position, so in the flip graph of [20] no flip and no reduction starts at the five, and no flip ends at them.

Table 1: The five decompositions of rank 47 (sources in Section 1).

Decomposition	$\lvert\mathrm{Aut}(S)\rvert$	$\rho(S)$
AT47	12	278
KM47	2	276
C3	4	272
C4	12	282
C5	12	274

Remark 3.2 (†)**.** Every decomposition from the first 41,809 descents (search runs) of our flip-graph searches is equivalent to one of the five. The later decompositions of these searches carry the invariant hash of one of the five classes, and a random sample of 4,083 of them was matched explicitly (Appendix B.1 defines the hash and gives the details).

4 The peeling lemma

Throughout this section we fix a finite list L of rank-one tensors $a_s \otimes b_s \otimes c_s$ in $A \otimes B \otimes C$ over any field $\mathbb{F}$, and we write Σ_L for their sum. For $t \in L$ we write $L - t$ for the list without t.

Definition 4.1. A term $t \in L$ is *free in the first position* if $a_t \notin \operatorname{span}\{a_s : s \in L - t\}$, that is, if t is a coloop of the vector configuration $(a_s)_{s\in L}$. Freeness in the second and third positions is defined in the same way. A term is *peelable* in L if it is free in at least two positions, and L is *non-peelable* if none of its terms is peelable.

The following lemma is a direct consequence of the substitution lemma [17, Lemma 2], [2, Lemma B.1], [23, Prop. 5.3.1.1], combined with a projection in a second position. The same two-position argument appears for direct sums in [24, Thm. 4.1] and [7, Lemma 7.1]. We include the short proof because we need the lemma in the form below, where the third factor is unrestricted.

Lemma 4.2 (Peeling lemma)**.** *If t is peelable in L, then* $\mathrm{R}(\Sigma_L) = \mathrm{R}(\Sigma_{L-t}) + 1$.

Proof. Clearly $\mathrm{R}(\Sigma_L) \le \mathrm{R}(\Sigma_{L-t}) + 1$. For the converse we may assume that t is free in the first two positions. Let $r = \mathrm{R}(\Sigma_L)$ and $\Sigma_L = \sum_{i=1}^{r} \alpha_i \otimes \beta_i \otimes \gamma_i$. Choose $a^* \in A^*$ with $a^*(a_t) = 1$ and

$a^*(a_s) = 0$ for $s \neq t$. Contracting the first factor with a^* gives

$$b_t \otimes c_t = \sum_{i=1}^{r} a^*(\alpha_i)\, \beta_i \otimes \gamma_i.$$

The left side is nonzero, so $a^*(\alpha_i) \neq 0$ for some i. After renumbering and rescaling we may assume $a^*(\alpha_1) = 1$. The map $\varphi(v) = v - a^*(v)\alpha_1$ kills α_1 and fixes a_s for $s \neq t$. Hence

$$(\varphi \otimes 1 \otimes 1)\Sigma_L = \Sigma_{L-t} + (a_t - \alpha_1) \otimes b_t \otimes c_t = \sum_{i=2}^{r} \varphi(\alpha_i) \otimes \beta_i \otimes \gamma_i$$

has rank at most $r - 1$. Now choose $b^* \in B^*$ with $b^*(b_t) = 1$ and $b^*(b_s) = 0$ for $s \neq t$, and let $\psi(w) = w - b^*(w)\, b_t$. The map ψ kills b_t and fixes b_s for $s \neq t$. So $1 \otimes \psi \otimes 1$ maps the tensor above to Σ_{L-t}. Linear maps do not increase rank, and therefore $\mathrm{R}(\Sigma_{L-t}) \leq r - 1$. □

Two free positions are needed: in the list $L = (e_0 \otimes e_0 \otimes e_0,\ e_1 \otimes e_0 \otimes e_0)$ the second term is free in the first position only, and both Σ_L and Σ_{L-t} have rank one.

Freeness is monotone. If t is free in some position within L, then it is free in that position within any sublist of L that contains t. This makes the result of repeated peeling canonical.

Lemma 4.3. *Delete peelable terms from L one at a time until no peelable term is left. The final list* $\mathrm{core}(L)$ *does not depend on the order of deletion. It is the largest non-peelable sublist of L, and*

$$\mathrm{R}(\Sigma_L) = \mathrm{R}(\Sigma_{\mathrm{core}(L)}) + |L| - |\,\mathrm{core}(L)|.$$

Proof. Let $t_1, \ldots, t_m$ be the terms deleted in one run, and let $N \subseteq L$ be a non-peelable sublist. Suppose some t_i lies in N, and take the least such i. Then $N \subseteq L - \{t_1, \ldots, t_{i-1}\}$, and t_i is peelable in this list, hence in N by monotonicity. This is a contradiction. Hence N is contained in the final list of the run. The final list is itself non-peelable, so it is the largest non-peelable sublist of L and does not depend on the run. The rank formula follows from Lemma 4.2. □

Corollary 4.4. *A list S is* $\mathrm{LO}(D)$ *if and only if* $\mathrm{R}(\Sigma_X) = |X|$ *for all non-peelable $X \subseteq S$ with $|X| \leq D$.*

Proof. One direction is trivial. For the other, let $X \subseteq S$ with $|X| \leq D$. The list $\mathrm{core}(X)$ is non-peelable of size at most D, so $\mathrm{R}(\Sigma_{\mathrm{core}(X)}) = |\,\mathrm{core}(X)|$. Lemma 4.3 then gives $\mathrm{R}(\Sigma_X) = |X|$. □

Lemma 4.5. *Let h be an invertible linear map on each position composed with a permutation of the positions, for example an element of G, and write $h(L)$ for the list of images of the terms of L. Let τ be the permutation of the positions. Then t is free in position k in L if and only if $h(t)$ is free in position $\tau(k)$ in $h(L)$. Moreover, t is peelable in L if and only if $h(t)$ is peelable in $h(L)$,* $\mathrm{core}(h(L)) = h(\mathrm{core}(L))$, *and* $\mathrm{R}(\Sigma_{h(L)}) = \mathrm{R}(\Sigma_L)$.

Proof. The map h preserves linear dependence among the factors in each position, up to the permutation of positions, and it preserves rank. □

4.1 Peeling inside substitution

Let L be a list with first factors in $\mathbb{F}^n$, and let j be a pivot such that the slice M_j of Σ_L is nonzero. For $u \in \mathbb{F}^{n-1}$ let $\varphi_u \colon \mathbb{F}^n \to \mathbb{F}^{n-1}$ map e_j to $\sum_{i \neq j} u_i e_i$ and fix e_i for $i \neq j$. The branch tensor in (2) is

$$(\varphi_u \otimes 1 \otimes 1)\Sigma_L = \sum_{s \in L} \varphi_u(a_s) \otimes b_s \otimes c_s.$$

After the terms with $\varphi_u(a_s) = 0$ are deleted, this is again a list L_u of rank-one terms. Lemma 4.3 therefore applies to every branch. To prove that a branch has rank at least k, it suffices to prove $\mathrm{R}(\Sigma_{\mathrm{core}(L_u)}) \geq k - |L_u| + |\operatorname{core}(L_u)|$. We call substitution with this peeling step at all nodes *peeled substitution*. It can be much cheaper than plain substitution. For example, the computation of depth 16 without peeling (Table 5) treats each subset of size 16 in full. For one subset of C5 there, the shortest plain-substitution certificate known to us has four levels and about $9 \cdot 10^7$ branches. That subset has two peelable terms, so the peeling enumeration never lists it. It lists the core of 14 terms instead, and peeled substitution certifies that core with three levels.

5 Enumeration and certification

We now fix one of the five decompositions S and a depth D. By Corollary 4.4 and the invariance of $\mathrm{LO}(D)$ under equivalence (Section 2.3), Theorem 1.1 reduces to two tasks for each of the five: list the non-peelable subsets $X \subseteq S$ with $|X| \leq D$, and prove $\mathrm{R}(\Sigma_X) \geq |X|$ for each of them.

5.1 Enumeration

We search depth first over increasing sequences of term indices. A node is a set $Y \subseteq S$, and its subtree consists of the sets $Y \cup Z$ with Z a set of indices larger than $\max Y$.

Symmetry. The search numbers the terms in its own order: the $\mathrm{Aut}(S)$-orbits of terms in order of decreasing size, with ties broken by the least index, and within an orbit by increasing index in Appendix A. Lexicographic comparisons refer to this order, and we compare sets as sorted sequences. Since $\mathrm{Aut}(S)$ permutes the terms and preserves ranks and non-peelability (Lemma 4.5), we list only the lexicographically least set of each $\mathrm{Aut}(S)$-orbit of subsets. If $\sigma(Y)$ precedes Y for some $\sigma \in \mathrm{Aut}(S)$, then no set in the subtree of Y is lexicographically least, and we skip the subtree. Indeed, let $X = Y \cup Z$ with $Z > \max Y$, and let i be the first position where the sorted sequences $\sigma(Y)$ and $Y = (y_1, y_2, \dots)$ differ. Then $\sigma(X)$ has at least i elements below y_i, while X has exactly $i-1$, and $\sigma(X)$ agrees with X or lies below it at the earlier positions. So $\sigma(X)$ precedes X. A listed set X carries the size $|\mathrm{Aut}(S)|/|\mathrm{Stab}(X)|$ of its orbit as a weight, where $\mathrm{Stab}(X)$ is the stabilizer of X in $\mathrm{Aut}(S)$. The pruning is sound for any group of automorphisms of S, so it does not depend on knowing the full group. The subsets in Tables 3 and 4 are the representatives that the search found, written in the numbering of Appendix A. In that numbering the first subset of AT47 in Table 4 is not lexicographically least in its orbit.

Free terms. A term is free in a position if and only if it lies in no linear dependency among the factors in that position. Along the current search path we keep echelon forms of the factors in the three positions. Each row carries a tag, the set of terms it combines, and tags combine by symmetric difference, so a tag is exactly the support of the combination. A factor that reduces to zero gives a dependency, whose support is its tag. These dependencies form a basis of the space of all dependencies, since each involves a term that the earlier ones do not. The union of the supports

of a basis is the union of the supports of all dependencies, which is the set of terms that are not free. We therefore know the free terms of Y at constant cost per node.

Universe pruning. All sets in the subtree of Y lie in the universe $W = Y \cup \{s \in S : s > \max Y\}$. If a term of Y is free in two positions within W, then by monotonicity it is peelable in every set of the subtree, and we skip the subtree. To test this, we store a basis of the linear dependencies among the factors of the terms in W, separately in the three positions, as matrices with columns indexed by W. A term is free within W if and only if its column is zero. When an index leaves W, the update is one elimination step, a contraction in the dual matroid. In our C++ implementation of the enumeration, the correctness of this update is supported by the agreement of counts (Table 5).

5.2 Certification

Let X be a listed non-peelable set with $|X| = k$. We write $T = \Sigma_X$ in reduced coordinates: in each position, the factors of X are expressed in a basis of their span. For a list of terms, the greedy reduced coordinates are defined position by position: a factor of a term, taken in list order, becomes a basis vector if it is not in the span of the factors kept before it. We try the following bounds in this order and stop at the first one that gives $\mathrm{R}(T) \geq k$.

1. The three flattening ranks.
2. Koszul flattenings with $p = 1$ for four fixed projections $\pi\colon \mathbb{F}_2^{4\times 4} \to \mathbb{F}_2^3$ of the first position. They are chosen so that $\pi(a_s) \neq 0$ for all $s \in S$, since a projection that kills a term of X cannot certify X (Section 2.4). In ambient coordinates their matrices change by one term per step of the search, so we update them incrementally.
3. Koszul flattenings with $p = 1$ and $p = 2$, for random projections in reduced coordinates.

We call tests 1–3 the in-line tests, since they run during the enumeration. In the depth-16 run and in the first part of the depth-18 run they also included one level of substitution. We pass the sets left after the in-line tests to post-processing, which has three stages. In the first two we apply peeled substitution with the bounds above at the leaves, first with two levels and then with three or four levels and all pivots. In the third, for the few sets left after that, we use up to five levels of peeled substitution with the leaf bounds above, or of plain substitution with Koszul flattenings for $p = 3$ at the leaves.

Certificates. Apart from the four subsets of C3 (Section 7), every bound except one is found by a deterministic certifying program, with fixed seeds, that can emit it as an explicit certificate (Appendix B.5); the exception is the subset of C4 (Section 6.2).

5.3 Counts

The numbers of non-peelable subsets of each size $k \leq 18$, weighted by orbit size, are in Table 2. The outcome of the post-processing is in Appendix B.2.

For the five decompositions the reduction to non-peelable subsets removes almost all subsets. About 98.4% of the subsets of size 16 have an empty core, and the non-peelable subsets are about 0.05% of the subsets of size 16. Over the five decompositions there are about $3.5 \cdot 10^9$ non-peelable subsets of size 16 (Table 2). For comparison, about $6.46 \cdot 10^{12}$ of the $5\binom{47}{16} \approx 7.52 \cdot 10^{12}$ subsets of size 16, or 86%, have all three flattenings rank-deficient, so the flattening test does not certify them.

Table 2: Non-peelable subsets of size k, weighted by orbit size. There are no nonempty ones with $k \leq 6$. All of them have full rank.

k	AT47	KM47	C3	C4	C5
7	0	8	0	26	0
8	183	198	204	180	219
9	188	145	178	158	182
10	1,827	1,723	1,807	1,721	1,985
11	7,735	7,887	7,761	7,626	7,875
12	73,106	76,875	73,892	70,885	76,369
13	593,114	638,232	617,673	582,344	633,839
14	5,663,006	6,189,240	5,925,614	5,588,900	6,113,109
15	59,345,163	65,789,884	61,773,855	58,027,883	63,924,882
16	680,464,489	750,094,461	692,083,701	671,378,921	722,958,867
17	7,362,510,297	7,977,283,543	7,358,929,678	7,517,279,220	7,788,773,768
18	63,184,632,311	66,992,981,254	62,635,166,253	67,371,750,921	66,540,556,157

6 Proof of the main theorem

6.1 Depth 18

Table 3: The four hardest subsets of C3 (term numbers as in Table 6). Spans gives the dimensions of the spans of the factors in the three positions.

	Terms	$\lvert Y\rvert$	Spans
Y_1	0 5 8 9 10 11 13 15 16 19 24 25 28 34 35 39 44 45	18	$10 \times 11 \times 10$
Y_2	0 3 4 5 9 15 17 19 23 25 26 35 38 41 42 44 45 46	18	$9 \times 12 \times 10$
Y_3	0 3 4 5 9 15 17 19 23 26 35 36 37 38 41 42 44 46	18	$9 \times 10 \times 12$
Y_4	0 5 8 10 11 13 16 19 24 25 28 32 34 35 39 44 45	17	$9 \times 11 \times 9$

Proof of Theorem 1.1. As in Section 5, we only need to prove $\mathrm{R}(\Sigma_X) = |X|$ for the non-peelable subsets X of size at most 18 of the five decompositions, up to their automorphism groups, which preserve non-peelability (Lemma 4.5). We list these subsets with the enumeration (Section 5.1; their numbers are in Table 2). All of them except nine orbits are certified by the in-line tests or in post-processing (Section 5.2). Five of the nine, the subsets of Table 4, are settled in Section 6.2. The other four are the orbits of Table 3, which have full rank by Theorem 7.10 and Lemma 4.5. □

Proof of Corollary 1.2. Apply Lemma 2.2 with $r = 47$ and $D = 18$. □

Remark 6.1 (†)**.** Local optimality to depth 16, which Theorem 1.1 implies, was also proved in a separate run of the same argument, in which every listed subset was certified by the in-line tests or in post-processing. At depth 16 there are two further checks: an enumeration without peeling, with an independent checker on all of its post-processing certificates, and a Lean proof with 40 compiled evaluations (Table 5).

6.2 The hardest settled subsets

We settle the five subsets of Table 4 as follows.

For the subset X of C4, write Σ_X in the reduced coordinates of the second position, with the greedy basis (Section 5.2) of its second-position factors taken in the term order 0, 1, 11, 17, 3, 22, 38, 15, 39, 40, 6, 26, 9, 14, 4, 31, 36, 32. Its first two vectors b_0 and b_1 are the second factors of terms 0 and 1. In these coordinates Σ_X has 16 linearly independent slices $M_0, \dots, M_{15}$ in the second position. We apply Lemma 2.3 in this position with $M_2, \dots, M_{15}$ as the $k = 14$ independent slices. The tensors in the minimum are then the pencils $e_0 \otimes s_0 + e_1 \otimes s_1$ with $s_0 \in M_0 + \operatorname{span}(M_2, \dots, M_{15})$ and $s_1 \in M_1 + \operatorname{span}(M_2, \dots, M_{15})$. Each pair (s_0, s_1) comes from exactly one matrix c, so there are 2^{28} pencils. Equivalently, the pencils are $\big((1 \otimes f_0 \otimes 1)\Sigma_X, (1 \otimes f_1 \otimes 1)\Sigma_X\big)$, where f_0 and f_1 range over the linear forms on the span of the second factors with $f_i(b_j) = \delta_{ij}$ for $i, j \in \{0, 1\}$. The two families therefore depend only on b_0 and b_1. The count 57 below refers to this choice of b_0 and b_1. A pencil has rank at least the matrix rank of each of its slices, and only 57 of the 2^{14} matrices in each of the two families have rank at most 3. For each of the remaining $57 \cdot 57 = 3{,}249$ pairs, one of the flattenings $\left[\, s_0 \;\; s_1 \,\right]$ and $\left[\begin{smallmatrix} s_0 \\ s_1 \end{smallmatrix}\right]$ has rank at least 4. Hence every pencil has rank at least 4, and $\mathrm{R}(\Sigma_X) \geq 14 + 4 = 18$.

The other four subsets of Table 4 were certified with the methods of the third stage of post-processing, in further runs of the certifying programs with other fixed seeds: the two subsets of AT47 by peeled substitution with four and three levels, and the two subsets of KM47 by plain substitution with all pivots, with two and four levels.

Table 4: The five hardest settled subsets, all of size 18 (term numbers as in Table 6, Spans as in Table 3). Size is the number of nodes of the certificate as counted by the checkers, and for C4 the number of pencils.

Scheme	Terms	Spans	Method	Size
C4	0 1 3 4 6 9 11 14 15 17 22 26 31 32 36 38 39 40	$10 \times 16 \times 10$	Lemma 2.3, $k = 14$	2^{28} pencils
AT47	13 14 15 17 20 21 24 26 29 32 33 34 37 42 43 44 45 46	$10 \times 10 \times 11$	peeled, 4 levels	$1.3 \cdot 10^6$ nodes
AT47	13 14 15 17 18 19 20 24 26 28 32 34 36 37 40 43 44 45	$12 \times 10 \times 11$	peeled, 3 levels	$1.8 \cdot 10^8$ nodes
KM47	0 2 10 11 12 16 17 18 19 20 22 24 27 32 33 35 41 42	$10 \times 15 \times 10$	plain, 2 levels	$1.3 \cdot 10^8$ nodes
KM47	4 5 6 8 10 11 12 16 20 21 23 25 30 33 34 36 37 44	$9 \times 11 \times 12$	plain, 4 levels	$1.4 \cdot 10^7$ nodes

7 Substitution over subspaces with lifting

This section develops a lower-bound method for any tensor T over $\mathbb{F}_2$ that is concise in one position and any bound c. It proves $\mathrm{R}(T) \geq c$ through statements about the minimal decompositions of T. Neither these statements, one for each node of a certificate, nor the capacity, rank and lifting leaves (Section 7.3) bound the border rank, so the method can pass the border-rank barrier. Compared with the substitution certificates (Section 5.2), its certificates (i) substitute in one position only, with an arbitrary nonzero linear form as pivot, and record the subspace spanned by the eliminated first factors; (ii) pass from each node to its parent a statement about the decompositions of the whole tensor, where plain substitution would pass a lower bound for the rank of a quotient; and (iii) settle some leaves by SAT instances on the whole tensor. We apply the method to the four subsets of C3 in Table 3 (Section 7.4).

Throughout this section the field is $\mathbb{F}_2$, $T \in A \otimes B \otimes C$, and $c \geq 1$ is the bound to be proved. For the subsets of C3 we take $T = \Sigma_Y$ in reduced coordinates (Section 5.2) and $c = |Y|$. Any of the three positions of Y may serve as A.

7.1 Quotients and the invariant

We assume that $A \neq 0$ and that T is *concise* in the first position, that is, $T \notin A' \otimes B \otimes C$ for every proper subspace $A' \subset A$. In particular $T \neq 0$. The first factors of every decomposition of T span A, since otherwise T would lie in $A' \otimes B \otimes C$ for their span A'. For a subspace $U \subseteq A$ let $\pi_U \colon A \to A/U$ be the quotient map, and let $T/U = (\pi_U \otimes 1 \otimes 1)T$. A decomposition $T = \sum_l \alpha_l \otimes \beta_l \otimes \gamma_l$ gives

$$T/U = \sum_{l\colon\ \alpha_l \notin U} \pi_U(\alpha_l) \otimes \beta_l \otimes \gamma_l, \tag{3}$$

because the terms with $\alpha_l \in U$ vanish. Applied to $T = \Sigma_Y$, (3) writes T/U as the sum of the list L_U of the rank-one terms $\pi_U(a_t) \otimes b_t \otimes c_t$ with $t \in Y$ and $a_t \notin U$, in the order of Y.

A set E of terms of a decomposition *spans* U if the first factors of the terms in E span U. A decomposition of T is *minimal* if it has $\mathrm{R}(T)$ terms. Over $\mathbb{F}_2$ the terms of a minimal decomposition are pairwise distinct, since two equal terms cancel.

Definition 7.1. For a subspace $U \subseteq A$, let $N(U)$ be the statement that if $\mathrm{R}(T) < c$, then no minimal decomposition of T has a set of terms that spans U.

Lemma 7.2. *Let $U \subseteq A$ be a subspace of dimension d.*
(a) *$N(0)$ holds if and only if $\mathrm{R}(T) \geq c$.*
(b) *If $\mathrm{R}(T/U) \geq c - d$, then $N(U)$ holds. In particular, $N(U)$ holds if $d \geq c$.*

Proof. (a) Minimal decompositions exist, and in each of them the empty set of terms spans 0. Hence $N(0)$ fails exactly when $\mathrm{R}(T) < c$. (b) Let Δ be a minimal decomposition of T with $|\Delta| < c$, and let $E \subseteq \Delta$ span U. Then $|E| \geq d$, and by (3) the terms of $\Delta \setminus E$ give a decomposition of T/U with at most $c - 1 - d$ terms. Hence $\mathrm{R}(T/U) \leq c - 1 - d$. □

The converse of (b) fails, because $N(U)$ concerns only those decompositions of T/U that come from a decomposition of T with a set of terms spanning U. Thus $\mathrm{R}(T/U)$ can be smaller than $c - d$ while $N(U)$ holds, and Section 7.4 gives many such subspaces for Y_4.

7.2 Substitution with general pivots

For a nonzero point p of A/U let $U_p = \pi_U^{-1}(\langle p \rangle)$. This is the subspace $U + \langle \alpha \rangle$ for any $\alpha \in A$ with $\pi_U(\alpha) = p$, and it has dimension $\dim U + 1$. Distinct points give distinct subspaces, since over $\mathbb{F}_2$ the line $\langle p \rangle$ has only one nonzero point.

Lemma 7.3. *Let $U \subsetneq A$ be a subspace, and let f be a nonzero linear form on A/U. If $N(U_p)$ holds for every $p \in A/U$ with $f(p) = 1$, then $N(U)$ holds.*

Proof. Suppose that $N(U)$ fails, so that some minimal decomposition Δ of T with $|\Delta| < c$ has a set E of terms that spans U. The linear form $f \circ \pi_U$ on A is nonzero, and the first factors of Δ span A because T is concise. Hence $f(\pi_U(\alpha_t)) = 1$ for some term t of Δ, with first factor α_t. Then $\alpha_t \notin U$, so $t \notin E$, and $E \cup \{t\}$ spans $U + \langle \alpha_t \rangle = U_p$ with $p = \pi_U(\alpha_t)$. Since $f(p) = 1$, this contradicts $N(U_p)$. □

There are $2^{\dim(A/U)-1}$ points p with $f(p) = 1$, and as many children U_p. If f is a coordinate function x_j for a basis $e_0, e_1, \dots$ of A/U, the points with $f(p) = 1$ are $p = e_j + \sum_{i \neq j} u_i e_i$, and $T/U_p = (T/U)/\langle p \rangle$ is the branch tensor of (2) for T/U, with pivot j and parameter u, once $(A/U)/\langle p \rangle$ is identified with the span of the e_i with $i \neq j$. So for a coordinate function the lemma is one step of plain substitution, and for a general f it is the same step in another basis of A/U.

Suppose that $N(U_p)$ is known for every point p outside a set Q. Then Lemma 7.3 applies with some f if and only if Q does not span A/U, since f must vanish on Q. With coordinate functions only, Q would have to lie in a coordinate hyperplane, so general pivots settle more nodes.

Definition 7.4. A *DAG certificate* for (T, c) is a finite set $\mathcal{N}$ of subspaces of A with $0 \in \mathcal{N}$, in which every $U \in \mathcal{N}$ carries a label. The label is either a substitution label f, where f is a nonzero linear form on A/U such that $U_p \in \mathcal{N}$ for every p with $f(p) = 1$, or a leaf label (Section 7.3). A leaf label at U is *valid* if the corresponding rule proves $N(U)$.

The edges from a node U with a substitution label f to its children U_p form a directed acyclic graph, because the dimension rises by one along each edge. Since the statement $N(U)$ depends only on U, whatever sequence of pivots a search follows to reach it, a subspace reached along several paths is a single node with a single label.

Theorem 7.5. *If all leaf labels of a DAG certificate for (T, c) are valid, then $N(U)$ holds for every $U \in \mathcal{N}$. In particular,* $\mathrm{R}(T) \geq c$.

Proof. We prove $N(U)$ by downward induction on $\dim U$. Let $U \in \mathcal{N}$, and assume $N(U')$ for all $U' \in \mathcal{N}$ with $\dim U' > \dim U$. If U has a leaf label, $N(U)$ holds because the label is valid. If U has a substitution label f, all its children U_p with $f(p) = 1$ lie in $\mathcal{N}$ and have dimension $\dim U + 1$, so $N(U_p)$ holds, and Lemma 7.3 gives $N(U)$. For $U = 0$, Lemma 7.2(a) gives $\mathrm{R}(T) \geq c$. □

The nodes reachable from 0 form a DAG certificate by themselves, so the other nodes play no role and need not carry valid labels.

7.3 Leaves

Let $U \in \mathcal{N}$ have dimension d, and let $k = c - d$. Lemma 4.3, applied to the list L_U, gives

$$\mathrm{R}(T/U) = p_U + \mathrm{R}(X_U), \qquad \text{where } X_U = \Sigma_{\mathrm{core}(L_U)} \text{ and } p_U = |L_U| - |\,\mathrm{core}(L_U)|.$$

We write X_U in the greedy reduced coordinates of $\mathrm{core}(L_U)$, with first position $\mathbb{F}_2^e$. The rank of X_U is the same in these coordinates as in $(A/U) \otimes B \otimes C$, because projecting a decomposition of X_U onto the spans of the factors of X_U, along fixed complements, gives a decomposition of X_U with at most as many terms. A minimal decomposition therefore stays minimal under the projection, so none of its factors vanishes and its terms stay distinct. The first three kinds of leaves prove $\mathrm{R}(X_U) \geq k - p_U$, which gives $N(U)$ by Lemma 7.2(b). The fourth proves $N(U)$ directly.

Flattening leaves. A flattening bound for X_U with target $k - p_U$, given and checked as at the leaves of substitution certificates.

Capacity leaves. Let $X \in \mathbb{F}_2^e \otimes B' \otimes C'$ be a tensor. For a subspace $K \subseteq \mathbb{F}_2^e$ we form X/K as above, and for a linear form g on $\mathbb{F}_2^e$ we write $X(g) = (g \otimes 1 \otimes 1)X$, a matrix in $B' \otimes C'$. A *multiset* on a set P is a map $m\colon P \to \mathbb{Z}_{\geq 0}$. Its size is $|m| = \sum_{p \in P} m(p)$, and we write $m(P') = \sum_{p \in P'} m(p)$ for $P' \subseteq P$. The first factors of a decomposition of X, counted with multiplicity, form a multiset on $\mathbb{F}_2^e \setminus \{0\}$.

Lemma 7.6. *Let $K \subseteq \mathbb{F}_2^e$ be a subspace. In every decomposition of X, at least $\mathrm{R}(X/K)$ of the first factors lie outside K.*

Proof. By (3), applied to X and K, the terms whose first factors lie outside K give a decomposition of X/K. □

Let lb assign to each proper subspace $K \subsetneq \mathbb{F}_2^e$ an integer $\mathrm{lb}(K) \le \mathrm{R}(X/K)$. A multiset m on $\mathbb{F}_2^e \setminus \{0\}$ is *admissible* if $m(\mathbb{F}_2^e \setminus K) \ge \mathrm{lb}(K)$ for every proper subspace K. The *capacity* $\mathrm{cap}(X)$ is the least size of an admissible multiset. For $e = 0$ the empty multiset is admissible and $\mathrm{cap}(X) = 0$.

Proposition 7.7. *The first factors of every decomposition of X form an admissible multiset. Hence $\mathrm{R}(X) \ge \mathrm{cap}(X)$.*

Proof. For the multiset m of first factors, $m(\mathbb{F}_2^e \setminus K)$ is the number of first factors outside K, which is at least $\mathrm{R}(X/K) \ge \mathrm{lb}(K)$ by Lemma 7.6. A minimal decomposition gives an admissible multiset of size $\mathrm{R}(X)$. □

For $e \ge 1$ the subspace $K = 0$ gives the constraint $|m| \ge \mathrm{lb}(0)$, so $\mathrm{cap}(X) \ge \mathrm{lb}(0)$, and the constraints for $K \ne 0$ can raise the capacity further. Let K have codimension $j \ge 1$. The linear forms g that vanish on K are the forms on $\mathbb{F}_2^e/K$, and $X(g)$ is a quotient of X/K. If $j = 1$, then $X/K \cong X(g)$ for the nonzero such g, and we take $\mathrm{lb}(K) = \mathrm{rk}\, X(g) = \mathrm{R}(X/K)$. If $j \ge 2$, we take for $\mathrm{lb}(K)$ the maximum of the following lower bounds for $\mathrm{R}(X/K)$:

(i) the ranks $\mathrm{rk}\, X(g)$ for the nonzero forms g that vanish on K;
(ii) the *averaging bound* $\lceil 2^{1-j} \sum_g \mathrm{rk}\, X(g) \rceil$, where g runs over the $2^j - 1$ nonzero forms that vanish on K;
(iii) the three flattening ranks of X/K and, if $j \ge 3$, the bounds (1) with $p = 1$ for X/K and all surjections $\pi\colon \mathbb{F}_2^e/K \to \mathbb{F}_2^3$;
(iv) if $3 \le j < e$, the capacity of X/K, computed with the same rules for the quotients of X/K.

Bound (i) holds because rank does not increase under linear maps, (iii) is Section 2.4, and (iv) is Proposition 7.7 for X/K, whose quotients are quotients of X. For (ii), let $X/K = \sum_{l=1}^{r} \alpha_l \otimes \beta_l \otimes \gamma_l$ with $r = \mathrm{R}(X/K)$ and all $\alpha_l \ne 0$. For each g, the matrix $X(g) = \sum_l g(\alpha_l)\, \beta_l \otimes \gamma_l$ has rank at most the number of l with $g(\alpha_l) = 1$. Each nonzero α_l has $g(\alpha_l) = 1$ for exactly 2^{j-1} of the 2^j linear forms g on $\mathbb{F}_2^e/K$. Summing over g gives $\sum_g \mathrm{rk}\, X(g) \le 2^{j-1} r$.

The averaging bound counts points over $\mathbb{F}_2$, and it can exceed the border rank: for the tensor W of Example 2.4, the three nonzero forms on the first position give slices of ranks 2, 1 and 2, so (ii) with $K = 0$ gives $\mathrm{R}(W) \ge \lceil 5/2 \rceil = 3$, whereas $\underline{\mathrm{R}}(W) = 2$.

A capacity leaf asserts $\mathrm{cap}(X_U) \ge k - p_U$. We use such leaves only for $e \le 5$. The capacity is then the optimum of an integer program with at most 31 variables $m(p)$ and 373 constraints, one for each proper subspace of $\mathbb{F}_2^5$. The checker decides feasibility by a depth-first search over the variables with a node limit. A search that reaches the limit counts as undecided. Inside rule (iv) this only weakens a bound, so the capacities used are exact or lower bounds, and a capacity leaf, or the condition $\mathrm{cap}(X_U) \ge r$ of a rank leaf, counts only if the search proves it. A listing of the admissible multisets of a given size that reaches the node limit, or exceeds a cap on the number of instances, fails instead of being truncated.

Rank leaves. Let $r = k - p_U - 1$ and suppose that $e \le 5$ and $\mathrm{cap}(X_U) \ge r$. Then $\mathrm{R}(X_U) \ge r + 1$ unless X_U has a decomposition with exactly r terms. Such a decomposition is minimal, so its terms are distinct, and its first factors form an admissible multiset of size r by Proposition 7.7. A rank leaf therefore asserts that for every admissible multiset m of size r the following instance is

unsatisfiable: $X_U = \sum_{l=1}^{r} \alpha_l \otimes \beta_l \otimes \gamma_l$, where $\alpha_1, \ldots, \alpha_r$ are the points of m, repeated according to their multiplicities, and the β_l and γ_l are nonzero. Since the terms are distinct, we may require the terms with equal α_l to be in strictly increasing order of (β_l, γ_l), for a fixed total order. A SAT solver decides these instances.

Lifting leaves. A lifting leaf uses the whole tensor T. We use such leaves only for $\dim(A/U) \le 5$, so that the admissible multisets for T/U can be enumerated. Fix a lower bound $s_0 \le \mathrm{R}(T/U)$, for instance $\max(\mathrm{cap}(T/U),\ p_U + \mathrm{cap}(X_U))$. For an integer s and a multiset m of size s on $(A/U) \setminus \{0\}$, let $\mathrm{Lift}(s, m)$ be the statement:

> $T = \sum_{l=1}^{c-1} \alpha_l \otimes \beta_l \otimes \gamma_l$ with pairwise distinct terms, where exactly s of the α_l lie outside U, their images $\pi_U(\alpha_l)$ are the points of m, counted with multiplicity, and the other $c-1-s$ of the α_l lie in $U \setminus \{0\}$.

We call the s terms with $\alpha_l \notin U$ the outer terms and the others the inner terms. The statement requires exactly $c-1$ terms. A decomposition with fewer terms is brought to this size by splitting terms.

Lemma 7.8 (Padding)**.** *Let Δ be a list of pairwise distinct rank-one terms in $A \otimes B \otimes C$ with $|\Delta| \le c-2$, and suppose that $2^{\dim C - 1} \ge c-1$. Let $x \otimes y \otimes z$ be a term of Δ. Then there is a $z' \in C \setminus \{0, z\}$ such that replacing $x \otimes y \otimes z$ by the two terms $x \otimes y \otimes z'$ and $x \otimes y \otimes (z + z')$ gives a list of $|\Delta| + 1$ pairwise distinct rank-one terms with the same sum. The same holds with B in place of C.*

Proof. The pairs $\{z', z + z'\}$ with $z' \notin \{0, z\}$ partition $C \setminus \{0, z\}$ into $2^{\dim C - 1} - 1$ pairs. Over $\mathbb{F}_2$ a rank-one tensor determines its three factors. So a new term coincides with another term of Δ only if that term has the form $x \otimes y \otimes z''$ with z'' in the chosen pair, and each of the other $|\Delta| - 1 \le c-3$ terms of Δ excludes at most one pair. Since $2^{\dim C - 1} - 1 \ge c-2$, some pair is not excluded. Its two terms are nonzero, they differ from each other because $z \ne 0$, and they differ from the remaining terms of Δ. □

Proposition 7.9. *Suppose that $2^{\max(\dim B, \dim C) - 1} \ge c-1$, and that $\mathrm{Lift}(s, m)$ fails for every s with $s_0 \le s \le c-1-d$ and every admissible multiset m of size s for T/U. Then $N(U)$ holds.*

Proof. Suppose that $N(U)$ fails. Then some minimal decomposition Δ of T with $|\Delta| < c$ has a set E of terms that spans U. Let J be the set of terms of Δ whose first factors lie in U, so that $E \subseteq J$ and $|J| \ge d$. Let $s = |\Delta| - |J|$, and let m be the multiset of the points $\pi_U(\alpha_t)$ for the terms $t \notin J$. By (3) these s terms give a decomposition of T/U. Hence $s \ge \mathrm{R}(T/U) \ge s_0$, and m is admissible by Proposition 7.7. Moreover $s \le |\Delta| - d \le c-1-d$. If $|\Delta| = c-1$, then Δ satisfies $\mathrm{Lift}(s, m)$, because its terms are distinct.

If $|\Delta| < c-1$, we apply Lemma 7.8 repeatedly, in the larger of the second and third positions, until there are $c-1$ terms. If $J \ne \emptyset$, we always split a term of J. The two new terms have the same first factor, which lies in $U \setminus \{0\}$, so s and m do not change. If $J = \emptyset$, then $d = 0$, and we split terms with first factors outside U, which exist because $T \ne 0$. Splitting such a term with image p replaces (s, m) by $(s+1, m + \{p\})$. Adding a point keeps a multiset admissible, and at the end $s = c-1 = c-1-d$. In both cases the final list satisfies $\mathrm{Lift}(s, m)$ for some admissible m and some s in the stated range. This contradicts the hypothesis. □

The condition on $\dim B$ and $\dim C$ cannot be dropped. Let $T = e_0 \otimes e_0 \otimes e_0 \in \mathbb{F}_2 \otimes \mathbb{F}_2 \otimes \mathbb{F}_2$, $c = 3$ and $U = 0$. Since T is the only rank-one tensor in this space, no two terms can be distinct, so

every statement $\mathrm{Lift}(s, m)$ fails, and without the condition Proposition 7.9 would give $\mathrm{R}(T) \geq 3$. In our certificates $c - 1 \leq 17$, and the larger of $\dim B$ and $\dim C$ is at least 11, so the condition holds with a wide margin.

The statement $\mathrm{Lift}(s, m)$ is a SAT instance. In a basis of A whose first d vectors form the reduced echelon basis of U and whose other vectors are unit vectors mapping to a basis of A/U, an outer term has first factor $(x_l; p_l)$ with p_l prescribed by m and $x_l \in \mathbb{F}_2^d$ free, an inner term has first factor $(x_l; 0)$ with $x_l \neq 0$, and all β_l and γ_l are free nonzero vectors. Since the terms are distinct, the outer terms with equal p_l may be required to be in strictly increasing order of (x_l, β_l, γ_l), and the inner terms likewise.

The instance for (s, m) can often be refuted more cheaply through the *quotient relaxation*: $T/U = \sum_{l=1}^{s} p_l \otimes \beta_l \otimes \gamma_l$ with $p_1, \ldots, p_s$ the points of m and nonzero β_l, γ_l, where equal terms are allowed. If this relaxation is unsatisfiable, so is $\mathrm{Lift}(s, m)$, since π_U maps a solution of $\mathrm{Lift}(s, m)$ to a solution of the relaxation by (3). Two outer terms that differ only in x_l give equal terms of the relaxation, so we allow equal terms there. The proof of Proposition 7.9 would also work with relaxations that forbid equal terms. The decomposition it constructs has no two outer terms with the same image p and the same β_l and γ_l: in a minimal decomposition such terms would merge into one inner term, and padding never creates such a pair. If the relaxations for all (s, m) are unsatisfiable, then $\mathrm{R}(T/U) \geq c - d$. So lifting instances are needed exactly at the nodes where some relaxation has a solution.

The format of the DAG certificates and the steps of their check are given in Appendix B.5.

7.4 The subsets of C3

All four subsets of Table 3 are non-peelable. Koszul flattenings with $p = 2, 3$ give $\mathrm{R}(\Sigma_Y) \geq |Y| - 2$ for all four, that is, 16 for Y_1, Y_2, Y_3 and 15 for Y_4, and none of the flattening bounds (Section 2.4) that we know goes further. Substitution trees of up to five levels with randomized pivot orders did not settle these subsets. Their DAG certificates and the outcome of their check are described in Appendix B.4. The check itself is specified in Appendix B.5.

Theorem 7.10 (†)**.** *The subsets Y_1, Y_2, Y_3 and Y_4 of C3 in Table 3 have full rank. In particular $\mathrm{R}(\Sigma_{Y_4}) = 17$, and $\mathrm{R}(\Sigma_Y) = 18$ for the other subsets Y in the statement.*

Proof. The sum Σ_Y has $|Y|$ terms, so $\mathrm{R}(\Sigma_Y) \leq |Y|$. For the lower bound, Σ_Y is concise in the position used as A, and the nodes of its certificate (Table 7) that are reachable from 0 form a DAG certificate for $(\Sigma_Y, |Y|)$ whose leaf labels are all valid: the DAG checker recomputed every reachable node, checked its flattening and capacity leaves, and settled by the rank or the lifting rule every other reachable leaf (Appendix B.5). Theorem 7.5 gives $\mathrm{R}(\Sigma_Y) \geq |Y|$. □

In the certificate for Y_4, 122,003 nodes have two or more parents, and unfolded into a tree the certificate would have 803,089 nodes. At 491 of the lifting leaves no leaf that uses Lemma 7.2(b) can settle the node, because $\mathrm{R}(\Sigma_{Y_4}/U) = c - d - 1$ there: the checker found a decomposition of X_U with $k - p_U - 1$ terms, while $\mathrm{cap}(X_U) \geq k - p_U - 1$. At the other 70 lifting leaves $\mathrm{cap}(X_U) < k - p_U - 1$, which excludes the rank rule.

7.5 Relation to earlier work

Wang [38] gives an automated framework for certified rank lower bounds over finite fields, with forced products and substitution with backtracking, for formats up to $\langle 3, 4, 4 \rangle$. Forced products use rank-one slices of the tensor and enumerate the remaining coefficients, whereas peeling uses

the known terms of a subset and needs no enumeration. Two parts of our construction also occur in Wang's framework. Its substitution lemma [38, Lemma 3] eliminates first factors in arbitrary directions, several in turn: in our notation it passes from T/U to T/U_p for arbitrary points p, and Lemma 7.3 is the one-form step, stated for $N(U)$. Subspaces also index its dynamic program, which assigns certified lower bounds to the ranks of all restrictions of the first argument of a bilinear map, up to symmetry. In our notation such a restriction is a quotient T/U, whereas a DAG certificate is evaluated at exactly the nodes reachable from 0.

Bounds for small formats over $\mathbb{F}_2$ also combine capacity inequalities, by which the number of first factors of a decomposition outside a subspace is at least the rank of the quotient of the tensor by that subspace, with an enumeration of the multisets of first factors that satisfy all of them [38, 11, 39, 42, 37]. Lemma 7.6 and Proposition 7.7 are the capacity inequality and the admissible profiles of D'Ambrosio [11, Sec. 4, (8)] and Wang [39, Lemma 2.1]. The substitution with backtracking of Wang [38, Sec. 5.4] already combines the inequality, for subspaces spanned by first factors, with an exhaustive search over multisets of first factors. Yang et al. [42, Lemma 2.2] call the inequality the occupation inequality, and they exclude multisets of first factors by integer programming with Farkas-type certificates [42, Sec. 3.3]. The averaging bound for a subspace K of codimension j is their cover inequality [42, (3.6)–(3.7)] for the $2^j - 1$ hyperplanes that contain K, since every nonzero point of $\mathbb{F}_2^e/K$ lies in exactly $2^{j-1} - 1$ of them. Earlier, Shpilka's lower bounds for $n \times n$ matrices over finite fields [34] discard the products whose linear forms vanish on suitably chosen matrices (Lemmas 3 and 6 in the numbering of its arXiv text). Over $\mathbb{F}_2$ a code argument bounds the remaining products, and over $\mathrm{GF}(p)$ they are bounded by an inequality of the above kind for a three-dimensional space of matrices, due to Bläser [5] (Lemma 9 in the arXiv text of [34]). For a subspace K of codimension one, Lemma 7.6 is the correspondence between bilinear algorithms and linear codes [8, 27]. The first factors are the columns of a generator matrix, and the codeword of the form g that vanishes on K has weight at least $\operatorname{rk} X(g)$. For larger codimension, Lemma 7.6 bounds the support of a subcode, in the sense of the generalized Hamming weights of Wei [40].

The completion search of [37] decides instances like those of the rank leaves, in which all first factors are prescribed, by an exhaustive search for the second and third factors, and D'Ambrosio [11] mentions SAT instances with a fixed profile of first factors as diagnostics. In $\mathrm{Lift}(s, m)$ only the images of the first factors in A/U are prescribed, and the instance concerns T itself. D'Ambrosio [11, Sec. 5] and Wang [39, Sec. 4] exclude by hand the few multisets of first factors that satisfy all capacities in their problems, with arguments on the second and third factors, and Yang et al. [42, Sec. 1.1] refute the profile of first factors forced by their occupation bounds with relations between the factors of the whole tensor. We are not aware of earlier lower-bound arguments that propagate statements such as $N(U)$ instead of rank bounds for quotients, or that use instances such as $\mathrm{Lift}(s, m)$, in which the first factors are prescribed only modulo a subspace.

8 Verification

The statements marked with a dagger rest on computations: the search over G and the matching, the enumeration, the certificates produced by the deterministic certifying programs, the direct computation for the subset of C4 and the DAG certificates. Table 5 summarizes how the main statements are certified and checked.

8.1 Independent checks

Independently written checkers, developed from specifications separately from the enumeration and certifying programs, checked the certificates as listed in Table 5. The other further checks there come

Table 5: How the main computer-assisted statements are certified and checked. All further checks agree with the certification.

Statement	Certified by	Further checks	Lean
Theorem 3.1	(a) expansion; (b) the invariant ρ; (c) a search over G; (d) inspection	–	(a) proved
Remark 3.2	explicit equivalences and invariant hashes (Appendix B.1)	–	–
Table 2	the enumeration (Section 5.1)	count-only program; a separate sampler ($k = 16, 17, 18$), within two standard errors	–
LO(16), Remark 6.1	in-line tests and post-processing (Section 5.2)	enumeration without peeling (sizes 5 to 16), an independent checker on all of its post-processing certificates and a sample of its in-line ones	proved, 40 compiled evaluations
Theorem 1.1	as for LO(16), with Section 6.2 and Theorem 7.10 for nine orbits	all post-processing certificates, samples of the in-line ones, the four certificates of Table 4, all 2^{28} pencils for C4	proved; the subset of C4 kernel-only
Theorem 7.10	DAG certificates (Section 7)	DAG checker, with LRAT proofs checked by `lrat-check`	proved, with LRAT proofs checked in Lean

from a count-only program for the non-peelable subsets and an enumeration of all pencils for the subset of C4, both written from specifications in the same way, from a sampler that shares no code with the enumeration, and from an earlier enumeration program, run without peeling. The DAG checker counts an unsatisfiable instance only if `lrat-check` [41] verifies its LRAT proof [10], which CaDiCaL [4] produced, against the formula that was solved. Certificates other than the checked ones rely on the correctness of the certifying programs. For the DAG certificates the LRAT proofs certify that the formulas are unsatisfiable; the encoding of the instances, the node computations and the enumeration of the admissible multisets rest on the DAG checker, on the scripts that distributed its runs and carried results over between runs and certificate versions, and on the tests of these programs, which are in the archive (see Data and code availability).

8.2 Formal proofs in Lean

We formalize Theorems 3.1(a), 1.1 and 7.10, as well as local optimality to depth 16, in Lean 4 with Mathlib [12, 36]; on the way to depth 18 the development also proves local optimality to depth 17 of the five decompositions. Tensors are taken over $\mathbb{F}_2$, the rank of a tensor is the least r such that it is a sum of r outer products, and equivalence allows invertible linear maps of $\mathbb{F}_2^{16}$ on each factor, a permutation of the three factors and a reordering of the terms, so it contains the action of G. No proof uses `sorry`, Lean's placeholder for a missing proof. The kernel checks the flattening, Koszul and substitution bounds, the peeling lemma, the soundness of a certificate checker, the soundness of the enumeration with symmetry and universe pruning, and the transfer to equivalent lists. Lean runs its own enumeration, which follows Sections 4 and 5.1 but recomputes the universe matrices instead of updating them, so the theorems depend neither on the C++ enumeration nor on the contraction update. In the Lean runs a non-peelable set gets a Koszul certificate, a certificate from an in-line search or, at depths 17 and 18, a peeled-substitution certificate from a Lean version of the search of Section 5.2, and a proved checker verifies it. Data from the C++ searches enter only as certificates that proved checkers verify. A run may also accept, without a certificate, the sets in the orbits of finitely many listed sets, among them the nine orbits that remain after post-processing (Section 6.1).

Lean proves their full rank separately: by substitution certificates that proved checkers verify, by a proof for the subset of C4 that follows Section 6.2, and, for the subsets of C3, by the proof of Theorem 7.10. For that proof Lean formalizes substitution over subspaces with lifting (Section 7), with the standing assumption $A \neq 0$: it proves Lemma 7.3, Theorem 7.5, the leaf rules (Section 7.3) with the capacity bound and the search that computes it, Lemma 7.8 and Proposition 7.9, and it recomputes the nodes of each certificate as in Appendix B.5. It also proves that every decomposition described by a rank instance, a lifting instance or a relaxation gives a satisfying assignment of its CNF formula, so the encoding is not trusted. The refutations are LRAT proofs for CNF formulas generated by the same Lean function that the soundness proof uses, and the verified LRAT checker of Lean core checks them.

Apart from the evaluations by `native_decide`, which runs a decision procedure as compiled code, the kernel checks every step of the Lean proofs of this section, and each evaluation adds an axiom. The trusted base of a theorem with such axioms is therefore the Lean kernel and the compiled code of its evaluations. That code depends on the Lean compiler and runtime and, for precompiled modules, on Lean's C toolchain. In the proofs for the subsets of C3, the evaluated code includes the LRAT checker of Lean core, whose soundness is a theorem of Lean core. The proof for the subset of C4 is kernel-only, and so is the proof of local optimality to depth 8 for every decomposition equivalent to one of the five. The formalization does not cover parts (b)–(d) of Theorem 3.1, that is, the pairwise inequivalence of the five decompositions, the orders of their automorphism groups and the distinctness of their factors, nor that the Lean data for AT47 and KM47 are the decompositions of [14] and [29]. The symmetry pruning uses automorphisms checked by Lean.

9 Concluding remarks

Whether $\mathrm{R}_{\mathbb{F}_2}(\langle 4,4,4\rangle) = 47$ is itself a question of local optimality: it holds if and only if one, or equivalently every, decomposition S of rank 47 is locally optimal to depth 47, since $X = S$ sums to $\langle 4,4,4\rangle$ itself, and a set $X \subseteq S$ with $\mathrm{R}(\Sigma_X) < |X|$ would give a decomposition of $\langle 4,4,4\rangle$ with fewer than 47 terms. Is any of the five decompositions locally optimal to depth 19? If one of them is not, then $\mathrm{R}_{\mathbb{F}_2}(\langle 4,4,4\rangle) \leq 46$. What limits the depth is the number of non-peelable subsets, which grows roughly tenfold with each size from 11 to 18 in Table 2 and reaches about $6.3 \cdot 10^{10}$ to $6.7 \cdot 10^{10}$ per decomposition at size 18, weighted by orbit size, together with the cost of certifying all of them.

Do the sums of the four subsets of C3 in Table 3 have border rank below their size? We do not know. If they do, no flattening bound can prove that they have full rank, which would be consistent with the bound $|Y| - 2$ that Koszul flattenings give for them (Section 7.4).

Data and code availability

The archive at `https://doi.org/10.5281/zenodo.23652770` contains the five decompositions; the programs for the enumeration and the certification; the certificate checkers, two in Python and two in C++, with their specifications and tests; the count-only program, the sampler, the script for the bound for the subset of C4 and the independent program that checks it (Appendix B.3); the result tables; the stored certificates, which are samples of the depth-18 run, certificates of the hardest subsets and the DAG certificates (the three largest stored substitution certificates and the four DAG certificates are separate data files of the same record, listed in its `MANIFEST.md`); the search program for the DAG certificates and the DAG checker with its driver, specification, tests (listed in `repo/gsublift/checker/README.md`) and reports, together with per-instance outcome

tables derived from the journals of the checks; and the Lean development. The README files of the archive give the commands, the runs and their costs, the tests of the checkers, and the Lean theorems with their axioms.

Acknowledgment

This research received no specific funding.

Use of AI tools

Generative AI tools contributed to this work as follows. They did not contribute to choosing the topic, the rank of 4×4 matrix multiplication over $\mathbb{F}_2$, a long-standing open problem that the author chose to work on. They helped with the conceptual approach and made central technical contributions: they were used to find and prove the results, including the peeling lemma and the enumeration and certification methods, and to write the programs, the certificate checkers and the Lean formalization. They were also used for writing and exposition (drafting and revising the text), for checking for errors (reviewing the proofs, the programs and the text), and for other supporting tasks (searching the literature and running the computations). The author takes full responsibility for the content of the paper, including the parts produced with these tools.

References

[1] Abhinav Agarwal. Certified local rank and uniqueness barriers for a 48-term matrix-multiplication decomposition. arXiv:2609.13596, 2026.

[2] Boris Alexeev, Michael A. Forbes, and Jacob Tsimerman. Tensor rank: Some lower and upper bounds. In *2011 IEEE 26th Annual Conference on Computational Complexity (CCC)*, pages 283–291. IEEE, 2011.

[3] Yamato Arai, Yuma Ichikawa, and Koji Hukushima. Adaptive flip graph algorithm for matrix multiplication. In *Proceedings of the 2024 International Symposium on Symbolic and Algebraic Computation (ISSAC '24)*, pages 292–298. ACM, 2024.

[4] Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. CaDiCaL 2.0. In Arie Gurfinkel and Vijay Ganesh, editors, *Computer Aided Verification (CAV 2024)*, Lecture Notes in Computer Science, pages 133–152. Springer, 2024.

[5] Markus Bläser. A $\frac{5}{2}n^2$-lower bound for the rank of $n \times n$-matrix multiplication over arbitrary fields. In *40th Annual Symposium on Foundations of Computer Science (FOCS '99)*, pages 45–50. IEEE Computer Society, 1999.

[6] Markus Bläser. On the complexity of the multiplication of matrices of small formats. *Journal of Complexity*, 19(1):43–60, 2003.

[7] Markus Bläser. Fast matrix multiplication. *Theory of Computing Graduate Surveys*, 5:1–60, 2013.

[8] Roger W. Brockett and David Dobkin. On the optimal evaluation of a set of bilinear forms. *Linear Algebra and its Applications*, 19(3):207–235, 1978.

[9] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, 1997.

[10] Luís Cruz-Filipe, Marijn J. H. Heule, Warren A. Hunt, Matt Kaufmann, and Peter Schneider-Kamp. Efficient certified RAT verification. In Leonardo de Moura, editor, *Automated Deduction – CADE 26*, Lecture Notes in Computer Science, pages 220–236. Springer, 2017.

[11] Ian D'Ambrosio. The exact $\mathbb{F}_2$-rank of 2×3 by 3×4 matrix multiplication. Preprint, Zenodo, 2026.

[12] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction – CADE 28*, Lecture Notes in Computer Science, pages 625–635. Springer, 2021.

[13] Jean-Guillaume Dumas, Clément Pernet, and Alexandre Sedoglavic. A non-commutative algorithm for multiplying 4×4 matrices using 48 non-complex multiplications. arXiv:2506.13242, 2025.

[14] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis, and Pushmeet Kohli. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

[15] Marijn J. H. Heule, Manuel Kauers, and Martina Seidl. Local search for fast matrix multiplication. In Mikoláš Janota and Inês Lynce, editors, *Theory and Applications of Satisfiability Testing – SAT 2019*, Lecture Notes in Computer Science, pages 155–163. Springer, 2019.

[16] Marijn J. H. Heule, Manuel Kauers, and Martina Seidl. New ways to multiply 3×3-matrices. *Journal of Symbolic Computation*, 104:899–916, 2021.

[17] J. E. Hopcroft and L. R. Kerr. On minimizing the number of multiplications necessary for matrix multiplication. *SIAM Journal on Applied Mathematics*, 20(1):30–36, 1971.

[18] Manuel Kauers. Rank-47 multiplication schemes for 4×4 matrices over $\mathbb{Z}_2$. Web directory, `http://www.algebra.uni-linz.ac.at/people/mkauers/matrix-mult/solutions/444/x47/`, 2022. 99,101 scheme files dated 4 November 2022; accessed 23 October 2026.

[19] Manuel Kauers. Rank-47 multiplication schemes for 4×4 matrices over $\mathbb{Z}_2$, directory `47`. Web directory, `http://www.algebra.uni-linz.ac.at/people/mkauers/matrix-mult/solutions/444/47/`, 2022. 19 scheme files dated 6 to 29 October 2022; accessed 24 October 2026.

[20] Manuel Kauers and Jakob Moosbauer. Flip graphs for matrix multiplication. In *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation (ISSAC '23)*, pages 381–388. ACM, 2023.

[21] Manuel Kauers and Isaac Wood. Exploring the meta flip graph for matrix multiplication. arXiv:2510.19787, 2025.

[22] J. M. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2012.

[23] J. M. Landsberg. *Geometry and Complexity Theory*. Cambridge University Press, Cambridge, 2017.

[24] J. M. Landsberg and Mateusz Michałek. Abelian tensors. *Journal de Mathématiques Pures et Appliquées*, 108(3):333–371, 2017.

[25] J. M. Landsberg and Giorgio Ottaviani. Equations for secant varieties of Veronese and other varieties. *Annali di Matematica Pura ed Applicata*, 192(4):569–606, 2013.

[26] J. M. Landsberg and Giorgio Ottaviani. New lower bounds for the border rank of matrix multiplication. *Theory of Computing*, 11:285–298, 2015.

[27] Abraham Lempel and Shmuel Winograd. A new approach to error-correcting codes. *IEEE Transactions on Information Theory*, 23(4):503–508, 1977.

[28] Zeke Medley, Anagha Gokul, Quan Luu, and Panagiotis Manolios. GPU-accelerated search for fast matrix multiplication over $\mathbb{F}_2$. arXiv:2609.17533, 2026.

[29] Jakob Moosbauer. `flips` (GitHub repository). `https://github.com/jakobmoosbauer/flips`, 2022. Repository cited in [20]; contains `solutions/444-47-mod2.exp`.

[30] Jakob Moosbauer. *Search Techniques for Matrix Algorithms*. PhD thesis, Johannes Kepler University Linz, 2023.

[31] Jakob Moosbauer and Michael Poole. Flip graphs with symmetry and new matrix multiplication schemes. In *Proceedings of the 2025 International Symposium on Symbolic and Algebraic Computation (ISSAC '25)*, pages 233–239. ACM, 2025.

[32] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. arXiv:2506.13131, 2025.

[33] V. Ya. Pan. Methods of computing values of polynomials. *Russian Mathematical Surveys*, 21(1):105–136, 1966.

[34] Amir Shpilka. Lower bounds for matrix product. *SIAM Journal on Computing*, 32(5):1185–1200, 2003.

[35] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969.

[36] The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020)*, pages 367–381. ACM, 2020.

[37] Tejasvi Singh Tomar. The rank of 3×3 matrix multiplication over $\mathbb{F}_2$ is 23. arXiv:2610.11823, 2026.

[38] Chengu Wang. Automated lower bounds for bilinear complexity over finite fields. arXiv:2603.07280, 2026.

[39] Chengu Wang. A lower bound of 21 for 3×3 matrix multiplication over $\mathbb{F}_2$. arXiv:2609.06725, 2026.

[40] Victor K. Wei. Generalized Hamming weights for linear codes. *IEEE Transactions on Information Theory*, 37(5):1412–1418, 1991.

[41] Nathan Wetzler, Marijn J. H. Heule, and Warren A. Hunt. DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In Carsten Sinz and Uwe Egly, editors, *Theory and Applications of Satisfiability Testing – SAT 2014*, Lecture Notes in Computer Science, pages 422–429. Springer, 2014.

[42] Shuxing Yang et al. A structural proof of the lower bound 21 for 3×3 matrix multiplication over $\mathbb{F}_2$. arXiv:2609.18722, 2026.

[43] Nadim F. Zaru. Exact symmetry-stratified search for fast matrix multiplication over $\mathbb{F}_2$: A second inequivalent rank-47 scheme for 4×4, rigidity theorems, and large same-rank families at 5×5 and 6×6. Zenodo, version 1.0.1, 2026.

A The five decompositions

The five decompositions are in Table 6. A matrix $M = (m_{ij})$ with $0 \le i, j \le 3$ has the code $\sum_{i,j} m_{ij} 2^{4i+j}$ in hexadecimal, so the last digit encodes row 0, with bit j for column j. With $\langle 4,4,4 \rangle = \sum_{i,j,k} E_{ij} \otimes E_{jk} \otimes E_{ki}$, each of the five sums to $\langle 4,4,4 \rangle$ over $\mathbb{F}_2$.

B Matching, post-processing, pencils and DAG certificates

Command lines, file names, hashes and the details of the runs and tests are in the README files of the archive (see Data and code availability).

B.1 Matching of public and searched decompositions of rank 47

We matched the decompositions in the following sources with the five classes by explicit equivalences: the decomposition of Fawzi et al. [14], the file `444-47-mod2.exp` of the repository [29] of [20], the two collections on the web site of Kauers [19, 18], the record of Zaru [43], and copies of AT47 and KM47 in later catalogues and artifacts. As lists of terms, AT47 and KM47 are the decompositions of [14] and [29]. We analysed the collection [18] as an archive of 99,101 files from the same web site. Up to the order of their terms, these files contain only four distinct decompositions. Two of them lie in the class of AT47 and two in that of KM47. Of the 19 files of the collection [19], 13 lie in the class of KM47, five in that of AT47, and one file in that of C3. The second decomposition of rank 47 of Zaru [43], given there in the cyclic convention with the same encoding of matrices, lies in the class of C5. No decomposition in these sources is equivalent to C4.

Our flip-graph searches consisted of about 226,000 descents of several kinds: plain descents from the standard algorithm, descents from decompositions glued from smaller formats, descents restricted to decompositions with a symmetry of order 2 or 3, and descents after adding 2, 3, 6 or 12 extra terms to a known decomposition. They produced about 8.8 million decompositions of rank 47, counted with multiplicity except that the first 41,809 descents kept only distinct ones, and no descent reached rank 46. The 712,634 distinct decompositions from those first descents were all matched to their classes by explicit equivalences. Of the later decompositions, about 8.1 million, a random sample of 4,083 was matched in the same way. The invariant hash used for the rest is computed from the matrix ranks of the factors and of the sums of pairs of factors in each position, minimized over the six permutations of the positions, and G preserves these data.

Table 6: The five decompositions. Row t gives the codes of a_t, b_t, c_t for each decomposition.

	AT47			KM47			C3			C4			C5		
t	a_t	b_t	c_t	a_t	b_t	c_t	a_t	b_t	c_t	a_t	b_t	c_t	a_t	b_t	c_t
0	0004	0100	0013	0008	edee	baa8	0555	9b90	9ef1	00b0	ccac	1e3d	0001	0001	0001
1	0064	0890	1002	7048	6d66	bba8	94d3	b994	929d	df62	0343	9909	2000	2212	0078
2	0054	0405	0102	d000	38b3	8008	c0cc	8990	6606	3333	fc0f	d0d0	9800	b282	b000
3	0044	0d95	0002	0409	0303	4554	a00a	8a20	5e0d	9b22	0353	55c9	0800	1888	b004
4	0001	0001	0105	98a0	62b9	4ea8	9579	0008	eee0	0090	1cac	0033	0230	4272	0b00
5	0101	039b	0004	6000	6db6	0800	1001	202a	00e0	9944	1206	44dd	c06e	0200	2850
6	0501	0302	0014	0400	0ed3	4958	0cc0	0dd0	dd0d	ff42	fc4c	3330	953e	4280	2840
7	0367	0090	1000	0159	0202	4444	6acc	8d90	9609	1c91	bc0c	3300	950d	0008	1000
8	3002	0045	0108	540a	0dd0	4d58	0222	fdf0	9e4d	5055	6006	45dd	4500	4d84	7000
9	6002	0320	0018	98c0	b2b9	0220	6066	8a90	9e01	2022	0060	a536	903b	0002	0010
10	2002	0375	0008	5000	3de3	4558	944b	202d	9edd	ff72	fc7f	7eed	4600	4484	7800
11	3057	0005	0100	0800	c213	4aa8	844a	505d	0e00	bb00	3053	158d	a800	2282	b040
12	6507	0300	0010	070a	0210	4551	8468	0558	d00d	cccc	6c06	fedd	9080	b288	00d0
13	1000	3046	0c08	0739	0209	0202	955c	9090	9e0e	cc00	1071	f566	0100	4d8d	0004
14	9008	3002	0cd8	0049	0308	1302	2ec2	8dd0	d60d	9f22	0303	ccc0	0537	0004	0100
15	8008	7402	0ca8	704e	0b00	1100	1011	b993	0060	9922	fc53	40dd	9b30	4282	0040
16	0301	0098	1004	520a	0220	4440	0110	022f	00d0	cc41	fc08	dddd	2030	4274	0070
17	9a08	3000	84d0	0850	1213	4664	c02c	8880	ee86	2f92	fc0c	dd0d	0004	0100	7d05
18	0a08	3020	c0d4	075f	0200	0111	c00c	8080	ee0e	9c55	1706	55cc	0240	4444	0f00
19	0808	b820	c064	0d00	0213	4004	853d	9b91	9e61	1ca1	cc0c	2e0d	a008	2202	b0c8
20	0020	58c0	a002	f8a0	92b9	0888	95c3	0994	0d0d	cc72	fcb3	9000	c008	8088	a0d8
21	00a8	5040	ab02	340a	0bb0	0808	0ccc	9d90	9909	dd00	1271	ee7d	7530	4284	0800
22	c008	5400	0ba8	000d	0300	1001	0cca	4d90	9d0d	9499	5606	11cd	4000	8188	7008
23	c058	0400	0320	9000	68b3	46a8	c06c	8190	6e0e	3300	3073	9090	c080	8288	20d0
24	4828	5820	8020	5008	3033	8998	e22c	8d80	9e09	ec52	0cbc	3033	2060	4244	0870
25	0400	9a02	0064	07a0	0219	4602	0111	9b96	0090	d09d	1606	ffcc	0040	4144	0d02
26	0088	d048	a602	0700	021f	4357	9569	0998	6000	2222	fc9f	ae3d	a60e	0080	3840
27	0098	9008	7602	0050	121e	4557	95c2	9994	9990	9c99	ac06	10dd	0540	4d88	0d00
28	6a08	0020	8090	0009	030e	4657	9009	2020	9e3d	cc50	1cbd	a033	0c04	0888	6d04
29	c0a8	5000	8b20	0006	0d00	0100	84c2	c994	0600	0050	1c71	4edd	0cae	4000	2940
30	0008	1000	7bdf	9400	08b3	4c08	0cc2	ddd0	994d	2c92	0c0c	c000	4800	8888	f000
31	0c08	9800	7064	0060	6269	0200	9099	b990	92fd	9900	1253	0099	0002	0010	0b73
32	1110	000e	0400	70a0	9219	2aa8	0660	fd20	9e0d	9099	0606	00cc	0032	4276	0b72
33	30a8	0040	8900	5800	3213	4cc8	0aaa	fd90	920d	0030	cc7c	3030	0062	4044	0a72
34	8880	e000	8420	000a	0de0	4c58	955a	909d	9000	d0dd	6606	eecd	0008	1000	b0d9
35	0c68	0800	3020	7028	9299	2220	953c	9091	0e0e	cc9c	1c06	eedd	0c40	4888	2d00
36	4440	0e00	0020	9ea0	02b9	4400	95c9	0990	6d0d	9999	fc06	00dd	0604	4404	7c04
37	2220	00e0	8000	00a0	6219	4ca8	95cc	9990	99f6	af11	3003	851d	0200	4414	0b04
38	0002	0010	1009	9040	b8b3	8aa8	0550	0220	5e3d	2092	0cac	f033	1000	b28b	0008
39	0c98	9000	7420	1048	b0bb	8888	955b	000d	00dd	dd40	1242	aa39	2080	2222	00f0
40	4000	5620	00a8	0759	020f	4757	0ee0	fdd0	924d	2292	fcac	ee3d	00a2	0222	0b62
41	0010	940c	0602	00d0	1219	2002	8578	0228	5eed	ec62	0070	65f6	02a0	4222	0b40
42	0398	0008	5400	0058	1211	2331	2022	8aa0	5e8d	9c11	4703	9900	95d0	4288	2000
43	1288	3048	8400	700a	0b80	1908	a66a	8d50	de0d	9c22	fc03	d0dd	0080	1222	00d2
44	0200	30a8	c004	0600	02df	0400	e00e	8a80	9e81	4444	6506	55cd	0010	4277	0002
45	8418	9402	0420	7000	6de6	b3a8	842c	9c91	9009	0040	1171	55c6	9008	b28a	b0d8
46	9508	0002	0450	0040	656e	b9a8	952c	9991	0066	dc51	0707	cccc	0504	4d8c	7d04

B.2 Post-processing

At depth 16, 20,866 orbit representatives went to post-processing. At depth 18 there were 14,229,163. Of these, 14,085,423 were settled in the first stage of post-processing (Section 5.2), 143,645 in the second and 86 in the third. Post-processing left nine orbit representatives (Section 6.1).

B.3 The pencils for the subset of C4

Besides our script for the bound (Section 6.2), an independent program enumerates all 2^{28} matrices c in Gray-code order. For a pencil $e_0 \otimes s_0 + e_1 \otimes s_1$ it uses the bound $\max(\mathrm{rk}[\, s_0 \; s_1 \,],\ \mathrm{rk}\left[\begin{smallmatrix} s_0 \\ s_1 \end{smallmatrix}\right])$. The ranks of s_0, s_1 and $s_0 + s_1$ never exceed this bound, because the column space of $[\, s_0 \; s_1 \,]$ contains their column spaces. Its minimum over all 2^{28} pencils is 4, attained by 25 matrices c, among them $c = 0$, so the bound $\mathrm{R}(\Sigma_X) \geq 18$ has no margin. Like the count 57 in Section 6.2, the count 25 refers to the choice of b_0 and b_1 made there.

B.4 The DAG certificates

In the files of Y_1, Y_2 and Y_3, a coarsening after the search put leaves at 13, 21 and 42 nodes of codimension 5 in place of the substitution labels recorded there by the search: 13 SAT leaves (leaves settled by a rank or lifting rule) for Y_1, 10 capacity and 11 SAT leaves for Y_2, and 24 capacity and 18 SAT leaves for Y_3. Every line of an unreachable node in these files is for a subspace of codimension 4 that contains one of these leaves. In the certificate for Y_4, all 1,678 SAT leaves have codimension 4.

Table 7: The DAG certificates for Y_1, Y_2, Y_3 and Y_4. The abbreviations SUB, CAP, OPT, RANK and LIFT stand for substitution labels, capacity leaves, SAT leaves, the rank rule and the lifting rule. OPT leaves are counted by the rule that settles them in the check, and instances are counted over the OPT leaves of each certificate.

	Y_1	Y_2	Y_3	Y_4
Position used as A, and dim A	third, 10	first, 9	first, 9	first, 9
Nodes reachable from 0	1,506,516	380,442	268,164	454,575
SUB nodes	36,499	15,157	11,391	16,766
Flattening leaves	1,409,094	328,848	226,085	408,074
CAP leaves	57,630	34,772	28,377	28,057
OPT leaves	3,293	1,665	2,311	1,678
OPT leaves settled by RANK	1,541	1,057	1,469	1,117
by LIFT, through relaxations only	0	4	11	1
by LIFT, with full instances	1,752	604	831	560
Instances of RANK leaves	16,257	22,914	34,352	30,029
LIFT instances refuted through the relaxation	145,805	66,461	99,404	73,956
LIFT instances refuted directly	74,815	25,353	26,910	24,602

B.5 Certificate formats and their checks

Here X, k and the tests 1–3 are as in Section 5.2. A certificate for X is a tree written in pre-order, with leaves of six kinds. A TRIV leaf asserts a target of at most 0, and an NZ leaf a target of at most 1 for a nonzero tensor. The Koszul flattenings with $p = 1, 2, 3$ (Section 2.4) are labelled KOS, YNG and YNG7, where 7 is the dimension of V for $p = 3$. FLAT, KOS, YNG and YNG7 leaves name a position and, except for FLAT, a projection π. The inner nodes of the tree are SUB nodes, which name a position and a pivot and have one child for each branch of (2). A SUB node is valid only if the pivot slice of the current tensor, taken after peeling in the peeled format, is nonzero. A check recomputes the tensor at each node from the scheme, the index list of X and the path from the root. At a leaf with target k it requires the matrix rank to be at least k, $2k-1$, $6(k-1)+1$ or $20(k-1)+1$, respectively. For FLAT this is the flattening bound, and for the other three these are the conditions of (1). For peeled substitution, the check first peels the list at every node and lowers the target by the number of peeled terms. A certificate for one of the fast tests 1–3 is a single leaf.

The DAG certificates (Section 7) have the following format. A certificate file lists the nodes, one per line. A line gives the reduced row echelon basis of U in the coordinates of A, followed by the label: SUB with the form f in the coordinates of A/U that come from the non-pivot coordinates of the echelon basis, a flattening leaf with its position and projections as in Section 5.2, CAP, or OPT. A node labelled OPT is a leaf that the checker settles by a RANK or a LIFT argument. The file contains no multisets and no proofs. The coordinates of A, B and C are the greedy reduced coordinates (Section 5.2) of Σ_Y for the order of Table 3.

From the decomposition C3, the term list of Y and the certificate, a checker recomputes the following.

(C1) The tensor T in reduced coordinates, with a check that its terms are distinct with nonzero factors and that T is concise in the position used as A.

(C2) For each node reachable from 0: the canonical form of the echelon basis, the children U_p of a SUB node, which must be present in the file, the list L_U, its core and the tensor X_U. Lines of nodes that are not reachable from 0 are parsed and checked for canonical form, but not evaluated.

(C3) The flattening ranks of the flattening leaves and the capacities of the CAP leaves.

(C4) For each OPT leaf: the admissible multisets, from its own implementation of the capacity bound, and the CNF formulas of all RANK and LIFT instances and relaxations. The leaf is valid if $\mathrm{cap}(X_U) \geq k - p_U - 1$ and every RANK instance is unsatisfiable, or if every instance $\mathrm{Lift}(s, m)$ of Proposition 7.9 is unsatisfiable or has an unsatisfiable relaxation. For LIFT the checker takes $s_0 = \max\big(\mathrm{lb}(0),\, p_U + \mathrm{cap}(X_U)\big)$, where $\mathrm{lb}(0)$ is the bound for $\mathrm{R}(T/U)$ itself (Section 7.3). No admissible multiset has size below $\mathrm{cap}(T/U)$, so this gives the same instances as $s_0 = \max\big(\mathrm{cap}(T/U),\, p_U + \mathrm{cap}(X_U)\big)$.

(C5) The padding condition of Proposition 7.9 at every LIFT leaf.

The checker we used is described in Section 8.

Finding certificates. We found the certificates with a memoized depth-first search over subspaces, with a fixed position as A. At a node without a flattening or capacity leaf, the search tries the children U_p. As soon as the children it cannot settle span A/U, the node fails. Otherwise every nonzero form f that vanishes on them is a valid pivot, and the search takes one for which the fewest children are SUB or OPT nodes. The search stops at codimension 4. A node of codimension 4 without a flattening or capacity leaf becomes an OPT leaf, provided that its numbers of admissible multisets are small enough for the SAT stage. The files of Y_1, Y_2 and Y_3 were then coarsened at some nodes of codimension 5 (Appendix B.4).