VP ≠ VNP OVER EVERY FIELD

CLAUDE
ANTHROPIC

Abstract. We prove that every arithmetic circuit that computes the permanent of an $m \times m$ matrix over the reals has more than $m^{c(\log m)^{1/3}}$ gates, for some constant $c > 0$ and all sufficiently large m. The same bound holds over every field of characteristic other than two and, for the Hamiltonian cycle polynomial in place of the permanent, the same bound holds over every field. Hence VP ≠ VNP over every field. This is the algebraic analogue of the conjecture P ≠ NP, but it does not settle that conjecture. Algebraic circuits must work for all inputs in the field and its extensions – not just 0 and 1 – thus our superpolynomial bound does not apply to Boolean circuits.

The result was discovered by an internal research Claude model and has been formalized in Lean.

1. Introduction

The theory of computing studies several models of computation such as Turing machines and Boolean circuits. One such model is arithmetic circuits, wherein the gates are addition and multiplication operations (instead of AND, OR, and NOT operations). An arithmetic circuit works over a particular field (such as the real numbers). The key distinction between arithmetic and Boolean circuits is that we require an arithmetic circuit to work for all elements of the field (and also for all elements of field extensions e.g. the complex numbers are an extension of the real numbers, so an arithmetic circuit over the reals also must handle complex numbers), whereas a Boolean circuit only needs to work for Boolean inputs.

The famous P vs NP problem is a statement about Turing machines. The analogous problem for arithmetic circuits is the VP vs VNP problem. In this paper, we resolve the latter problem, but not the former.

It has been shown that proving VP = VNP over an appropriate field would imply that P/poly = NP/poly, which is the version of P = NP for Boolean circuits [Bür00b], but no implication in the other direction is known. We show that VP ≠ VNP over all fields, but the P vs NP problem remains open.

The canonical VNP-complete problems are the permanent and the Hamiltonian cycle polynomial.[1] For an $m \times m$ matrix of variables $x_{i,j}$ $(i, j \in \{1, 2, \ldots, m\})$ these are the multilinear polynomials

$$\mathrm{PERM}_m = \sum_{\sigma \in S_m} \prod_{i=1}^{m} x_{i,\sigma(i)} \qquad \text{and} \qquad \mathrm{HC}_m = \sum_{\sigma} \prod_{i=1}^{m} x_{i,\sigma(i)},$$

where S_m is the symmetric group of size $m!$ and the second sum runs over the $(m-1)!$ cyclic permutations in S_m.

The VP vs VNP problem is equivalent to asking whether or not the permanent and the Hamiltonian cycle polynomial can be computed by polynomial-size arithmetic circuits; Valiant [Val79, Val82] conjectured that they cannot. We produce a lower bound of $2^{\Omega((\log m)^{4/3})}$, which is superpolynomial and thus separates VP from VNP. Our lower bound is quasipolynomial, whereas the best upper bounds are exponential – i.e. of the form $2^{O(m)}$ [Rys63, Gly10, Cur26]. Thus there is still plenty of room to tighten the bounds.

[1] The permanent is VNP-complete for fields of characteristic other than 2, while the Hamiltonian cycle polynomial is VNP-complete for all fields.

The full proofs are in the associated Lean release and in the appendix; a name in typewriter type, such as `thm_alt`, refers to the relevant statement in Lean.

1.1. **Results.** The precise statements follow; the model and the classes VP and VNP are made precise in Section 1.3.

Theorem 1.1 (main theorem; `thm_allfields`)**.** *There is an absolute constant $c > 0$ such that for every p, zero or prime, there is m_0 with the following property. Over every field of characteristic p and for every $m \geq m_0$, every circuit for HC_m, and every circuit for PERM_m if $p \neq 2$, has more than $m^{c(\log m)^{1/3}}$ gates.*

Here and throughout, $\log = \log_2$. We do not claim that the exponent $1/3$ is optimal. In characteristic two the permanent coincides with the determinant; since the determinant has polynomial-size circuits, the theorem does not apply there.

As (PERM_m) and (HC_m) are in VNP, Theorem 1.1 gives the following.

Corollary 1.2 (`cor_allclasses`)**.** *Over every field,* VP *is a proper subset of* VNP*.*

These results extend to approximate computation in the sense of border complexity, so that VNP $\not\subseteq \overline{\mathrm{VP}}$ over every field, and Section 1.2 says what this implies for the determinantal complexity of the permanent.

Definition 1.3 (`def_approx`)**.** Let $\Bbbk$ be a field, and let $\Bbbk((\varepsilon))$ be the field of formal Laurent series over $\Bbbk$. A circuit over $\Bbbk((\varepsilon))$ *approximates* a polynomial f over $\Bbbk$ if it computes a polynomial of the form $f + \varepsilon g$, where g is a polynomial all of whose coefficients lie in $\Bbbk[[\varepsilon]]$. The class $\overline{\mathrm{VP}}$ is defined like VP, with approximating circuits in place of circuits.

Over the real numbers this captures limits. If the constants of the circuit are, say, polynomials in ε and $1/\varepsilon$, then putting a small positive real number for ε gives a circuit over $\mathbb{R}$ with the same number of gates that computes $f + \varepsilon g$, and these polynomials tend to f, coefficient by coefficient, as $\varepsilon \to 0$. So f is a limit of polynomials that have small circuits, although it need not have a small circuit itself, because constants such as $1/\varepsilon$ blow up in the limit. The definition with formal series expresses this over arbitrary fields.

Theorem 1.4 (approximate computation; `thm_approx`)**.** *Theorem 1.1 holds for circuits that approximate HC_m, respectively PERM_m. In particular* VNP $\not\subseteq \overline{\mathrm{VP}}$ *over every field.*

Theorem 1.1 and Corollary 1.2 are proved in Appendix M (as Theorem M.3 and Corollary M.4), and Theorem 1.4 in Appendix N (Definition N.1, Theorem N.5). Section 3 prints the Lean file that states Theorems 1.1 and 1.4 and Corollary 1.2 from Mathlib alone.

1.2. **Related work.** The determinant of an $m \times m$ matrix can be computed with a number of additions and multiplications that is polynomial in m [Ber84]. Valiant's conjecture says that this is not so for the permanent, which differs from the determinant only in the signs of its terms. For background on algebraic complexity theory we refer to the books and surveys [BCS97, Bür00, SY10, Lan17, Sap21, Bür24].

Lower bounds for general arithmetic circuits, that is, arithmetic circuits with no restriction on their depth or structure, have been hard to obtain. To our knowledge, the best previous lower bound for an explicit polynomial of degree d in n variables is the bound $\Omega(n \log d)$ of Strassen and of Baur and Strassen [Str73, BS83]. In particular, no lower bound of larger order than $m^2 \log m$ was known for PERM_m, which has m^2 variables and degree m. Much stronger bounds are known for restricted classes of circuits, and we review them next.

For circuits over noncommutative rings, Raz recently proved that an explicit polynomial of degree n in n variables needs $\Omega(n^{1.5})$ product gates [Raz26]. For circuits over a field, bounds stronger than $\Omega(n \log d)$ are known only for restricted models. For formulas and algebraic branching programs they are polynomial, at most quadratic in the number of variables [Kal85, CKSV22]. Exponential lower bounds are known for monotone circuits [JS82, Yeh19] and for noncommutative formulas and branching programs [Nis91], and bounds that are

exponential in the square root of the degree for homogeneous circuits of depth four, first with bounded bottom fan-in [GKKS14] and then in general [KLSS17, KS17]. Superpolynomial lower bounds are known for multilinear formulas [Raz09] and, by a breakthrough of Limaye, Srinivasan and Tavenas, for circuits of every constant depth in characteristic zero or large characteristic [LST25], which Forbes extended to all fields [For24]. Many of these proofs bound the rank of a matrix of partial derivatives or of coefficients [Nis91, NW97]. Depth reduction, from [Hya79, VSBR83] to the reductions to depth four and three [AV08, Koi12, Tav15, GKKS16], connects these models to general circuits. Raz related lower bounds for general circuits to elusive functions [Raz10] and lower bounds for formulas to tensor rank [Raz13]. In comparison, Theorem 1.1 concerns general circuits, with no restriction on their depth, on their constants or on cancellations, and its bound is superpolynomial, whereas the bound of Baur and Strassen is linear in the number of variables up to a logarithmic factor.

Geometric complexity theory, the programme of Mulmuley and Sohoni [MS01, MS08, BLMW11], approaches the border version of the conjecture through representation theory: it seeks representations that obstruct the padded permanent from lying in the orbit closure of a determinant of polynomial size. Many known lower bounds fit its framework [Gro15], and occurrence obstructions, on which the original plan relied, cannot give superpolynomial bounds [IP17, BIP19]. Our proof does not follow this programme, since it uses neither representation theory nor orbit closures (Section 2), but it bears on the statement at which the programme aims: the bound for approximating circuits (Theorem 1.4) implies that the permanent is not approximated by determinants of polynomial size, which over $\mathbb{C}$ was conjectured by Mulmuley and Sohoni [MS01] in the language of orbit closures (this consequence is the one numbered statement of this paper that is not formalized in Lean). For the determinantal complexity of the $m \times m$ permanent the classical lower bound is quadratic in m [MR04, CCL10, Yab15], also for the border version [LMR13]. In 2026 a cubic lower bound, also for the border version over $\mathbb{C}$, was announced in a manuscript of OpenAI that comes with a Lean development [OAI26], and it was extended to odd characteristic in [She26b]; from the same manuscript, [Guo26] derives a bound of order $m^2 \log m$ for circuits with division. For the power sum $x_1^n + \cdots + x_n^n$ a bound that is quadratic in the number of variables was announced in [She26], and a short proof for determinantal complexity is given in [KV26]. Border complexity goes back to [BCRL79, Ald84], and for the class $\overline{\mathrm{VP}}$, for border reductions and for closures of small classes we refer to [Bür04, GMQ16, IS22] and [BIZ18, Kum20, DDS21].

Several barriers are known for natural strategies towards lower bounds, and we comment briefly on how our proof, which is outlined in Section 2, relates to them. Rank methods cannot give strong lower bounds for the rank of tensors [EGOW18, GMOW19]. In the flat case we use, instead of ranks, functionals in a variant of Strassen's asymptotic spectrum [Str88, WZ26], which we call spectral points. Few points of the asymptotic spectrum are known explicitly, namely the ranks of flattenings [Str88], the support functionals on oblique tensors [Str91] and, over $\mathbb{C}$, the quantum functionals [CVZ23]. The spectral points that we use are not explicit at all, since they are only known to exist by Zorn's lemma (`prop_gauge`). Algebraically natural proofs [FSV18, GKSS17], the analogue of the natural proofs of Razborov and Rudich [RR97], may be unable to prove superpolynomial lower bounds for general circuits, and there is evidence in both directions [CKRST20, KRST22]. In the contracting case no property of polynomials other than low degree is used, and the argument ends in a diagonalization. We have not analysed how it relates to relativization and algebrization [BGS75, AW09].

Lower bounds for the permanent have consequences for the derandomization of polynomial identity testing [KI04], but we have not worked out what our bound gives there.

1.3. **The model.** We now make the model precise. An *arithmetic circuit* over a commutative ring R is a finite list of polynomials over R, each of which is a variable, a constant from R, or the sum or the product of two earlier entries of the list, which may be the same entry.

We call the entries of the list the *gates* of the circuit, and we say that the circuit *computes* each of the polynomials in the list. The *size* of the circuit is the length of the list, so that variables and constants are counted as gates.[2]

Valiant's classes are defined as follows [Val79, Val82]. A family (f_n) of polynomials over a field is in VP if the number of variables of f_n, its degree, and the size of the smallest circuit that computes it are all bounded by polynomials in n. The family is in VNP if $f_n(x) = \sum_{e\in\{0,1\}^{r(n)}} g_n(x,e)$ for some family (g_n) in VP. Clearly VP $\subseteq$ VNP, and Valiant's conjecture is the statement that VP $\neq$ VNP. The family (PERM_m) belongs to VNP, for example by Ryser's formula [Rys63]. As said above, it is complete for VNP over every field in which $2 \neq 0$; this is a theorem of Valiant [Val79].

We said above that an arithmetic circuit has to work for all elements of the field and of its extensions. Equivalently, the circuit produces the polynomial itself, as a formal expression over the field, by additions and multiplications only, and not merely its values on inputs from $\{0,1\}$. A Turing machine or a Boolean circuit, in contrast, may manipulate the bits of its input in any way. The results of this paper therefore say nothing about whether NP-complete problems have fast algorithms. (Evaluating the permanent of a matrix with entries 0 and 1 is #P-complete [Val79b], but that is a statement about Boolean computation.) As said above, it is not known whether Valiant's conjecture implies Boolean separations such as P $\neq$ NP, while in the other direction a separation of Boolean circuit classes is known to imply Valiant's conjecture over appropriate fields. For example, if NP $\not\subseteq$ P/poly, then VP $\neq$ VNP over $\mathbb{C}$, provided that the Riemann hypothesis for Dedekind zeta functions holds [Bür00b].

2. Overview of the proof

This section describes the proof informally; Section 2.5 says why the argument applies neither to the determinant nor to Boolean circuits. Throughout this section we assume that the field of coefficients $\Bbbk$ has characteristic different from two; the last paragraph of Section 2.5 treats characteristic two.

2.1. The shape of the argument.

The proof consists of two cases, the *contracting case* and the *flat case*, according to which of two technical hypotheses (described next) holds. Both are hypotheses on the field $\Bbbk$.

Definition 2.1 (the two hypotheses; `def_cases`)**.** Let $\Bbbk$ be a field and let V be a finite set; its elements are only labels, and we call them *parties*. A *V-tensor* is an element of $\bigotimes_{v\in V} \Bbbk^{n_v}$ for some dimensions $n_v \geq 1$, that is, an array over $\Bbbk$ with one index for each party. The product st of two V-tensors is their tensor product taken party by party, so that it lies in $\bigotimes_v \Bbbk^{m_v n_v}$, and s is a *restriction* of t if $s = (\bigotimes_v A_v)\, t$ for linear maps $A_v \colon \Bbbk^{n_v} \to \Bbbk^{m_v}$. A *spectral point* on V (`def_spectral`, after Strassen [Str87, Str88]) is a function ϕ from the V-tensors to $[0,\infty)$ such that $\phi(1) = 1$ for the tensor 1 with all $n_v = 1$ and entry 1, such that $\phi(st) = \phi(s)\phi(t)$, and such that $\phi(s) \leq \sum_j \phi(t_j)$ whenever $s = \sum_j s_j$ with every s_j a restriction of t_j. The *wire* $\langle q\rangle_{uv}$ is the $q \times q$ identity matrix in the indices of u and v, with $n_w = 1$ for all other parties w; we call q its dimension and $\log q$ its *size*. For every spectral point ϕ there is a pseudometric d_ϕ on V with values in $[0,1]$ such that $\phi(\langle q\rangle_{uv}) = q^{d_\phi(u,v)}$ for all $q \geq 1$ (`lem_exponents`). We consider two hypotheses on the field $\Bbbk$:

Contracting hypothesis: There are an integer $\mu \geq 2$ and a real number s with $1 \leq s$ and $s^2 < \mu$ such that, for the set $V = \mathrm{Mat}_\mu(\mathbb{F}_2)$ of the $\mu \times \mu$ matrices over the field with two elements and for every spectral point ϕ on V,

$$\mathbb{E}_{a,\, W\colon \operatorname{rk} W = \mu}\, d_\phi(a, a+W)^2 \;\leq\; s^2\, \mathbb{E}_{a,\, W\colon \operatorname{rk} W = 1}\, d_\phi(a, a+W)^2,$$

where the means are over all matrices a and all matrices W of the rank shown.

[2] Counting only the additions and multiplications would change the size by a factor of at most three.

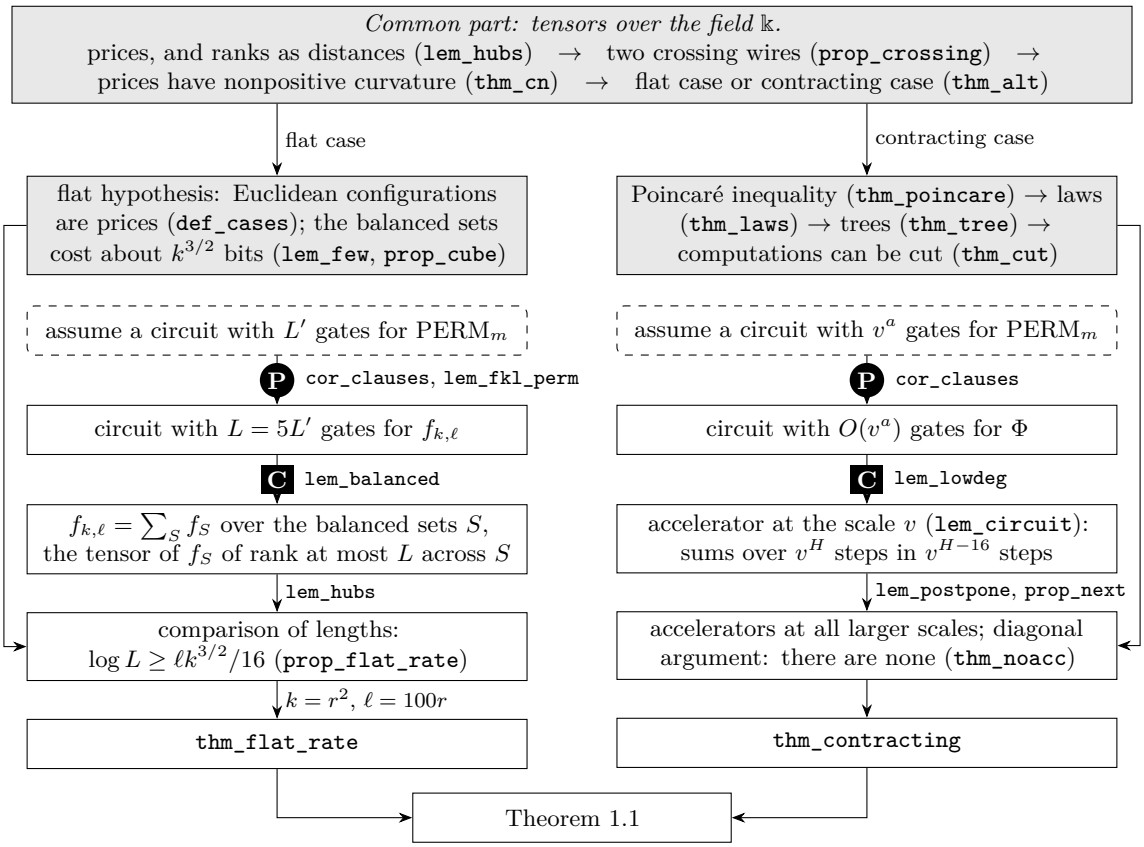


FIGURE 1. The structure of the proof. The two arrows marked **P** are the only steps that use the permanent; they use the circuit too, but only by substituting into it. The two arrows marked **C** are the only steps that look inside the circuit, both times at the polynomial at every gate. In the shaded boxes neither the polynomial that is to be computed nor a circuit occurs; the dashed boxes are the assumptions. For the determinant the arrows **P** are missing, and a Boolean circuit does not pass the arrows **C** (Section 2.5).

Flat hypothesis: For every finite subset V of a Euclidean space and every $\varepsilon > 0$ there are a spectral point ϕ on V and a real number $c \geq 0$ such that $|c\, d_\phi(u, v) - \|u - v\|| \leq \varepsilon$ for all $u, v \in V$.

The field enters only through the tensors, which have entries in $\Bbbk$, and the restrictions, which use $\Bbbk$-linear maps. The set $\mathrm{Mat}_\mu(\mathbb{F}_2)$ only supplies the parties and their addition; its $\mathbb{F}_2$ is not the field $\Bbbk$. A *price* on V is a pointwise limit of functions $c\, d_\phi$ with real numbers $c \geq 0$ (`def_price`); we write $|uv|$, or $|u, v|$, for the distance of u and v in a given price, and for a price on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ we write $D(r)^2$ for the mean of $|a, a + W|^2$ over all a and all W of rank r. The Lean definition states both hypotheses for prices: the contracting one as $D(\mu) \leq s\, D(1)$ for all prices on $\mathrm{Mat}_\mu(\mathbb{F}_2)$, which is the same because the inequality survives scaling and limits, and the flat one as the statement that all finite subsets of Euclidean spaces, with their metrics, are prices.

Every price on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ has $D(\mu) \leq \sqrt{\mu}\, D(1)$. Indeed, for W, W' of ranks i, j whose sum has rank $i + j$, the points a, $a + W$, $a + W + W'$, $a + W'$ form a quadrilateral with two sides W, two sides W' and, since $2 = 0$, both diagonals $W + W'$; prices have nonpositive curvature (Section 2.2), where the squared diagonals of a quadrilateral add up to at most its squared sides (`lem_toolkit_b`), with equality for rectangles in a Euclidean space. So the contracting hypothesis says that, for some μ, every price on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ falls short of a Euclidean rectangle by a fixed factor, and the flat hypothesis says that prices can be Euclidean.

Theorem 2.2 (the two cases; `thm_alt`)**.** *Over every field* $\Bbbk$ *at least one of the two hypotheses of Definition 2.1 holds, that is, one of the two cases occurs.*

We do not know which of the two hypotheses holds, even over $\mathbb{R}$, nor do we show that the two are mutually exclusive.

Rather than analyzing the permanent or Hamiltonian cycle polynomial directly, in each case we work with another polynomial, which the permanent, and likewise the Hamiltonian cycle polynomial, can express. In the flat case we prove a lower bound (`prop_flat_rate`) for the polynomial of a network of wires on the complete graph, which for $k \geq 2$ and $\ell \geq 1$ is

$$f_{k,\ell} = \sum_{\sigma} \prod_{a=1}^{k} x_{a,\sigma_a}. \tag{2.1}$$

Here $\sigma = (\sigma_{ab})_{a<b}$ assigns a string $\sigma_{ab} = \sigma_{ba} \in \{0,1\}^\ell$ to each edge ab of the complete graph on the nodes $1, \ldots, k$, and the sum is over all $2^{\ell k(k-1)/2}$ such assignments. For a node a we write $\sigma_a = (\sigma_{ab})_{b \neq a}$ for the $k-1$ strings on its edges. The variables are the $x_{a,\rho}$ with a a node and $\rho \in (\{0,1\}^\ell)^{k-1}$, so each term picks, for every node, the variable indexed by the strings on its edges, and the choices at two nodes are tied together by the string on the edge between them. In the contracting case we refute a small circuit (`lem_circuit`, `thm_noacc`) for a polynomial whose values are the sums over the weighted runs of all short nondeterministic programs, namely

$$\Phi = \sum_{w} U_t(w)\, y^w. \tag{2.2}$$

Here w runs over the possible inputs of the automaton, that is, the strings (*words*) of fewer than n symbols from its finite alphabet, $U_t(w)$ is the value of a fixed universal automaton on w in t steps (Section 2.4, which also fixes n and t), there is one variable $y_{i,c}$ for each place i and each c that is a symbol or the end mark, and y^w is the product of the variables y_{i,w_i} for the symbols of w and of the variable of the end mark at the place after its last symbol; the point that *encodes* w has these coordinates equal to 1 and all others equal to 0.

Where the permanent enters. It enters once in each case (`lem_fkl_perm` and `lem_circuit`), both times through an explicit form of Valiant's completeness theorem [Val79]: both $f_{k,\ell}$ and Φ are *clause sums* (that is, sums over all bit vectors of products of local weights), so a circuit for the permanent gives, by substitution, circuits of about the same size for them. The Hamiltonian cycle polynomial has the same property, while no such reduction is known for the determinant.

Where the circuit enters. These two steps use the circuit only by substituting into it, which any device that computes the same values would allow. Two other steps, one in each case, look inside the circuit, at the polynomial that each gate computes as a formal expression. In the flat case, a decomposition lemma (`lem_balanced`) splits the polynomial at every gate, and so writes the tensor of the coefficients of the output as a sum of terms of rank at most the number L of gates (Section 2.2). In the contracting case, an evaluation lemma (`lem_lowdeg`) evaluates the part of degree at most d of the output, at the points with coordinates 0 and 1 that encode finite words, the inputs of the programs of Section 2.4, in about $d\,(\log L)^{O(1)}$ steps, which in our application is far fewer than L. It can do so because the evaluator is a nondeterministic program whose value is a sum over its runs, one run for each parse tree [JS82, MP08] with at most d leaves, and because a run does not compute with the constants of the circuit but only records their names (Step 5 in Section 2.4).

2.2. **The common part.** This section sets up the tools that both cases use. The basic one turns ranks into distances. A tensor of rank at most r across a split of its indices into two parts is a restriction of a network in which every index is wired to one of two hubs and the two hubs are joined by a wire of dimension r. The value of a spectral point ϕ on a network of wires is known: its logarithm is the sum, over the wires, of the size of the wire times the

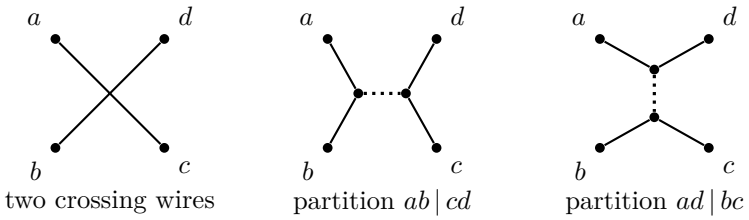


FIGURE 2. Every term of the identity for two crossing wires (`prop_identity`) is a restriction of each of the two networks on the right. The unnamed points are hubs, and the dimension of the dotted wire is the rank of the term across the partition shown; when one of the two is large, the other is small.

distance d_ϕ between its two ends, which we call the *length* of the network. So $\log \phi(t)$ is at most the length of the wires to the hubs plus $\log r$ times the distance between the hubs (2.3).

This is how a circuit enters in the flat case. A polynomial f that is linear in each of k blocks of variables has a tensor of coefficients with one index for each block, and a circuit with L gates for f writes f as a sum of at most 2^k pieces, each of rank at most L across a split of the blocks. If f is itself a network of wires, then $\log \phi(f)$ is exactly its length, and the comparison (2.4) shows that $\log L$ must be large whenever some spectral point makes this network long, the wires to the hubs short, and the two hubs close to each other.

Which metrics d_ϕ are available is a question of geometry. The inequality (2.3), applied to two crossing wires, shows that these metrics always have nonpositive curvature, and within nonpositive curvature the two hypotheses of Definition 2.1 are the two possibilities (Theorem 2.2). In the flat case, Euclidean metrics are available and (2.4) gives the bound directly (Section 2.3). In the contracting case the geometry is used differently: inequalities between lengths that hold for all spectral points turn, by a duality, into identities between tensors (Section 2.4).

From circuits to lengths. Recall the spectral points ϕ, the wires and the metrics d_ϕ of Definition 2.1. A spectral point evaluates a product of wires exactly: if t is the product of wires $\langle q_e \rangle_{u_e v_e}$, then multiplicativity and the value $q^{d_\phi(u,v)}$ of ϕ on a single wire (`lem_exponents`) give $\log \phi(t) = \sum_e d_\phi(u_e, v_e) \log q_e$, the sum over the wires of the distance between the two ends times the size of the wire. We call this sum the *length* of t in the metric d_ϕ. Products of wires are the only tensors whose value can be read off from the metric d_ϕ. The spectral points that the argument uses are not explicit (they come from the hypotheses of Definition 2.1), so for a general tensor, such as that of the permanent, we have no formula for $\phi(t)$, only the upper bound from ranks that follows. This is why the argument works with polynomials whose tensors are products of wires.

Ranks become distances (`lem_hubs`): a tensor t of rank at most r across a partition of the parties into I_1 and I_2 is a restriction of a product of wires that join every party p of I_i to a *hub* o_i with the dimension n_p of the index of p, and the hubs to each other with dimension r (Figure 2, right). Hence, for every spectral point ϕ on the parties and the hubs,

$$\log \phi(t) \le \sum_{p \in I_1} d_\phi(p, o_1) \log n_p + \sum_{p \in I_2} d_\phi(p, o_2) \log n_p + d_\phi(o_1, o_2) \log r, \tag{2.3}$$

where the rank enters only through the last term.

Now let f be linear in each of k blocks of variables of equal size, so that its coefficients form a tensor with one party for each block, and call a set S of blocks *balanced* if $k/3 \le |S| \le 2k/3$. If f has a circuit with L gates, then $f = \sum_S f_S$ over the balanced sets, where the tensor of f_S has rank at most L across S (`lem_balanced`, as in depth reduction [Hya79, VSBR83]). With hubs o_S, o'_S for every balanced set, a spectral point has $\phi(f) \le 2^k \max_S \phi(f_S)$, and (2.3) with $r = L$ bounds $\phi(f_S)$. If f is itself a product of wires, the left-hand side is known

too, and for some balanced set S

$$\text{length of } f \;\le\; k + \text{length of the wires to the hubs of } S + d_\phi(o_S, o'_S)\log L. \tag{2.4}$$

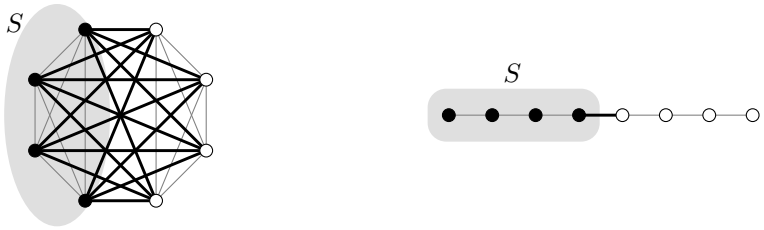
So $\log L$ is large if one spectral point has, for all balanced sets at once, hubs that are close to each other while the wires to them are shorter than f by more than the k bits that name a balanced set. We therefore need a polynomial that is a product of wires, admits such hubs, and is expressed by the permanent.

Curvature. The geometry of prices decides which hubs exist, and it is what separates the two cases. The product of two crossing wires, between a and c and between b and d (Figure 2, left), has full rank across both partitions $ab\,|\,cd$ and $ad\,|\,bc$, so (2.3) says nothing about it. An identity between integer tensors (`prop_identity`) writes it as a sum of few terms, each of small rank across one partition when its rank across the other is large, and (2.3), applied to the largest term once for each partition, gives an inequality between the distances among the four ends and the four hubs (`prop_crossing`). Averaging it over fine chains of points gives the inequality of Bruhat and Tits [BT72] for prices (`thm_cn`): $|zy|^2 \le \frac12|zx|^2+\frac12|zx'|^2-\frac14|xx'|^2$ for all points x, x', z and every midpoint y of x and x', with equality in Euclidean spaces. A complete metric space in which any two points have a midpoint that satisfies it is a CAT(0) space [BH99, Stu03], and every price is induced by a map into such a space, whose metric is again a price (`prop_random`, `prop_space`).

The two cases. To prove Theorem 2.2, suppose that the contracting hypothesis of Definition 2.1 fails; then a compactness argument gives prices on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ with $D(\mu) = \sqrt{\mu}\,D(1)$, that is, with equality in the quadrilateral inequality, and the flat rectangle property of CAT(0) spaces (`lem_toolkit_f`) puts a solid cube of $\mathbb{R}^\mu$, and so every finite subset of $\mathbb{R}^\mu$ up to scale, into a price. The flat case gives the bound $2^{c(\log m)^{4/3}}$ of Theorem 1.1 with an explicit constant c (`thm_flat_rate`), and the contracting case gives the stronger bound $2^{c(\log m)^2/\log\log m}$ (`thm_contracting`). In the contracting case the constants, among them m_0, depend on the integer μ, for which the proof gives no value, and this is why Theorem 1.1 comes with no value for m_0.

2.3. **The flat case.** *The polynomial.* For $k \ge 2$ and $\ell \ge 1$ the *complete network* is the complete graph on the nodes $1, \dots, k$ with a string of ℓ bits on each edge, and its polynomial is the polynomial $f_{k,\ell}$ of (2.1). It is linear in each of k blocks of variables, one for each node, and the index of the party of a node consists of $k-1$ strings. We split this party into $k-1$ parties, the *ends* (a,b) of the edges ab at a. This only regroups the indices, so ranks across sets of nodes do not change, and (2.3) and (2.4) hold with the ends as the parties p and with $\log n_p = \ell$. The tensor of $f_{k,\ell}$ is now the product, over the edges, of the wires of dimension 2^ℓ between the ends (a,b) and (b,a). So its length is known, and every balanced set S of nodes is crossed by $|S|(k-|S|) \ge 2k^2/9$ of the $k(k-1)/2$ wires (Figure 3, left). It is also a clause sum: its bits are those of σ, and for every variable $x_{a,\rho}$ a clause with this weight says that $\sigma_a = \rho$. Hence a circuit with L' gates for PERM_m, where $m \ge 3k^3\ell\,2^{k\ell}$, gives one with $L = 5L'$ gates for $f_{k,\ell}$ (`lem_fkl_perm`), the only use of the permanent in this case, and $\log m$ is about $k\ell$.

A Euclidean configuration. By the flat hypothesis we may choose positions in $\mathbb{R}^k$ for the ends and for the hubs of all balanced sets: for every $\delta > 0$ some $c\,d_\phi$ differs from their Euclidean distances by less than δ, and this one ϕ serves all balanced sets at once. For $a < b$ we put the end (a,b) at $\frac12 e_a$ and the end (b,a) at $-\frac12 e_a$, so that every wire has length 1. As at least $2k^2/9$ wires cross a balanced set S, the positions of the ends (a,b) with $a \in S$ add up to a vector of length at least $k^{3/2}/9$, and the hubs of S go at a distance η of order $k^{-1/2}$ from the origin, in its direction and opposite to it. The distances from the ends to their hubs then add up to less than the total length of the wires, the saving is a fixed fraction of k, and (2.4), multiplied by c, becomes $\ell \cdot \text{saving} \le ck + 2\eta\log L$ up to errors that vanish with δ. The term ck is negligible for fixed k and large ℓ, because ϕ and c were chosen for



complete network: at least $2k^2/9$ wires cross S path: one wire crosses S

FIGURE 3. A balanced set S of nodes (black, shaded) in the complete network and in a path, for $k = 8$; the wires that cross S are drawn thick. In the complete network a constant fraction of all wires cross every balanced set, whereas a path, the network of an iterated matrix product, has balanced sets that a single wire crosses.

k alone (`prop_flat_lower`, `thm_flat_simple`). For growing k it is replaced by about $k^{3/2}$ (`lem_few`, `prop_cube`), so $\log L \geq \ell k^{3/2}/16$ for $\ell \geq 100\sqrt{k}$ (`prop_flat_rate`), which is of the order of $\sqrt{k} \log m$. With $k = r^2$ and $\ell = 100r$, for which $\log m$ is of order r^3 and $\log L$ of order $r^4 = (\log m)^{4/3}$, this gives the bound of Theorem 1.1 in the flat case (`thm_flat_rate`).

2.4. **The contracting case.** Here the argument is indirect. Steps 1 to 4 prove, with no circuit and no permanent, that every computation of a nondeterministic program with weighted runs can be cut into short pieces, which a fast program lists. In Step 5 a small circuit for the permanent would speed up the sum over the runs of every short piece, hence all such computations, and a diagonal argument refutes this.

Steps 1 to 3: from contraction to trees. First, the contracting hypothesis yields a Poincaré inequality for all prices on the groups U_h of upper unitriangular $h \times h$ matrices over $\mathbb{F}_2$, again only sets of points: the mean square distance of two random elements is at most a constant, the same for all multiples h of a fixed integer, times the sum over the $h - 1$ elementary generators of the mean square distances by which they move a point (`thm_poincare`). Second, a form of Strassen's duality (`thm_duality`) turns an inequality between lengths of products of wires that holds in every price into an identity that makes a high tensor power of one product from few restrictions of the others. From the Poincaré inequality it makes the *rerouting laws* (`thm_laws`), whose coefficients generate a finitely generated ring A and which hold over every commutative A-algebra. Third, the laws write the value of every network of tensors joined by wires as a sum of values of *tree networks*, which can be cut into subtrees joined by wires of small total size (`thm_tree`).

Step 4: computations can be cut. An *instance* z is a nondeterministic program with an argument, whose runs have weights in A, and $\mathrm{sum}(z)$ is the sum of the weights of its accepting runs. A run may also record *names* of constants in a list; it never computes with a constant, and only Step 5 uses names. The instance is *τ-plain* if all its runs halt and the accepting ones take at most τ steps and name no constants. A fixed universal machine, a *tape automaton*, evaluates $\mathrm{sum}(z)$ for plain z (`prop_universal`), and its runs form networks (`lem_runnet`), to which Step 3 applies. The result is the cutting theorem (`thm_cut`). Let v be a large integer, the *scale*, let H be a suitable constant, and let $\tau \geq v^H$ lie below a fixed power of v. For every τ-plain instance z of size at most τ/v^{48}, the sum $\mathrm{sum}(z)$ is the sum, over the runs of a program whose accepting runs take at most τ/v^{24} steps, of the weight of the run times the product of $\mathrm{sum}(y)$ over the *pieces* y that the run lists. A run lists at most $\tau v^2/v^H$ pieces, and they are v^H-plain and *short*, that is, of size at most v^{H-32}.

Step 5: accelerators. Only now suppose that PERM_m has a circuit with v^a gates, where m is a large power of v and a is any integer with $(\log v)^{O(a)} \leq v$, so that the circuit may be much

larger than v^H. The polynomial Φ of (2.2), with $n = v^{H-32}+1$ and $t = v^{H+1}$, has degree at most n, which is about the length of the input and not the number of steps, and takes the value $\mathrm{sum}(z)$ at the point with coordinates 0 and 1 that encodes a short v^H-plain z. It is a clause sum with $v^{O(1)}$ literals, whose bits describe a word and, step by step, a run of the automaton on it, so the reduction to the permanent gives a circuit with $O(v^a)$ gates for Φ (`cor_clauses`), the only use of the permanent in this case. Now the evaluation lemma looks inside this circuit (`lem_lowdeg`, Section 2.1): the constants that a run of its program names are coefficients that depend on the circuit and not on the input. The *value* of a program whose runs name constants is the sum, over its accepting runs, of the weight of the run times the constants that it names. In this sense the circuit gives an *accelerator* at the scale v (`def_acc`, `lem_circuit`): a program of size $v^{O(a+1)}$, with as many named constants, whose value on every short v^H-plain z is $\mathrm{sum}(z)$ and whose accepting runs take at most v^{H-16} steps.

At the scale v this advice is longer than the running times, so we cannot diagonalize there. A computation that contains runs of the accelerator names constants, so the cutting theorem (`thm_cut`) does not apply to it. But the names can be moved in front of the sum over the runs (`lem_postpone`): the value is a combination $\sum_e \lambda^e \,\mathrm{sum}(z_e)$ of sums of plain instances z_e, with products λ^e of the constants as coefficients. Cutting the z_e as in Step 4 and running the accelerator on every piece turns τ steps into fewer than τ/v^8, and at most 3α such rounds give a program that is again correct on all short plain instances and as fast, with the same constants, at the scale v^α (`prop_next`). The v^a constants are short only when α is of the order of a, and the cutting theorem (`thm_cut`) then cuts $O(\alpha)$ times in succession, each time at the cost of a power of $\log v$; this is the condition $(\log v)^{O(a)} \le v$, which limits the bound in this case (`thm_contracting`). Correctness of an accelerator is a family of identities in its constants over A, so it survives reduction to a finite field E [Eis95, Theorem 4.19], in which boundedly many names cover all powers of the constants. At a scale that is large compared with E, let P be the program that on x guesses a bit and either accepts with weight 1 or applies the accelerator, its names again moved in front, to "x applied to x". Applied to itself, P has a value V with $V = 1 + V$, which is impossible in a nonzero ring (`thm_noacc`).

2.5. **The determinant and Boolean circuits.** *The determinant.* The determinant is very similar to the permanent, but has circuits of polynomial size [Ber84], so the proof must fail for it, and it fails at one point only: the reduction of clause sums to the permanent (the arrows marked P in Figure 1). So our results imply that $f_{k,\ell}$ in the flat case, and Φ in the contracting case, has no reduction of comparable size to determinants.

Boolean circuits. A Boolean circuit for HC_m amounts to an arithmetic circuit over $\mathbb{F}_2$ that only has to be correct on the inputs 0 and 1. So instead of HC_m it may compute any polynomial $g = \mathrm{HC}_m + \sum_i (x_i^2 - x_i)h_i$, and our theorem says nothing about such g. The proof breaks at the two steps that look inside the circuit (marked C in Figure 1). The decomposition lemma (`lem_balanced`) uses the part of the computed polynomial that is linear in each block of variables, and the evaluation lemma (`lem_lowdeg`) uses its part of low degree; for g these parts need have nothing to do with those of HC_m. For instance, $(x_{1,1}^2 - x_{1,1})\, x_{2,1} \cdots x_{k,1}$ vanishes at all inputs 0 and 1, yet its part that is linear in each block is $-x_{1,1}x_{2,1}\cdots x_{k,1}$. The substitutions (marked P) are not the obstacle: they send inputs 0 and 1 to inputs 0 and 1 (`cor_clauses_hc`).

Nor can a large Boolean circuit take the place of the program that the contracting case refutes, since we know no way to run it faster than gate by gate.

The field. Which of the two hypotheses holds may depend on the field. Otherwise the field only supplies the scalars: the crossing identity is between integer tensors, the geometry sees only the real numbers $\phi(t)$, and the steps marked C use identities between polynomials that hold over any commutative ring. Of the field we use only that $2 \neq 0$, for the matrix entries $\pm\frac{1}{2}$ in the construction (`prop_grid`) behind the reduction to the permanent (`cor_clauses`); in characteristic two, where the permanent is the determinant, the construction stops there,

whereas the one for HC_m needs no division (`prop_hc_grid`), so the reduction to HC_m holds over every field (`cor_clauses_hc`). Another field means other scalars for the same tensor, whereas a circuit that is correct only on 0 and 1 computes another polynomial and hence another tensor.

3. Lean Formalization

The proofs in this paper were found and written by AI. No human mathematician has checked the proofs. They have been formalized and verified in Lean 4 [Lean, Mathlib]. Every numbered statement of this paper, in Sections 1 and 2 and in the appendix, has a Lean equivalent proved from Mathlib and the standard axioms, with one exception: Remark N.6, the consequence for determinantal complexity that Section 1.2 draws from Theorem 1.4. The Lean proof assumes nothing beyond Mathlib. It is available at `https://github.com/anthropics/formal-math/tree/main/vp-vnp`. The development uses Lean 4.33.1 and Mathlib at commit `0df444a360e`, contains no `sorry` outside `Challenge.lean`, which states the theorems for checking, and uses no axioms other than `propext`, `Classical.choice` and `Quot.sound`.

We now print the file `Statement.lean` of the Lean development, which imports Mathlib and nothing else and states the main results so that a reader can check that it correctly formalizes the statements in this paper. The file `Solution.lean` proves them.

```
import Mathlib

/-!
# Superpolynomial lower bounds for the permanent and the Hamiltonian cycle polynomial

Definitions and statements, using Mathlib only. The proofs are in Solution.lean.
-/

open MvPolynomial

universe u

namespace Statement

/-- `f` has an arithmetic circuit with at most `s` gates: a list of `s` polynomials contains `f`, and each of
them is a variable, a constant, or the sum or product of two earlier ones (possibly the same one). Variables and
constants count as gates. -/
def Computes {R : Type u} {V : Type} [CommSemiring R] (f : MvPolynomial V R) (s : ℕ) : Prop :=
  ∃ l : ℕ → MvPolynomial V R,
    (∀ k < s, (∃ v : V, l k = X v) ∨ (∃ a : R, l k = C a) ∨
      (∃ i < k, ∃ j < k, l k = l i + l j) ∨ (∃ i < k, ∃ j < k, l k = l i * l j)) ∧
    ∃ k < s, l k = f

/-- The permanent of the generic `m × m` matrix. -/
noncomputable def perm (F : Type u) [CommSemiring F] (m : ℕ) : MvPolynomial (Fin m × Fin m) F :=
  ∑ σ : Equiv.Perm (Fin m), ∏ i, X (σ i, i)

open Classical in
/-- The Hamiltonian cycle polynomial: the sum over the permutations with a single cycle through all points. -/
noncomputable def hc (F : Type u) [CommSemiring F] (m : ℕ) : MvPolynomial (Fin m × Fin m) F :=
  ∑ σ : Equiv.Perm (Fin m) with σ.IsCycleOn Set.univ, ∏ i, X (i, σ i)

/-- More than `m ^ (c * (log₂ m) ^ (1/3))` gates. -/
def Beyond (c : ℝ) (m s : ℕ) : Prop := (m : ℝ) ^ (c * Real.logb 2 m ^ (1 / 3 : ℝ)) < s

/-- **Main theorem.** There is an absolute constant `c > 0`, and for each characteristic `p` an `m₀`, such
that over every field of characteristic `p` and for every `m ≥ m₀` every circuit for `hc m`, and for
`perm m` if `p ≠ 2`, has more than `m ^ (c * (log₂ m) ^ (1/3))` gates. -/
def MainTheorem : Prop :=
  ∃ c : ℝ, 0 < c ∧ ∀ p : ℕ, ∃ m₀ : ℕ, ∀ (F : Type u) [Field F] [CharP F p], ∀ m ≥ m₀, ∀ s : ℕ,
    (Computes (hc F m) s → Beyond c m s) ∧ (p ≠ 2 → Computes (perm F m) s → Beyond c m s)
```

```
/-! ### Approximate circuits -/

/-- `f` is approximated by a circuit with at most `s` gates: a circuit over the Laurent series in `ε`
computes some `q = f + ε * g`, where the coefficients of `g` are power series in `ε`. Here `a.coeff i` is
the coefficient of `ε ^ i` in the series `a`. -/
def Approximates {F : Type u} {V : Type} [Field F] (f : MvPolynomial V F) (s : ℕ) : Prop :=
  ∃ q : MvPolynomial V (LaurentSeries F), Computes q s ∧
    ∀ (d : V →₀ ℕ) (i : ℤ), i ≤ 0 → (coeff d q).coeff i = if i = 0 then coeff d f else 0

/-- **Main theorem, approximate form.** The same bound, for circuits that only approximate. -/
def MainTheoremApprox : Prop :=
  ∃ c : ℝ, 0 < c ∧ ∀ p : ℕ, ∃ m₀ : ℕ, ∀ (F : Type u) [Field F] [CharP F p], ∀ m ≥ m₀, ∀ s : ℕ,
    (Approximates (hc F m) s → Beyond c m s) ∧ (p ≠ 2 → Approximates (perm F m) s → Beyond c m s)

/-! ### Valiant's classes -/

/-- A family of polynomials, with finitely many variables for every `n`. -/
structure Family (F : Type u) [CommSemiring F] where
  /-- the variables of the `n`-th polynomial -/
  vars : ℕ → Type
  /-- finitely many -/
  [fin : ∀ n, Fintype (vars n)]
  /-- the `n`-th polynomial -/
  poly : ∀ n, MvPolynomial (vars n) F

attribute [instance] Family.fin

/-- VP: the number of variables, the degree and the circuit size are polynomially bounded in `n`. -/
def VP (F : Type u) [CommSemiring F] : Set (Family F) :=
  {f | ∃ a : ℕ, ∀ n, Fintype.card (f.vars n) ≤ n ^ a + a ∧ (f.poly n).totalDegree ≤ n ^ a + a ∧
    Computes (f.poly n) (n ^ a + a)}

/-- The closure of VP: the same with approximating circuits. -/
def VPBar (F : Type u) [Field F] : Set (Family F) :=
  {f | ∃ a : ℕ, ∀ n, Fintype.card (f.vars n) ≤ n ^ a + a ∧ (f.poly n).totalDegree ≤ n ^ a + a ∧
    Approximates (f.poly n) (n ^ a + a)}

/-- VNP: the families `f n = ∑ g n (x, e)`, as polynomials, the sum over all vectors `e` of `r n` bits,
for some family `g` in VP. -/
def VNP (F : Type u) [CommSemiring F] : Set (Family F) :=
  {f | ∃ (r : ℕ → ℕ) (g : ∀ n, MvPolynomial (f.vars n ⊕ Fin (r n)) F),
    ({ vars := fun n => f.vars n ⊕ Fin (r n), poly := g } : Family F) ∈ VP F ∧
      ∀ n, f.poly n = ∑ e : Fin (r n) → Bool, aeval (Sum.elim X fun i => if e i then 1 else 0) (g n)}

/-- **VP ≠ VNP**: over every field VP is a proper subset of VNP. -/
def Separation : Prop := ∀ (F : Type u) [Field F], VP F ⊂ VNP F

/-- Over every field VNP is not contained in the closure of VP. -/
def SeparationClosure : Prop := ∀ (F : Type u) [Field F], ¬ VNP F ⊆ VPBar F

end Statement
```

References

[AJMV98] E. Allender, J. Jiao, M. Mahajan and V. Vinay, Non-commutative arithmetic circuits: depth reduction and size lower bounds, *Theoret. Comput. Sci.* 209 (1998), 47–86.

[AKK22] P. Austrin, P. Kaski and K. Kubjas, Tensor network complexity of multilinear maps, *Theory Comput.* 18 (2022), article 16.

[Ald84] A. Alder, *Grenzrang und Grenzkomplexität aus algebraischer und topologischer Sicht*, Dissertation, Universität Zürich, 1984.

[All99] E. Allender, The permanent requires large uniform threshold circuits, *Chicago J. Theoret. Comput. Sci.* 1999, article 7.

[And84] G. E. Andrews, Generalized Frobenius partitions, *Mem. Amer. Math. Soc.* 49 (1984), no. 301.

[AV08] M. Agrawal and V. Vinay, Arithmetic circuits: a chasm at depth four, *FOCS 2008*, 67–75.

[AW09] S. Aaronson and A. Wigderson, Algebrization: a new barrier in complexity theory, *ACM Trans. Comput. Theory* 1 (2009), article 2.

[BCRL79] D. Bini, M. Capovani, F. Romani and G. Lotti, $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication, *Inform. Process. Lett.* 8 (1979), 234–235.

[BCS97] P. Bürgisser, M. Clausen and M. A. Shokrollahi, *Algebraic complexity theory*, Springer, 1997.

[BEGKN94] M. Blum, W. Evans, P. Gemmell, S. Kannan and M. Naor, Checking the correctness of memories, *Algorithmica* 12 (1994), 225–244.

[Ber84] S. J. Berkowitz, On computing the determinant in small parallel time using a small number of processors, *Inform. Process. Lett.* 18 (1984), 147–150.

[BGS75] T. Baker, J. Gill and R. Solovay, Relativizations of the $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$ question, *SIAM J. Comput.* 4 (1975), 431–442.

[BH99] M. R. Bridson and A. Haefliger, *Metric spaces of non-positive curvature*, Springer, 1999.

[BIP19] P. Bürgisser, C. Ikenmeyer and G. Panova, No occurrence obstructions in geometric complexity theory, *J. Amer. Math. Soc.* 32 (2019), 163–193.

[BIZ18] K. Bringmann, C. Ikenmeyer and J. Zuiddam, On algebraic branching programs of small width, *J. ACM* 65 (2018), article 32.

[BLMW11] P. Bürgisser, J. M. Landsberg, L. Manivel and J. Weyman, An overview of mathematical issues arising in the geometric complexity theory approach to VP $\neq$ VNP, *SIAM J. Comput.* 40 (2011), 1179–1209.

[BN08] I. D. Berg and I. G. Nikolaev, Quasilinearization and curvature of Aleksandrov spaces, *Geom. Dedicata* 133 (2008), 195–218.

[BS83] W. Baur and V. Strassen, The complexity of partial derivatives, *Theoret. Comput. Sci.* 22 (1983), 317–330.

[BS83b] E. Becker and N. Schwartz, Zum Darstellungssatz von Kadison–Dubois, *Arch. Math. (Basel)* 40 (1983), 421–428.

[BT72] F. Bruhat and J. Tits, Groupes réductifs sur un corps local: I. Données radicielles valuées, *Publ. Math. IHÉS* 41 (1972), 5–251.

[Bür00] P. Bürgisser, *Completeness and reduction in algebraic complexity theory*, Springer, 2000.

[Bür00b] P. Bürgisser, Cook's versus Valiant's hypothesis, *Theoret. Comput. Sci.* 235 (2000), 71–88.

[Bür04] P. Bürgisser, The complexity of factors of multivariate polynomials, *Found. Comput. Math.* 4 (2004), 369–396.

[Bür24] P. Bürgisser, Completeness classes in algebraic complexity theory, arXiv:2406.06217, 2024; to appear in *Foundations of Computation and Machine Learning: The Work of Leslie Valiant* (R. A. Servedio, ed.), ACM Books.

[CCL10] J.-Y. Cai, X. Chen and D. Li, Quadratic lower bound for permanent vs. determinant in any characteristic, *Comput. Complexity* 19 (2010), 37–56.

[CDM16] F. Capelli, A. Durand and S. Mengel, The arithmetic complexity of tensor contraction, *Theory Comput. Syst.* 58 (2016), 506–527.

[CKRST20] P. Chatterjee, M. Kumar, C. Ramya, R. Saptharishi and A. Tengse, On the existence of algebraically natural proofs, *FOCS 2020*; arXiv:2004.14147.

[CKSV22] P. Chatterjee, M. Kumar, A. She and B. L. Volk, Quadratic lower bounds for algebraic branching programs and formulas, *Comput. Complexity* 31 (2022), article 8.

[Cur26] R. Curticapean, Get permanent results faster by counting stars, arXiv:2611.03636, 2026.

[CVZ19] M. Christandl, P. Vrana and J. Zuiddam, Asymptotic tensor rank of graph tensors: beyond matrix multiplication, *Comput. Complexity* 28 (2019), 57–111.

[CVZ23] M. Christandl, P. Vrana and J. Zuiddam, Universal points in the asymptotic spectrum of tensors, *J. Amer. Math. Soc.* 36 (2023), 31–79.

[CZ19] M. Christandl and J. Zuiddam, Tensor surgery and tensor rank, *Comput. Complexity* 28 (2019), 27–56.

[DDS21] P. Dutta, P. Dwivedi and N. Saxena, Demystifying the border of depth-3 algebraic circuits, *FOCS 2021*, 92–103.

[DHM02] C. Damm, M. Holzer and P. McKenzie, The complexity of tensor calculus, *Comput. Complexity* 11 (2002), 54–89.

[DJ02] J. Dymara and T. Januszkiewicz, Cohomology of buildings and of their automorphism groups, *Invent. Math.* 150 (2002), 579–627.

[EGOW18] K. Efremenko, A. Garg, R. Oliveira and A. Wigderson, Barriers for rank methods in arithmetic complexity, *ITCS 2018*, LIPIcs 94, 1:1–1:19.

[Eis95] D. Eisenbud, *Commutative algebra with a view toward algebraic geometry*, Springer, 1995.

[EJ10] M. Ershov and A. Jaikin-Zapirain, Property (T) for noncommutative universal lattices, *Invent. Math.* 179 (2010), 303–347.

[Enf69] P. Enflo, On the nonexistence of uniform homeomorphisms between L_p-spaces, *Ark. Mat.* 8 (1969), 103–105.

[FLMV05] L. Fortnow, R. Lipton, D. van Melkebeek and A. Viglas, Time-space lower bounds for satisfiability, *J. ACM* 52 (2005), 835–865.

[For24] M. A. Forbes, Low-depth algebraic circuit lower bounds over any field, *CCC 2024*, LIPIcs 300, 31:1–31:16.

[FPV13] H. Fournier, S. Perifel and R. de Joannis de Verclos, On fixed-polynomial size circuit lower bounds for uniform polynomials in the sense of Valiant, *Inform. and Comput.* 240 (2015), 31–41; preliminary version in MFCS 2013.

[Fri21] T. Fritz, A generalization of Strassen's Positivstellensatz, *Comm. Algebra* 49 (2021), 482–499.

[FSV18] M. A. Forbes, A. Shpilka and B. L. Volk, Succinct hitting sets and barriers to proving lower bounds for algebraic circuits, *Theory Comput.* 14 (2018), article 18.

[GKKS14] A. Gupta, P. Kamath, N. Kayal and R. Saptharishi, Approaching the chasm at depth four, *J. ACM* 61 (2014), article 33.

[GKKS16] A. Gupta, P. Kamath, N. Kayal and R. Saptharishi, Arithmetic circuits: a chasm at depth 3, *SIAM J. Comput.* 45 (2016), 1064–1079.

[GKSS17] J. A. Grochow, M. Kumar, M. Saks and S. Saraf, Towards an algebraic natural proofs barrier via polynomial identity testing, arXiv:1701.01717, 2017.

[Gly10] D. G. Glynn, The permanent of a square matrix, *European J. Combin.* 31 (2010), 1887–1891.

[GMOW19] A. Garg, V. Makam, R. Oliveira and A. Wigderson, More barriers for rank methods, via a "numeric to symbolic" transfer, *FOCS 2019*, 824–844.

[GMQ16] J. A. Grochow, K. D. Mulmuley and Y. Qiao, Boundaries of VP and VNP, *ICALP 2016*, LIPIcs 55, 34:1–34:14.

[Gro03] M. Gromov, Random walk in random groups, *Geom. Funct. Anal.* 13 (2003), 73–146.

[Gro15] J. A. Grochow, Unifying known lower bounds via geometric complexity theory, *Comput. Complexity* 24 (2015), 393–475.

[Guo26] J. Guo, An $n^2 \log\log n$ lower bound for permanent circuits with valid division, arXiv:2609.29568, 2026.

[GS89] Y. Gurevich and S. Shelah, Nearly linear time, *Logic at Botik '89*, LNCS 363 (1989), 108–118.

[HPV77] J. Hopcroft, W. Paul and L. Valiant, On time versus space, *J. ACM* 24 (1977), 332–337.

[HR18] G. H. Hardy and S. Ramanujan, Asymptotic formulae in combinatory analysis, *Proc. London Math. Soc.* (2) 17 (1918), 75–115.

[Hya79] L. Hyafil, On the parallel evaluation of multivariate polynomials, *SIAM J. Comput.* 8 (1979), 120–123.

[IN05] H. Izeki and S. Nayatani, Combinatorial harmonic maps and discrete-group actions on Hadamard spaces, *Geom. Dedicata* 114 (2005), 147–188.

[IP17] C. Ikenmeyer and G. Panova, Rectangular Kronecker coefficients and plethysms in geometric complexity theory, *Adv. Math.* 319 (2017), 40–66.

[IS22] C. Ikenmeyer and A. Sanyal, A note on VNP-completeness and border complexity, *Inform. Process. Lett.* 176 (2022), 106243.

[JS82] M. Jerrum and M. Snir, Some exact complexity results for straight-line computations over semirings, *J. ACM* 29 (1982), 874–897.

[Kal85] K. A. Kalorkoti, A lower bound for the formula size of rational functions, *SIAM J. Comput.* 14 (1985), 678–687.

[Kas11] M. Kassabov, Subspace arrangements and property T, *Groups Geom. Dyn.* 5 (2011), 445–477.

[KI04] V. Kabanets and R. Impagliazzo, Derandomizing polynomial identity tests means proving circuit lower bounds, *Comput. Complexity* 13 (2004), 1–46.

[KLS10] R. Koekoek, P. A. Lesky and R. F. Swarttouw, *Hypergeometric orthogonal polynomials and their q-analogues*, Springer, 2010.

[KLSS17] N. Kayal, N. Limaye, C. Saha and S. Srinivasan, An exponential lower bound for homogeneous depth four arithmetic formulas, *SIAM J. Comput.* 46 (2017), 307–335.

[KO18] T. Kaufman and I. Oppenheim, Construction of new local spectral high dimensional expanders, *STOC 2018*, 773–786.

[KO23] T. Kaufman and I. Oppenheim, High dimensional expanders and coset geometries, *European J. Combin.* 111 (2023), 103696.

[Koi12] P. Koiran, Arithmetic circuits: the chasm at depth four gets wider, *Theoret. Comput. Sci.* 448 (2012), 56–65.

[KP09] P. Koiran and S. Perifel, A superpolynomial lower bound on the size of uniform non-constant-depth threshold circuits for the permanent, *CCC 2009*, 35–40.

[KRST22] M. Kumar, C. Ramya, R. Saptharishi and A. Tengse, If VNP is hard, then so are equations for it, *STACS 2022*, LIPIcs 219, 44:1–44:13.

[KS17] M. Kumar and S. Saraf, On the power of homogeneous depth 4 arithmetic circuits, *SIAM J. Comput.* 46 (2017), 336–387.

[KS93] N. J. Korevaar and R. M. Schoen, Sobolev spaces and harmonic maps for metric space targets, *Comm. Anal. Geom.* 1 (1993), 561–659.

[Kum20] M. Kumar, On the power of border of depth-3 arithmetic circuits, *ACM Trans. Comput. Theory* 12 (2020), article 5.

[KV26] M. Kumar and B. L. Volk, A quadratic lower bound on determinantal complexity, arXiv:2609.34462, 2026.

[Lan17] J. M. Landsberg, *Geometry and complexity theory*, Cambridge University Press, 2017.

[Lean] L. de Moura and S. Ullrich, The Lean 4 theorem prover and programming language, *CADE 28*, LNCS 12699 (2021), 625–635.

[LMR13] J. M. Landsberg, L. Manivel and N. Ressayre, Hypersurfaces with degenerate duals and the geometric complexity theory program, *Comment. Math. Helv.* 88 (2013), 469–484.

[LST25] N. Limaye, S. Srinivasan and S. Tavenas, Superpolynomial lower bounds against low-depth algebraic circuits, *J. ACM* 72 (2025), article 26; preliminary version in FOCS 2021.

[Mal03] G. Malod, *Polynômes et coefficients*, Thèse, Université Claude Bernard Lyon 1, 2003.

[Mathlib] The mathlib Community, The Lean mathematical library, *CPP 2020*, 367–381.

[MN13] M. Mendel and A. Naor, Spectral calculus and Lipschitz extension for barycentric metric spaces, *Anal. Geom. Metr. Spaces* 1 (2013), 163–199.

[MP08] G. Malod and N. Portier, Characterizing Valiant's algebraic complexity classes, *J. Complexity* 24 (2008), 16–38.

[MR04] T. Mignon and N. Ressayre, A quadratic bound for the determinant and permanent problem, *Int. Math. Res. Not.* 2004, 4241–4253.

[MS01] K. D. Mulmuley and M. Sohoni, Geometric complexity theory I: an approach to the P vs. NP and related problems, *SIAM J. Comput.* 31 (2001), 496–526.

[MS08] K. D. Mulmuley and M. Sohoni, Geometric complexity theory II: towards explicit obstructions for embeddings among class varieties, *SIAM J. Comput.* 38 (2008), 1175–1206.

[MS08b] I. L. Markov and Y. Shi, Simulating quantum computation by contracting tensor networks, *SIAM J. Comput.* 38 (2008), 963–981.

[MW17] C. D. Murray and R. R. Williams, Easiness amplification and uniform circuit lower bounds, *CCC 2017*, LIPIcs 79, 8:1–8:21.

[Nis91] N. Nisan, Lower bounds for non-commutative computation, *STOC 1991*, 410–418.

[NS23] E. Nestoridi and A. Sly, The random walk on upper triangular matrices over $\mathbb{Z}/m\mathbb{Z}$, *Probab. Theory Related Fields* 187 (2023), 571–601.

[NW97] N. Nisan and A. Wigderson, Lower bounds on arithmetic circuits via partial derivatives, *Comput. Complexity* 6 (1996/97), 217–234.

[OAI26] OpenAI, manuscript on the border determinantal complexity of the permanent, with a Lean development, September 24, 2026; see [Guo26, She26b].

[Opp17] I. Oppenheim, Averaged projections, angles between groups and strengthening of Banach property (T), *Math. Ann.* 367 (2017), 623–666.

[Orús14] R. Orús, A practical introduction to tensor networks: matrix product states and projected entangled pair states, *Ann. Physics* 349 (2014), 117–158.

[PPST83] W. J. Paul, N. Pippenger, E. Szemerédi and W. T. Trotter, On determinism versus non-determinism and related problems, *FOCS 1983*, 429–438.

[PS13] Y. Peres and A. Sly, Mixing of the upper triangular matrix walk, *Probab. Theory Related Fields* 156 (2013), 581–591.

[Raz09] R. Raz, Multi-linear formulas for permanent and determinant are of super-polynomial size, *J. ACM* 56 (2009), article 8.

[Raz10] R. Raz, Elusive functions and lower bounds for arithmetic circuits, *Theory Comput.* 6 (2010), 135–177.

[Raz13] R. Raz, Tensor-rank and lower bounds for arithmetic formulas, *J. ACM* 60 (2013), article 40.

[Raz26] R. Raz, Polynomial lower bounds for arithmetic circuits over non-commutative rings, ECCC TR26-061, arXiv:2604.22006, 2026.

[Res68] Yu. G. Reshetnyak, Inextensible mappings in a space of curvature no greater than K, *Siberian Math. J.* 9 (1968), 683–689.

[Rob91] J. M. Robson, An $O(T \log T)$ reduction from RAM computations to satisfiability, *Theoret. Comput. Sci.* 82 (1991), 141–149.

[RR97] A. A. Razborov and S. Rudich, Natural proofs, *J. Comput. System Sci.* 55 (1997), 24–35.

[Rys63] H. J. Ryser, *Combinatorial mathematics*, Carus Mathematical Monographs 14, Mathematical Association of America, 1963.

[RZ21] R. Robere and J. Zuiddam, Amortized circuit complexity, formal complexity measures, and catalytic algorithms, *FOCS 2021*, 759–769.

[Sap21] R. Saptharishi, *A survey of lower bounds in arithmetic circuit complexity*, version 9.0.3, 2021, `https://github.com/dasarpmar/lowerbounds-survey`.

[Sat09] T. Sato, An alternative proof of Berg and Nikolaev's characterization of CAT(0)-spaces via quadrilateral inequality, *Arch. Math. (Basel)* 93 (2009), 487–490.
[Sch78] C. P. Schnorr, Satisfiability is quasilinear complete in NQL, *J. ACM* 25 (1978), 136–145.
[Sch80] A. Schönhage, Storage modification machines, *SIAM J. Comput.* 9 (1980), 490–508.
[She26] K. Sheshadri, A fully Lean-verified quadratic lower bound on the border determinantal complexity of $x_1^n + \cdots + x_n^n$ over $\mathbb{C}$, including the symmetric and exact models, arXiv:2606.13628, 2026.
[She26b] K. Sheshadri, Cubic lower bounds for the permanent in odd characteristic, for Pfaffian representations, and for polynomial entries, arXiv:2610.40484, 2026.
[Sto95] R. Stong, Random walks on the groups of upper triangular matrices, *Ann. Probab.* 23 (1995), 1939–1949.
[Str73] V. Strassen, Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten, *Numer. Math.* 20 (1973), 238–251.
[Str87] V. Strassen, Relative bilinear complexity and matrix multiplication, *J. reine angew. Math.* 375/376 (1987), 406–443.
[Str88] V. Strassen, The asymptotic spectrum of tensors, *J. reine angew. Math.* 384 (1988), 102–152.
[Str91] V. Strassen, Degeneration and complexity of bilinear maps: some asymptotic spectra, *J. reine angew. Math.* 413 (1991), 127–180.
[Stu03] K.-T. Sturm, Probability measures on metric spaces of nonpositive curvature, *Contemp. Math.* 338 (2003), 357–390.
[SW14] R. Santhanam and R. Williams, On uniformity and circuit lower bounds, *Comput. Complexity* 23 (2014), 177–205.
[SY10] A. Shpilka and A. Yehudayoff, Arithmetic circuits: a survey of recent results and open questions, *Found. Trends Theor. Comput. Sci.* 5 (2010), 207–388.
[Tav15] S. Tavenas, Improved bounds for reduction to depth 4 and depth 3, *Inform. and Comput.* 240 (2015), 2–11.
[Val79] L. G. Valiant, Completeness classes in algebra, *STOC 1979*, 249–261.
[Val79b] L. G. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* 8 (1979), 189–201.
[Val82] L. G. Valiant, Reducibility by algebraic projections, in *Logic and Algorithmic*, Monogr. Enseign. Math. 30 (1982), 365–380.
[VC17] P. Vrana and M. Christandl, Entanglement distillation from Greenberger–Horne–Zeilinger shares, *Comm. Math. Phys.* 352 (2017), 621–627.
[Vin91] V. Vinay, Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits, *Structure in Complexity Theory 1991*, 270–284.
[vM07] D. van Melkebeek, A survey of lower bounds for satisfiability and related problems, *Found. Trends Theor. Comput. Sci.* 2 (2007), 197–303.
[Vra22] P. Vrana, A generalization of Strassen's theorem on preordered semirings, *Order* 39 (2022), 209–228.
[VSBR83] L. G. Valiant, S. Skyum, S. Berkowitz and C. Rackoff, Fast parallel computation of polynomials using few processors, *SIAM J. Comput.* 12 (1983), 641–644.
[Wan00] M.-T. Wang, Generalized harmonic maps and representations of discrete groups, *Comm. Anal. Geom.* 8 (2000), 545–563.
[Wil25] R. R. Williams, Simulating time with square-root space, *STOC 2025*, 13–23.
[WZ26] A. Wigderson and J. Zuiddam, Asymptotic spectra: theory, applications, and extensions, *Bull. Amer. Math. Soc.* 63 (2026), 199–331.
[Yab15] A. Yabe, Bi-polynomial rank and determinantal complexity, arXiv:1504.00151, 2015.
[Yeh19] A. Yehudayoff, Separating monotone VP and VNP, *STOC 2019*, 425–429.
[Zui19] J. Zuiddam, The asymptotic spectrum of graphs and the Shannon capacity, *Combinatorica* 39 (2019), 1173–1184.

Appendix: the full proof

The appendices give the written proof of the results of Section 1, in the order of the overview in Section 2. Appendices A to K prove the lower bound for the permanent over the real numbers, and Appendix L assembles it. Appendix M passes to all fields and to the Hamiltonian cycle polynomial, and Appendix N to approximate computation. Appendix O proves the standard facts on CAT(0) spaces that are used, Appendix P lists the terms coined here, and Appendix Q describes the Lean development. Appendices A to L are the sections of the proof, and the text refers to them as "Section A" to "Section L". Every numbered statement of the appendices except Remark N.6 has a Lean theorem, whose name is formed from the kind of the statement and a short name, such as `thm_alt` for Theorem D.2; Table 3 lists these names.

The printed proofs differ in completeness. Those of Sections A to G and of Appendix O, and the deduction of Theorem L.1 in Section L, are meant to be complete, if terse. The proofs of Proposition I.1, Theorem H.2(d), (e), Theorem J.3 and Proposition N.4 describe machines and programs and account for their sizes and running times. That these do exactly what is claimed can be verified only in the Lean development, and each of these four statements is followed by a formalization note that says so. Consequently the contracting case, and with it every main result, rests on Lean at these points.

Table 1 shows how the sections depend on each other. Section E treats the flat case completely, and nothing after it is needed for that case except Section L. The proof of the duality theorem in Section A can be skipped at a first reading, since the theorem is used only in Section G. The standard facts on CAT(0) spaces that Section C collects are proved in Appendix O. The main terms that are introduced in this paper or used in a special sense are listed in Appendix P.

Notation and conventions. Recall that log is the logarithm to the base two. We write ln for the natural logarithm, and for an integer $n \geq 1$ we let $\lg n = \lfloor \log n \rfloor + 1$ be the number of its binary digits. The set $\mathbb{N}$ contains 0, $[k] = \{1, \dots, k\}$, and for a condition P the bracket $[P]$ is 1 if P holds and 0 otherwise. The letters $c > 0$ and C denote a small and a large constant whose values may differ from one statement to the next, and inside a proof a constant taken from another statement carries the name of that statement as a subscript. Constants with names of their own, such as κ (the constant of Lemma B.1), μ, C_0, C_1, h, b and H, keep their values once they are chosen, and from Section G on the constants in $O(\cdot)$ and the word "large" depend on these choices only. The letters Λ, N and s are reused with local meanings, each introduced where it is used. In Sections A and E the letter c also denotes the factor by

Section	what is proved	uses	case
A	spectral points and prices; duality; random points	–	both
B	the inequality (B.2) for two crossing wires	A	both
C	prices satisfy the inequality of Bruhat and Tits	A, B	both
D	the flat case or the contracting case holds	C	both
E	lower bounds for $f_{k,\ell}$ and for the permanent	A, C, D	flat
F	a Poincaré inequality on unitriangular groups	C, D	contracting
G	rerouting laws	A, F	contracting
H	every network is a sum of tree networks	G	contracting
I	programs and tape automata; runs as networks	G, H	contracting
J	computations can be cut into pieces	H, I	contracting
K	small circuits give accelerators; there are none	E, G, I, J	contracting
L	Theorems L.1, L.2, Corollary L.3	D, E, K	both
M	all fields; the Hamiltonian cycle polynomial	D, E, K	both
N	approximate computation	E, K, M	both
O	proofs of the facts on CAT(0) spaces in Lemma C.4	C	both

Table 1. Map of the proof. Of Section G, Sections I and K use only the ring A; of Section H, Section I uses only the notion of a network.

which a pseudometric d_ϕ is multiplied in order to approximate given distances. In Sections A to L the field of coefficients is $\Bbbk = \mathbb{R}$, of which we use only that it is a field in which $2 \neq 0$ (Appendix M treats all fields), and we write $\Bbbk$ so that Appendices M and N can quote the statements. Tensors that are restrictions of each other are identified. We write $|xy|$, or $|x, y|$ when the points are given by expressions, for the distance of x and y in the price under discussion, and $\|\cdot\|$ for the Euclidean norm and for the two norms of loads in Sections G and H. Single bars also denote cardinalities, lengths and sizes. Finally, $\mathbb{E}$ is an expectation, and with a subscript, as in $\mathbb{E}_g$, it is the uniform average over a finite set.

Appendix A. Tensors, spectral points and prices

In this section we compare tensors shared by finitely many parties by restriction, as in Strassen's theory of the asymptotic spectrum [Str87, Str88]. A spectral point measures such tensors compatibly with restrictions, products and sums, it induces a pseudometric on the parties, and the prices are the limits of multiples of these pseudometrics. Lemma A.3 is our link to circuits, Theorem A.6 is used only in Section G, and Propositions A.7 and A.8 add new points to a price. Section A.4 and the proofs in Sections A.5 and A.6 rest on Zorn's lemma and may be skipped at a first reading. Of the field of coefficients $\Bbbk = \mathbb{R}$ we use only that it is a field in which $2 \neq 0$, and $2 \neq 0$ is needed only for the permanent.

A.1. **Tensors, restrictions and covers.** Let I be a finite nonempty set, whose elements we call *parties*, as in quantum information theory. A *tensor* on I is an array with entries in $\Bbbk$ that has one index for each party, which runs through a finite set, the *index set* at this party. A tensor may be zero. Following Strassen, we say that s is a *restriction* of t if $s = (\bigotimes_p A_p)\, t$ for some linear maps A_p, one for each party. We think of the parties as players who share the state t and may each process their own index. We say that s is *covered* by $t_1, \dots, t_r$, and that these form a *cover* of s, if $s = s_1 + \cdots + s_r$ with every s_j a restriction of t_j, and we measure a cover by the number r of its tensors.

In the *product* st each party holds its index of s and its index of t, and 1 denotes the tensor all of whose index sets have one element and whose entry is 1. Tensors that are restrictions of each other are identified, so that the product is commutative and associative, with unit 1. Every nonzero tensor has 1 as a restriction, and every tensor is covered by finitely many copies of 1.

For a map $\pi : I \to J$ of finite sets the *push-forward* πt of a tensor t on I is the tensor on J in which the party $\pi(p)$ holds the index of p, and a party outside the image holds an index set with one element. When J has two elements, πt is a matrix, a *flattening* of t. Push-forward commutes with products and with the composition of maps, and it sends 1 to 1, restrictions to restrictions and covers to covers. In particular, a tensor on a subset $P \subseteq I$ is also a tensor on I.

For a nonempty $P \subseteq I$ and an integer $q \geq 1$ the *unit tensor* of size q on P is $\langle q \rangle_P = \sum_{i=1}^{q} \bigotimes_{p \in P} e_i$, where $e_1, \dots, e_q$ is the standard basis of $\Bbbk^q$. For parties $u \neq v$ we call $\langle q \rangle_{uv} = \langle q \rangle_{\{u,v\}}$ the *wire* of dimension q between u and v, and we put $\langle q \rangle_{uu} = 1$. A product of wires is a *graph tensor*. In the literature on graph tensors [CVZ19, CZ19, VC17] a wire is an EPR pair of level q, and in tensor networks [Orús14] it is a bond of dimension q. Since $\langle q \rangle_P \, \langle q' \rangle_P = \langle qq' \rangle_P$, a wire of dimension 2^n is a product of n wires of dimension two, and we say that it carries n bits.

Example. On two parties u, v a tensor is a matrix, two matrices are restrictions of each other if and only if they have the same rank, and the matrix of rank $q \geq 1$ is the wire $\langle q \rangle_{uv}$. On three parties the restrictions of $\langle q \rangle_{uvw}$ are the tensors of tensor rank at most q, and $\langle q \rangle_{uv} \langle q \rangle_{vw} \langle q \rangle_{uw}$ is the tensor of the multiplication of $q \times q$ matrices.

A.2. **Spectral points, wires and hubs.** We measure tensors by functions that are compatible with the three notions just introduced.

Definition A.1. A *spectral point* on I is a function $\phi \geq 0$ on the tensors on I such that $\phi(1) = 1$, such that $\phi(st) = \phi(s)\phi(t)$ for all tensors s and t, and such that $\phi(s) \leq \sum_{j=1}^{r} \phi(t_j)$ whenever s is covered by $t_1, \ldots, t_r$.

For $r = 1$ the last condition says that ϕ is monotone under restriction, and for $r = 0$ it says that $\phi(0) = 0$. Since 1 is a restriction of every nonzero tensor, $\phi(t) \geq 1$ for every $t \neq 0$. Examples are the function that is 1 on all nonzero tensors and the rank of a flattening. This class is larger than Strassen's asymptotic spectrum [Str88], whose points are in addition additive under direct sums and normalized on unit tensors, and for example ϕ^a is again a spectral point for $0 < a \leq 1$.

The next lemma describes the values of a spectral point on unit tensors, and in particular on wires.

Lemma A.2 (exponents of unit tensors)**.** *Let ϕ be a spectral point on I. For every nonempty $P \subseteq I$ there is a real number $g_\phi(P) \geq 0$ such that $\phi(\langle q \rangle_P) = q^{g_\phi(P)}$ for all $q \geq 1$. The function defined by $d_\phi(u, v) = g_\phi(\{u, v\})$ for $u \neq v$ and by $d_\phi(u, u) = 0$ is a pseudometric on I with values in $[0, 1]$. Moreover $g_\phi(P) \leq \sum_e d_\phi(e)$, where e runs over the edges of any tree with vertex set P.*

We call $g_\phi(P)$ the *exponent* of P under ϕ.

Proof. Each claim follows from a restriction between products of unit tensors. The function $q \mapsto \phi(\langle q \rangle_P)$ is at least 1, multiplicative, because $\langle q \rangle_P \langle q' \rangle_P = \langle qq' \rangle_P$, and nondecreasing, because $\langle q \rangle_P$ is a restriction of $\langle q' \rangle_P$ for $q \leq q'$, and hence it is a power of q. A wire of dimension q is covered by q copies of 1, so $d_\phi \leq 1$. Contracting the two indices at v shows that $\langle q \rangle_{uw}$ is a restriction of $\langle q \rangle_{uv} \langle q \rangle_{vw}$, which gives the triangle inequality. Identifying the indices at every vertex of a tree in the same way shows that $\langle q \rangle_P$ is a restriction of the product of the wires on the edges of the tree, which gives the last inequality. □

The next lemma routes a tensor of rank at most r across a partition of the parties through two new parties, which we call *hubs*, as in Figure 4.

Lemma A.3 (routing a tensor through two hubs)**.** *Let $I = I_1 \sqcup I_2$, let a tensor t on I be a sum of r products of a tensor on I_1 and a tensor on I_2, and let n_p be the size of the index set of t at the party p. Let o_1, o_2 be two further parties. Then t is a restriction of*

$$\prod_{p \in I_1} \langle n_p \rangle_{po_1} \cdot \langle r \rangle_{o_1 o_2} \cdot \prod_{p \in I_2} \langle n_p \rangle_{po_2},$$

and this remains true when o_1 and o_2 are identified with given parties.

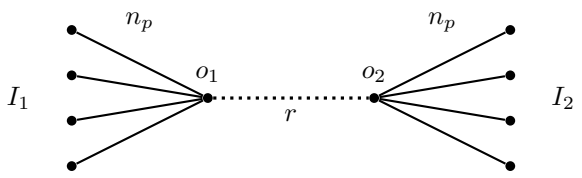


FIGURE 4. The graph tensor of Lemma A.3. The wire between a party p and its hub has dimension n_p, and the dotted wire between the hubs has dimension r.

Proof. Only the two hubs apply linear maps, and these are built from a decomposition $t = \sum_{i=1}^{r} \alpha_i \otimes \beta_i$ into tensors α_i on I_1 and β_i on I_2. We read α_i and β_i as functionals on the ends at o_1 and at o_2 of the wires from the parties. When o_1 applies $\sum_i \alpha_i \otimes e_i^*$ and o_2 applies $\sum_i \beta_i \otimes e_i^*$, where e_i^* acts on the wire between the hubs, the result is t. The last statement follows by push-forward along the map that sends o_1 and o_2 to the given parties. □

Together with Lemma A.2 this gives the inequality (2.3) for every spectral point on $I \cup \{o_1, o_2\}$.

A.3. **Prices.** By Lemma A.2 a spectral point ϕ gives the value $2^{n\,d_\phi(u,v)}$ to a wire that carries n bits between u and v, so $d_\phi(u,v)$ is what ϕ charges for one bit of such a wire, which explains the name in the next definition.

Definition A.4. A *price* on a finite set V is a limit of functions $c\,d_\phi$, where $c \geq 0$ is a real number and ϕ is a spectral point on V. A function on the pairs of elements of an arbitrary set Y is a price if all its pull-backs to finite sets are prices. We write $|xy|$ for the distance of x and y in a given price.

Here limits are pointwise, and the pull-back of a function w on the pairs of Y along a map $g : V \to Y$ is $(x,y) \mapsto w(g(x),g(y))$. Prices are pseudometrics. Pull-backs, nonnegative multiples and pointwise limits of prices are prices, where for a map $g : V \to W$ of finite sets and a spectral point ϕ on W one uses the spectral point $t \mapsto \phi(gt)$ on V. The prices that are bounded by a given function form a compact set, by Tychonoff's theorem.

A.4. **Upper functionals.** Spectral points are hard to construct directly, because they must be exactly multiplicative, whereas submultiplicative functions arise easily as infima over covers. The tensor rank is such a function, and for each of them we find a spectral point that is nowhere larger (Proposition A.5).

A function $p \geq 0$ on the tensors on I is an *upper functional* if $p(1) = 1$, $p(st) \leq p(s)p(t)$ for all s,t, and $p(s) \leq \sum_j p(t_j)$ whenever s is covered by $t_1,\dots,t_r$. One function is *below* another if it is at most as large at every tensor. For tensors x,y and a real number $c \geq 0$ we say that p *respects the triple* (x,y,c) if $p(tx) \leq c\,p(ty)$ for every tensor t.

Proposition A.5 (spectral points below upper functionals)**.** *Below every upper functional there is a spectral point that respects every triple which the upper functional respects.*

Proof. We show in two steps that a minimal upper functional is multiplicative. The upper functionals below the given one that respect the triples which it respects are closed under infima of chains, so by Zorn's lemma there is a minimal one, which we call p.

Let $r(t) = \inf_{n\geq 1} p(t^n)^{1/n}$, which by Fekete's lemma is also the limit for $n \to \infty$. Clearly $r \leq p$, $r(1) = 1$ and $r(st) \leq r(s)r(t)$. If s is covered by $t_1,\dots,t_l$, then s^n is covered by their words of length n. For $\delta > 0$ let c_δ be a constant with $p(t_j^a) \leq c_\delta (r(t_j)+\delta)^a$ for all j and all $a \geq 0$. A word w in which t_j occurs a_j times satisfies $p(w) \leq \prod_j p(t_j^{a_j}) \leq c_\delta^l \prod_j (r(t_j)+\delta)^{a_j}$, and over all words of length n these bounds add up to $c_\delta^l \big(\sum_j (r(t_j)+\delta)\big)^n$. As δ was arbitrary, this gives $r(s) \leq \sum_j r(t_j)$. If p respects (x,y,c), then $p(t^n x^n) \leq c^n p(t^n y^n)$, so r respects this triple as well. By minimality $r = p$, that is, $p(t^n) = p(t)^n$.

Let f be a tensor with $p(f) > 0$ and let $p_f(t) = \inf_{n\geq 0} p(tf^n)/p(f)^n$, where the sequence is nonincreasing, so that $p_f \leq p$. This is again such an upper functional, because $p_f(1) = 1$ by the previous step, submultiplicativity follows from $p(stf^{a+b}) \leq p(sf^a)\,p(tf^b)$, and the two other properties pass to the limit. By minimality $p_f = p$, so that $p(t) \leq p(tf)/p(f)$ and hence $p(tf) = p(t)p(f)$. If $p(f) = 0$, then $p(tf) = 0$ as well, so p is multiplicative. □

A.5. **Duality.** The next theorem is a form of Strassen's duality theorem [Str88], for the larger class of spectral points of Definition A.1.

Theorem A.6 (duality)**.** *Let $C \neq 0$ and $A_1,\dots,A_r$ be tensors on I, not all A_β zero, and suppose that every spectral point ϕ on I satisfies $\phi(C) \leq \max_\beta \phi(A_\beta)$. Then for every $\varepsilon > 0$ and all large N the power C^N is covered by at most $2^{\varepsilon N}$ tensors, each of which is a product of N factors from $A_1,\dots,A_r$.*

Informally, if no spectral point detects that C is larger than all the A_β, then C^N can be made from products of N of the A_β with only $2^{\varepsilon N}$ summands. All identities of Sections G to J arise in this way, and none of them is written down explicitly. For background see [Str87, BS83b, Zui19, Fri21, Vra22] and the survey [WZ26], and for an earlier use of duality in circuit complexity see Robere and Zuiddam [RZ21].

Proof. We apply Proposition A.5 to a function p that is defined by covers in which some factors C have been exchanged for factors A_β. For $v \in \mathbb{N}^r$ write $A^v = \prod_\beta A_\beta^{v_\beta}$ and $|v| = \sum_\beta v_\beta$, and let $\eta = 2^{-\varepsilon/2}$. A *certificate* of level $k \geq 0$ for a tensor t is a cover of tC^k by products $C^{u_j} A^{v_j}$ with $u_j + |v_j| = k$, and its *weight* $\sum_j \eta^{|v_j|}$ decreases with every exchange of a C for an A_β. Let $p(t)$ be the infimum of the weights of all certificates for t.

Certificates can be multiplied, and they can be raised to a common level, by multiplying all terms by C, and then added along covers. As the certificate of level 0 with the single term 1 gives $p(1) \leq 1$, and copies of 1 give every tensor a certificate of level 0, the function p is an upper functional if $p(1) \geq 1$. Suppose that this is so. Multiplying a certificate for tC by A_β gives a certificate for tA_β of one level higher and of η times the weight, so p respects the triples (A_β, C, η). By Proposition A.5 some spectral point ϕ satisfies $\phi(A_\beta) \leq \eta\, \phi(C)$ for all β, which contradicts the hypothesis, as $\phi(C) \geq 1$.

Hence $p(1) < 1$, and 1 has a certificate of some level k and of weight less than 1. Then $k \geq 1$, and every term has $v_j \neq 0$, since its weight alone would be 1 otherwise, so this certificate is a cover of C^k by terms in which at least one C has been exchanged. Starting from C^N, replace C^k by this cover as long as some term contains it. The total weight never increases, so it stays at most 1, and each final term has fewer than k factors C and, having N factors in all, weight at least η^N. Hence there are at most η^{-N} terms. Each remaining factor C is covered by a fixed number c of copies of a nonzero A_β, and replacing it in this way gives at most $c^k \eta^{-N} \leq 2^{\varepsilon N}$ terms for large N. □

A.6. **Extension and random points.** In Sections C and E we add new points to a price by placing each of them at one of the old points, chosen at random, and Proposition A.8 below provides an enlarged price in which distances are at most the expected distances of these positions. It follows from the next proposition for spectral points, in which I is the set of old parties, I' is the enlarged set, the maps π_i are the possible positions of the new parties and the θ_i are their probabilities.

Proposition A.7 (extension)**.** *Let $\iota : I \to I'$ and $\pi_i : I' \to I$, for $i = 1, \dots, n$, be maps of finite sets with $\pi_i \iota = \mathrm{id}$ for all i, and let $\theta_1, \dots, \theta_n \geq 0$ add up to 1. Let ϕ be a spectral point on I and let C be a tensor on I. Then there is a spectral point ϕ' on I' with*

$$\phi'(t) \leq \prod_i \phi(\pi_i t)^{\theta_i} \quad \text{for all tensors } t \text{ on } I', \qquad \phi'(\iota C) = \phi(C).$$

Proof. We turn the product on the right, which we call $f(t)$, into an upper functional by an infimum over covers. The function f is multiplicative and monotone under restriction. Let $p(t)$ be the infimum of $\sum_j f(b_j)$ over all covers (b_j) of t.

Suppose that $\iota A = \sum_j s_j$ for a tensor A on I, where each s_j is a restriction of b_j. As ιA has index sets with one element outside $\iota(I)$, so has every s_j, and therefore $s_j = \iota a_j$ with $\sum_j a_j = A$. Push-forward preserves restrictions, so $a_j = \pi_i s_j$ is a restriction of $\pi_i b_j$ for every i, and hence $\phi(A) \leq \sum_j \phi(a_j) \leq \sum_j f(b_j)$. On the other hand $p(\iota A) \leq f(\iota A) = \phi(A)$, as $\pi_i \iota = \mathrm{id}$. With $A = 1$ this shows that $p(1) = 1$, so p is an upper functional with $p \leq f$ and with $p(\iota A) = \phi(A)$ for all A.

If $C = 0$, Proposition A.5 finishes the proof. Otherwise let $c = \phi(C)$ and $q(t) = \inf_{m \geq 0} p(t\, \iota C^m)/c^m$. As $p(\iota C^m) = c^m$, this is an upper functional below p that respects the triple $(1, \iota C, 1/c)$, as in the last step of the proof of Proposition A.5. By Proposition A.5 there is a spectral point ϕ' below q that respects this triple, and it satisfies $\phi' \leq p \leq f$ and $c \leq \phi'(\iota C) \leq f(\iota C) = c$. □

We now state the consequence for prices.

Proposition A.8 (random points)**.** *Let w be a price on Y, let Q be a set disjoint from Y, and let σ be a random map $Y \cup Q \to Y$ that is the identity on Y, such that each $\sigma(o)$ has only finitely many values. Then there is a price on $Y \cup Q$ that extends w and satisfies $|oo'| \leq \mathbb{E}\, |\sigma(o)\sigma(o')|$ for all $o, o' \in Y \cup Q$.*

Proof. We reduce to finite sets by compactness and to a price of the form d_ϕ by a limit, and then apply Proposition A.7. It is enough to find, for all finite sets $Y_0 \subseteq Y$ and $Q_0 \subseteq Q$ such that σ maps Q_0 into Y_0, a price on $Y_0 \cup Q_0$ with the two properties, because these prices form compact sets, so that one price on $Y \cup Q$ then works for all of them at once. Now let Y and Q be finite, and let σ take the values π_i with probabilities θ_i. Passing to a limit, we may assume that $w = c\, d_\phi$, and then that $c = 1$ by homogeneity. Let C be the product of all wires $\langle 2\rangle_{uv}$ with $u, v \in Y$, and apply Proposition A.7 to the inclusion of Y in $Y \cup Q$ and the maps π_i. For every pair e of elements of $Y \cup Q$ the first condition, applied to $\langle 2\rangle_e$, gives $d_{\phi'}(e) \le \sum_i \theta_i\, d_\phi(\pi_i e)$, which is the expected distance. On Y this reads $d_{\phi'} \le d_\phi$, and by the second condition the sums over all pairs in Y are equal, so equality holds for each pair. □

Appendix B. An identity and an inequality for two crossing wires

This section proves an identity and an inequality for four parties a, b, c, d and two wires that *cross* (Figure 2), one between a and c and one between b and d. The crossing identity (Proposition B.2) writes their product as a sum of tensors Ψ_N, each of which is a sum of few products with respect to both partitions $\{a, d\} \mid \{b, c\}$ and $\{a, b\} \mid \{c, d\}$. By Lemma A.3 a spectral point turns this into the crossing inequality between distances in a price (Proposition B.3), with its consequence (B.2). Only (B.2) is used later, and only in the proof of Lemma C.2, so that the proof of the identity in Section B.3 may be skipped at first reading.

B.1. **Frobenius symbols.** A *symbol* is a pair (I, J) of finite subsets of $\mathbb{N}$ with $|I| = |J|$. The number $|I|$ is the *size* of the symbol, and $\sum_{i\in I}(2i+1) + \sum_{j\in J}(2j+1)$ is its *energy*. In this section the letters I and J always denote such subsets of $\mathbb{N}$, and not sets of parties. The symbols of energy $2n$ are the Frobenius symbols of the partitions of n [And84], which explains the name, but we use no property of partitions. For an integer $E \ge 0$ let $P(E)$ be the number of symbols of energy at most E. The following lemma says that $\ln P(E)$ grows like $\kappa\sqrt{E}$, with an upper bound that holds for every E.

Lemma B.1 (number of symbols)**.** *There is $\kappa > 0$ with $\ln P(E) \le \kappa\sqrt{E}$ for all integers $E \ge 0$ and $\ln P(E) \ge (\kappa - o(1))\sqrt{E}$ for $E \to \infty$.*

Proof. We first show that $P(k^2E) \ge P(E)^k$, by interleaving k symbols (I_r, J_r), $0 \le r < k$, of energy at most E into one symbol (I, J): we put $ki + r$ into I for $i \in I_r$, and $kj + k - 1 - r$ into J for $j \in J_r$. This map is injective, and it multiplies the total energy by k because $|I_r| = |J_r|$. Next let $0 < t < 1$. The number of finite sets $S \subseteq \mathbb{N}$ with $\sum_{i\in S}(2i+1) \le E$ is at most $t^{-E}\prod_i(1 + t^{2i+1}) \le t^{-E}e^{1/(1-t)}$. A symbol is a pair of such sets, and so the choice $1 - t = 1/(2\sqrt{E})$ gives $\ln P(E) \le 10\sqrt{E}$ for $E \ge 1$. Now let $\kappa = \sup_{E\ge1} \ln P(E)/\sqrt{E}$, which is finite by this bound and positive because $P(2) \ge 2$. For the lower bound let E_0 nearly attain the supremum and let $k = \lfloor\sqrt{E/E_0}\rfloor$. Then $k^2E_0 \le E$, and therefore $\ln P(E) \ge k \ln P(E_0) \ge (\sqrt{E/E_0} - 1)\ln P(E_0)$, which gives the lower bound as $E \to \infty$. □

Note that the proof uses no asymptotic formula for the number of partitions [HR18].

B.2. **The crossing identity.** We fix sizes $n_1, n_2 \ge 0$ and energies $e_1, e_2 \ge 0$, and let Σ_1 and Σ_2 be the sets of all symbols with size and energy (n_1, e_1), respectively (n_2, e_2). We assume that both sets are nonempty. We consider tensors on four parties a, b, c, d whose index set is Σ_1 at a and at c, and Σ_2 at b and at d, and we write s_a, s_b, s_c, s_d for their indices. Such a tensor is called *integral* if all its entries are integers. With this notation the crossing identity reads as follows.

Proposition B.2 (crossing identity)**.** *Let $\Theta = 2(e_1 + e_2)$. There are integral tensors $\Psi_0, \dots, \Psi_\Theta$ such that, all functions below being integer valued,*

(a) *$\sum_N \Psi_N$ is $\langle|\Sigma_1|\rangle_{ac}\langle|\Sigma_2|\rangle_{bd}$;*
(b) *Ψ_N is a sum of at most $(N+1)P(N)$ products of a function of (s_a, s_d) and one of (s_b, s_c);*

(c) Ψ_N *is a sum of at most* $(N'+1)P(N')$ *products of a function of* (s_a, s_b) *and one of* (s_c, s_d)*, where* $N' = \Theta - N$.

B.3. **Proof of the crossing identity.** The tensors Ψ_N will be, up to a sign, the coefficients of the determinant Ψ of a matrix with entries in $\mathbb{Z}[X]$, and we put $y = X^2$. The polynomials below are orthogonal for the geometric weight y^r, and the determinant interpolates between a matched to c and b to d, at $X = 1$, and a product form near $X = 0$ and near $X = \infty$.

The polynomials B *and* C. For $r, n \in \mathbb{N}$ let $\mathsf{B}(r,n)$ and $\mathsf{C}(r,n)$ be the coefficients of t^n in the power series $(1+t)^r/(1+yt)^{r+1}$ and $(1+yt)^r/(1+t)^{r+1}$. These are Meixner polynomials, namely $\mathsf{B}(r,n) = (-y)^n M_n(r;1,y)$ in the notation of [KLS10, Section 9.10], and the congruence (B.1) below is a truncation of their orthogonality relation. They are polynomials in y with integer coefficients, and $\mathsf{C}(r,n) = y^n\,\mathsf{B}(r,n)\big|_{y\mapsto 1/y}$, that is, $\mathsf{C}(r,n)$ is y^n times the polynomial obtained from $\mathsf{B}(r,n)$ by substituting $1/y$ for y; this follows by replacing y by $1/y$ and then t by yt in the first power series. The only other fact that we need about them is that, for all $R \geq 0$ and all $i, j \in \mathbb{N}$,

$$(1-y)\sum_{r<R} y^r \mathsf{B}(r,i)\mathsf{B}(r,j) \equiv [i=j]\, y^i \pmod{y^R}. \tag{B.1}$$

To prove it let $\beta_r(t) = (1+t)^r/(1+yt)^{r+1}$. Since $(1+yt)\beta_{r+1}(t) = (1+t)\beta_r(t)$, the sum $(1-yts)(1-y)\sum_{r<R} y^r\beta_r(t)\beta_r(s)$ telescopes to $1 - y^R(1+yt)(1+ys)\beta_R(t)\beta_R(s)$, and (B.1) follows by dividing by $1-yts$ and comparing the coefficients of $t^i s^j$.

The matrix. For symbols $s_a = (I_a, J_a)$, and likewise for b, c, d, we define a square matrix $\mathbf{M}$ with entries in $\mathbb{Z}[X]$. Its rows are indexed by the elements of I_a, I_d, I_c, I_b, and its columns by those of J_a, J_d, J_c, J_b, in this order of the parties and in increasing order for each party. We call $\{a, d\}$ the *first pair* and $\{c, b\}$ the *second pair* of parties. The entries are as follows, for an element i of the row and an element j of the column.

	a	d	c	b
a	0	$\mathsf{C}(i,j)$	$[i=j]X^{2i+1}$	$(y-1)X^{2i+1}\mathsf{B}(i,j)$
d	$-\mathsf{C}(j,i)$	0	$X^{2j+1}\mathsf{B}(j,i)$	$-[i=j]X^{2i+1}$
c	$-[i=j]X^{2i+1}$	$-X^{2i+1}\mathsf{B}(i,j)$	0	$-(y-1)\mathsf{C}(i,j)$
b	$-(y-1)X^{2j+1}\mathsf{B}(j,i)$	$[i=j]X^{2i+1}$	$(y-1)\mathsf{C}(j,i)$	0

Let $\Psi = \Psi(X; s_a, s_b, s_c, s_d) = \det \mathbf{M}$, which is a polynomial in X with integer coefficients for every choice of the four symbols.

The value at $X = 1$. At $X = 1$ every entry with a factor $y - 1$ vanishes. Hence the diagonal block of the second pair vanishes, and each of the two blocks between the pairs is block triangular with respect to the parties, with diagonal blocks $\pm([i=j])$. So, for symbols in the given classes, $\Psi|_{X=1}$ is a sign ϵ, which depends on n_1 and n_2 only, times $[J_c = I_a,\ I_c = J_a,\ J_b = I_d,\ I_b = J_d]$. Renaming the symbols at c and d by exchanging I and J identifies the index sets and does not affect (b) and (c). It gives (a), with Ψ_N defined as ϵ times the coefficient of X^N in Ψ, once we know that $\deg \Psi \leq \Theta$.

The coefficients. Fix N, let $R = N+1$, and let D be the diagonal matrix with the entries X^{2r+1}, $0 \leq r < R$. Let F have the rows $([r=i])_{r<R}$ for $i \in I_a$ and $(\mathsf{B}(r,i))_{r<R}$ for $i \in I_d$, and let G have the columns $([r=j])_{r<R}$ for $j \in J_c$ and $((y-1)\mathsf{B}(r,j))_{r<R}$ for $j \in J_b$. Define G', with rows indexed by I_c and I_b, and F', with columns indexed by J_a and J_d, by the same formulas. Three of the four blocks of FDG agree with those of $\mathbf{M}$ up to multiples of X^{2R+1}, and the fourth, at (d,b), is congruent to $-[i=j]X^{2i+1}$ by (B.1). So the block of $\mathbf{M}$ with the rows of a, d and the columns of c, b is congruent to FDG modulo X^{2R}, and in the same way the block with the rows of c, b and the columns of a, d is congruent to $-G'DF'$. Here F and F' depend on (s_a, s_d) only, and G and G' on (s_b, s_c) only. The same holds for the diagonal blocks $\mathbf{M}_1$ and $\mathbf{M}_2$ of the first and of the second pair. By the formula for a Schur complement, and up to a sign that depends on R only, the determinant of the matrix with

the blocks $\mathbf{M}_1$, FDG, $-G'DF'$, $\mathbf{M}_2$ equals that of

$$\begin{pmatrix} \mathbf{M}_1 & 0 & F & 0 \\ 0 & \mathbf{M}_2 & 0 & G' \\ 0 & DG & -1 & 0 \\ DF' & 0 & 0 & 1 \end{pmatrix},$$

where 1 is the identity matrix of size R. We use this only for the coefficients of degree at most $N < 2R$, which are those of Ψ. Each of the last $2R$ rows is the sum of its part in DG, respectively in DF', and the row that contains only its entry ± 1. We expand by multilinearity, and let T_1 and T_2 be the sets of rows of the third and of the fourth block row for which the first part is taken. The entries ± 1 remove the other rows together with their columns. Up to a sign that depends only on $T = (T_1, T_2)$, on R and on the sizes, what remains is $X^{E(T)}$, with $E(T) = \sum_{r\in T_1}(2r+1) + \sum_{r\in T_2}(2r+1)$, times the determinant of the block diagonal matrix with the blocks

$$\begin{pmatrix} \mathbf{M}_1 & F_{\cdot,T_1} \\ F'_{T_2,\cdot} & 0 \end{pmatrix}, \qquad \begin{pmatrix} \mathbf{M}_2 & G'_{\cdot,T_2} \\ G_{T_1,\cdot} & 0 \end{pmatrix}.$$

This vanishes if $|T_1| \neq |T_2|$, because the blocks are then not square, and otherwise it is a polynomial in X that depends on (s_a, s_d) times one that depends on (s_b, s_c). The T with $|T_1| = |T_2|$ and $E(T) \leq N$ are at most $P(N)$ symbols, and the degree $N - E(T)$ can be split between the two factors in at most $N + 1$ ways. This proves (b).

Reflection. We substitute $1/X$ for X in $\mathbf{M}$ and multiply the rows and the columns of the parties a and c by X^{2i+1}, those of d by $(y-1)y^i$ and those of b by y^{i+1}. The relation between B and C shows that one obtains the matrix $\mathbf{M}$ in which the parties b and d are exchanged, so that the new party d holds s_b, with the rows and columns of the new d multiplied by $1 - y$. Rows and columns are permuted alike, so that there is no sign. The multipliers at a, c, d, b contain X to the powers e_1, e_1, $e_2 - 2n_2$, $e_2 + 2n_2$, in all Θ, and the factors $(y-1)^{2n_2}$ and $(1-y)^{2n_2}$ on the two sides cancel in $\mathbb{Z}[X]$. So

$$X^{\Theta}\Psi(1/X; s_a, s_b, s_c, s_d) = \Psi(X; s_a, s_d, s_c, s_b).$$

Hence $\deg \Psi \leq \Theta$, and the coefficient of X^N at (s_a, s_b, s_c, s_d) is that of $X^{\Theta - N}$ at (s_a, s_d, s_c, s_b), so that (c) follows from (b). □

B.4. **The crossing inequality.** We now apply spectral points to the crossing identity. In the resulting inequality, the points x, v, y, u play the roles of the parties a, c, b, d, the points o, o' and w, w' are the hubs in the two products of wires on the right of Figure 2, and r, r' measure the dimensions of the two dotted wires on a logarithmic scale.

Proposition B.3 (crossing inequality)**.** *Let x, v, y, u, o, o', w, w' be points of a price and $p, q \geq 0$. There are $r, r' \geq 0$ with $r^2 + r'^2 \leq 2(p^2 + q^2)$ such that $p|xv| + q|yu|$ is at most each of*

$$p|xo| + q|yo| + p|vo'| + q|uo'| + r|oo'|,$$
$$p|xw| + q|uw| + q|yw'| + p|vw'| + r'|ww'|.$$

In particular, for every $t_0 > 0$,

$$\begin{aligned} 2(p|xv| + q|yu|) \leq\ & p(|xo| + |vo'| + |xw| + |vw'|) + q(|yo| + |uo'| + |uw| + |yw'|) \\ & + t_0(p^2 + q^2) + (|oo'|^2 + |ww'|^2)/(2t_0). \end{aligned} \tag{B.2}$$

Proof. We reduce the statement to the metric d_ϕ of a spectral point ϕ, and apply ϕ to the crossing identity for two large classes of symbols. We pull the price back to eight distinct points, which covers coincidences. Both bounds are homogeneous in the distances and the pairs (r, r') lie in a compact set, so that the bounds pass to limits, and it suffices to treat d_ϕ for a spectral point ϕ on the eight points. For a large integer n_0 let $E_1 = \lfloor p^2 n_0 \rfloor$ and $E_2 = \lfloor q^2 n_0 \rfloor$. By the pigeonhole principle there is, for $i = 1, 2$, a class Σ_i as in Section B.2, of

some size n_i and some energy $e_i \le E_i$, that contains at least the fraction $(E_i+1)^{-2}$, and of course at most all, of the symbols of energy at most E_i. So, in units of $\kappa\sqrt{n_0}$, the logarithms of $|\Sigma_1|$ and $|\Sigma_2|$ are $p+o(1)$ and $q+o(1)$ by Lemma B.1. We identify a, c, b, d with x, v, y, u, and write $|xv|$ for $d_\phi(x,v)$. The product of the two wires is covered by the $\Theta+1$ tensors Ψ_N of Proposition B.2(a), the spectral point ϕ is subadditive on covers, and $\ln(\Theta+1) = o(\sqrt{n_0})$. Hence, by Lemma A.2, some N has $\ln\phi(\Psi_N) \ge (p|xv| + q|yu| - o(1))\kappa\sqrt{n_0}$. Define $r, r' \ge 0$ by $r^2 = (\Theta - N)/n_0$ and $r'^2 = N/n_0$, so that $r^2 + r'^2 = \Theta/n_0 \le 2(p^2+q^2)$. In the same units the logarithm of the number of products in (c) is at most $r + o(1)$, and that in (b) is at most $r' + o(1)$, by the upper bound of Lemma B.1. Lemma A.3, with the hubs o, o' for (c) and w, w' for (b), bounds $\phi(\Psi_N)$ in both ways. A limit of (r, r') along a subsequence of the n_0 proves the first statement. Finally we add the two bounds and use $r\lambda \le \frac{t_0}{2}r^2 + \lambda^2/(2t_0)$, and the same with r', to obtain (B.2). □

Appendix C. Prices have nonpositive curvature

In this section we show that every price satisfies the inequality of Bruhat and Tits that characterizes nonpositive curvature (Theorem C.1), and that it is the pull-back of the metric of a complete CAT(0) space whose metric is again a price (Proposition C.3). The only properties of prices that we use are Proposition A.8, the inequality (B.2), and the fact that pull-backs, nonnegative multiples and limits of prices are prices.

C.1. **The curvature bound and the plan of its proof.** The main result of this section is the following.

Theorem C.1 (curvature bound for prices)**.** *Let x, v, m, z be points of a price with $|xm|, |vm| \le D$ and $|xv| = 2D$. Then*

$$|zm|^2 \le \tfrac{1}{2}|zx|^2 + \tfrac{1}{2}|zv|^2 - D^2. \tag{C.1}$$

The hypotheses say that m is a midpoint of x and v (in this section m denotes a point, not the size of a matrix). In the Euclidean plane (C.1) is the formula for the length of a median, and in general it is the CN inequality of Bruhat and Tits [BT72], [BH99, Exercise II.1.9(1)(c)]. Throughout the proof we write $\Gamma = |mz|^2 - \frac{1}{2}|xz|^2 - \frac{1}{2}|vz|^2$, so that the claim is $\Gamma \le -D^2$.

The plan of the proof is as follows. If a price on $\mathbb{Z}^2$ is given by a seminorm N, then (B.2) forces the parallelogram law for N, so that three values of N determine all others. One would like to differentiate the price at m, but the proof avoids differentiation. Instead, for a large integer n, we use Proposition A.8 to join x and v to m by paths of new points γ_i, and every γ_i to z by a path of new points $G(i,j)$ (Figure 5). This family of points is called a *fan*, and it has two *sheets*, one for x and one for v, because only the two paths to m together have a known length, namely $2D$. We multiply distances by n^2 and average their squares over boxes in the coordinates (i,k), where $j = i + k + n$. The boxes are short in k, so that j/n^2 follows i/n^2. The inequality (B.2) survives this averaging, and the averages depend only on the displacement and define a seminorm N that satisfies the parallelogram law (Lemma C.2). We bound N in three directions, and in a fourth direction it contains Γ by a telescoping sum, so that the parallelogram law compares them.

C.2. **Averaged squared distances.** We now describe the averaging in general. We consider two disjoint copies of $\mathbb{Z}^2$, the two sheets, so that $P+g$ lies on the sheet of P for $g \in \mathbb{Z}^2$, and we write $\|g\|_1 = |g_1| + |g_2|$. Suppose that for every $n = 1, 2, \dots$ we are given a price W_n on the union of the two sheets, and that there is a constant K with $W_n(P, P+g) \le K\|g\|_1$ for all n, all points P and all $g \in \mathbb{Z}^2$. The nth *box* consists of the points (i,k) with $0 \le i < n^2$ and $0 \le k < n$, on both sheets. For a sequence $f = (f_n)$ of uniformly bounded functions on the two sheets, the *average* $\mathbb{E}f$ is the limit, along a fixed free ultrafilter on the indices n, of the mean of f_n over the nth box. The average is linear, positive and invariant under

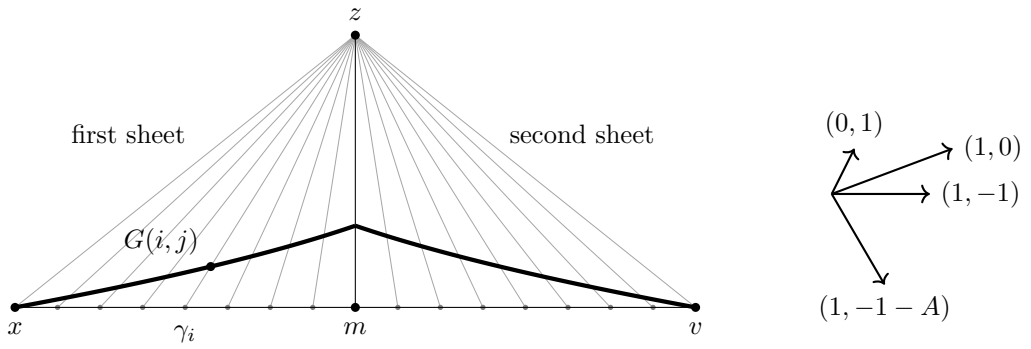


FIGURE 5. The fan, drawn as if the price were Euclidean. The averages are taken over the points near the two thick curves, where j/n^2 is close to i/n^2. On the right are the four directions in which N is estimated, in the coordinates (i, k).

translations, because the boundary of a box is negligible. We write W for (W_n), and for $g \in \mathbb{Z}^2$ we let $\nu(g) = (\mathbb{E}\, W(P, P+g)^2)^{1/2}$ and, over the integers $S \geq 1$,

$$N(g) = \lim_{S\to\infty} \nu(Sg)/S = \inf_S \nu(Sg)/S.$$

By the invariance under translations and Minkowski's inequality ν is even and subadditive, so that the limit exists and equals the infimum. Hence N is even, subadditive and homogeneous, that is, a seminorm on $\mathbb{Z}^2$, which we call *the seminorm* N. The next lemma says that this seminorm is Euclidean.

Lemma C.2 (parallelogram law for the seminorm N)**.** *Let W_n, for $n = 1, 2, \dots$, be prices on the union of the two sheets, let K be a constant with $W_n(P, P+g) \leq K\|g\|_1$ for all n, all points P and all $g \in \mathbb{Z}^2$, and let N be the seminorm defined above. Then $N(a+b)^2 + N(a-b)^2 = 2N(a)^2 + 2N(b)^2$ for all $a, b \in \mathbb{Z}^2$.*

Proof. We show that (B.2) holds for N in place of a price, and apply it to configurations of eight vectors, which are written in bold letters.

Inequality (B.2) *for ν and for N.* We apply (B.2) to W_n at the eight points $P + \mathbf{g}$, for eight fixed vectors $\mathbf{g} = \mathbf{x}, \mathbf{v}, \dots, \mathbf{w}'$, with the weights $p = |xv|$ and $q = |yu|$ measured in W_n, and take the average over P. By the Cauchy–Schwarz inequality, ν satisfies (B.2) with $\nu(\mathbf{g} - \mathbf{g}')$ in place of each distance, and with $p = \nu(\mathbf{x} - \mathbf{v})$ and $q = \nu(\mathbf{y} - \mathbf{u})$. Applying this to the eight vectors multiplied by S, dividing by S^2 and letting $S \to \infty$, we see that so does N, with the same t_0.

A symmetric configuration. We take $(\mathbf{x}, \mathbf{v}, \mathbf{y}, \mathbf{u}) = (\mathbf{p}, -\mathbf{p}, \mathbf{q}, -\mathbf{q})$ and $(\mathbf{o}, \mathbf{o}', \mathbf{w}, \mathbf{w}') = (\mathbf{a}, -\mathbf{a}, \mathbf{b}, -\mathbf{b})$ and divide by 4:

$$N(\mathbf{p})\,\partial\mathbf{p} + N(\mathbf{q})\,\partial'\mathbf{q} \leq t_0\big(N(\mathbf{p})^2 + N(\mathbf{q})^2\big) + \big(N(\mathbf{a})^2 + N(\mathbf{b})^2\big)/(2t_0), \tag{C.2}$$

where $\partial\mathbf{p} = 2N(\mathbf{p}) - N(\mathbf{p}-\mathbf{a}) - N(\mathbf{p}-\mathbf{b})$ and $\partial'\mathbf{q} = 2N(\mathbf{q}) - N(\mathbf{q}-\mathbf{a}) - N(\mathbf{q}+\mathbf{b})$. By the triangle inequality $|\partial\mathbf{p}|$ and $|\partial'\mathbf{q}|$ are at most $N(\mathbf{a}) + N(\mathbf{b})$.

Long vectors. Let $\mathbf{c} = \mathbf{a} + \mathbf{b}$, $\mathbf{e} = \mathbf{a} - \mathbf{b}$, $f = N(\mathbf{c})$ and $f' = N(\mathbf{e})$, and let $M = R^2$ for an integer $R \geq 1$. The constants in $O(\cdot)$ below do not depend on R. For $0 \leq i < R$ let $\mathbf{A}_i = M\mathbf{c} + i\mathbf{e}$ and $\mathbf{B}_i = M\mathbf{e} + i\mathbf{c}$. We apply (C.2) to $(\mathbf{p}, \mathbf{q}) = (\mathbf{A}_i, \mathbf{B}_i)$ and to $(\mathbf{A}_i - \mathbf{a}, \mathbf{B}_i - \mathbf{a})$, with $t_0 = 1/(2M)$. With $\delta_i = N(\mathbf{A}_i - \mathbf{a}) - N(\mathbf{A}_i - 2\mathbf{a})$ we have

$$\partial\mathbf{A}_i + \partial(\mathbf{A}_i - \mathbf{a}) = 2\big(N(\mathbf{A}_i) - N(\mathbf{A}_i - \mathbf{c})\big) + \delta_i - \delta_{i+1}.$$

As $t \mapsto N(t\mathbf{c} + i\mathbf{e})$ is convex on the integers $t \geq 0$, the first difference on the right is at least $f - 2N(i\mathbf{e})/M \geq f - 2f'/R$. The δ_i are bounded and telescope in the average over i. Since ∂ is bounded by $N(\mathbf{a}) + N(\mathbf{b})$ and $N(\mathbf{p}) = Mf + O(R)$, we have $N(\mathbf{p})\,\partial\mathbf{p} = Mf\,\partial\mathbf{p} + O(R)$ in both applications. Replacing $\mathbf{b}$ by $-\mathbf{b}$ exchanges $\mathbf{c}$ with $\mathbf{e}$, $\mathbf{A}_i$ with $\mathbf{B}_i$ and ∂ with ∂', so

that the same holds for ∂' at $\mathbf{B}_i$, with f' for f. We add the two inequalities, average over i and divide by $2M$, which gives

$$f^2 + f'^2 \le \tfrac{1}{2}(f^2 + f'^2) + N(\mathbf{a})^2 + N(\mathbf{b})^2 + O(1/R).$$

Thus $N(\mathbf{a}+\mathbf{b})^2 + N(\mathbf{a}-\mathbf{b})^2 \le 2N(\mathbf{a})^2 + 2N(\mathbf{b})^2$ for all $\mathbf{a}, \mathbf{b}$, and applied to $\mathbf{a}+\mathbf{b}$ and $\mathbf{a}-\mathbf{b}$ in place of $\mathbf{a}$ and $\mathbf{b}$ this inequality gives the reverse one. □

C.3. **Proof of Theorem C.1.** We carry out the plan of Section C.1. Let $\theta = 1/T$ with an integer $T \ge 2$.

The fan. For a given n we add, by Proposition A.8, points γ_i, $0 \le i \le n^2$, on each sheet, placed at m with probability i/n^2 and otherwise at x on the first sheet and at v on the second, with one random number for all of them. Then, by the same proposition, we add points $G(i,j)$, $0 \le j \le Tn^2$, placed at z with probability $j/(Tn^2)$ and otherwise at γ_i, again with one random number for all. Let $L_i = |z\gamma_i|$ and $\rho = \rho(j) = 1 - j\theta/n^2$. Then

(C.3)
$$|\gamma_i\gamma_{i+1}| \le \frac{D}{n^2}, \quad |zG(i,j)| = \rho L_i, \quad |G(i,j)G(i,j+1)| \le \frac{\theta L_i}{n^2}, \quad |G(i,j)G(i+1,j)| \le \frac{\rho D}{n^2}.$$

The second relation is an equality, because $|G(i,j)\gamma_i| \le (1-\rho)L_i$. The first bound gives $|L_{i+1} - L_i| \le D/n^2$ and $L_i \le D + |zm|$.

The prices W_n. For $(i,k) \in \mathbb{Z}^2$ on either sheet let $P(i,k) = G(i',j')$ on that sheet, where i' is i clamped to $[0,n^2]$ and j' is $i+k+n$ clamped to $[0,Tn^2]$, and let W_n be the pull-back under P of n^2 times the price. In the nth box we have $n \le j' < n^2 + 2n$, so that nothing is clamped there and $1-\rho \le 2\theta$. As $L_i \le D + |zm|$, we may take $K = 2D + |zm|$ in Section C.2. Let L denote the sequence of functions $(i,k) \mapsto L_{i'}$, and let $\Lambda_2 = \mathbb{E}L^2$, the average over both sheets, so that $\Lambda_2 \le (D + |zm|)^2$.

Four estimates. We claim that for all integers $A \ge 1$, with constants in $O(\cdot)$ that depend neither on A nor on θ,

(i) $N(1,0) \ge D$;
(ii) $N(1,-1)^2 \le D^2(1 - \theta + \theta^2)$;
(iii) $N(0,1)^2 \le \theta^2\Lambda_2$;
(iv) $N(1,-1-A)^2 \ge A^2\theta^2\Lambda_2 + A\theta\Gamma - O(A\theta^2)$.

Since $N(g) = \inf_S \nu(Sg)/S$, it is enough to prove a lower bound for $\nu(Sg)/S$ for every S, and an upper bound for one S. Below, $o(1)$ refers to $n \to \infty$.

Proof of (i). Steps by $(1,0)$ lead to a common end, because γ_{n^2} is at m on both sheets and all the points G use one random number. For a point (i,k) of a sheet let $\pi(i,k)$ be the distance of $P(i,k)$ from x on the first sheet and from v on the second, so that $W_n(P,P') \ge n^2(\pi(P') - \pi(P))$. By the invariance under translations the average of $n^2(\pi(P + S(1,0)) - \pi(P))$ is S times the one for $S = 1$. For $S = 1$ and $0 \le k < n$ the sum over $0 \le i < n^2$ telescopes to $n^2(\pi(n^2,k) - \pi(0,k))$. Here $\pi(0,k) = O(1/n)$, and $P(n^2,k)$ has distance 0 from its copy on the other sheet, so that the values of $\pi(n^2,k)$ on the two sheets add up to at least $|xv| = 2D$. Hence the mean over the box is at least $D - O(1/n)$ for $S = 1$. So $\mathbb{E}\,W(P, P+S(1,0)) \ge SD$, and $\nu(S(1,0)) \ge SD$ by the Cauchy–Schwarz inequality.

Proof of (ii). We take $S = 1$. A step by $(1,-1)$ changes i only, and $j \ge i$ in the box, so that by the last bound of (C.3) its W_n-length is at most $\rho D \le (1-\theta p)D$ with $p = i/n^2 \in [0,1]$. Over the box p has mean $\frac{1}{2} - o(1)$, and $(1-\theta p)^2 \le 1 - 2\theta p + \theta^2$, which gives (ii).

Proof of (iii). A step by $(0,1)$ changes j only, so that (iii) follows from the third bound of (C.3) with $S = 1$.

Proof of (iv). Fix S and let $n \ge AS$, so that the new j is not negative. Let $L = L_i$ and $L' = L_{i+S}$, where $L_i = L_{n^2}$ for $i > n^2$. A step by $S(1,-1-A)$ leads from $G(i,j)$ to $G(i+S, j-AS)$, and by the second relation of (C.3) the distances of these two points from z give

$$W_n(P, P + S(1,-1-A)) \ge \bigl|\, n^2\rho(L' - L) + AS\theta L' \,\bigr|.$$

We square this, use $2L'(L'-L) \geq L'^2 - L^2$, and take the average. The average of L'^2 is Λ_2, and $n^2(L'^2 - L^2)$ telescopes along i to the mean $S\Gamma + o(1)$. The factor ρ of this term is removed by means of $1-\rho \leq 2\theta$ and $n^2|L'^2 - L^2| \leq 2SD(D+|zm|)$. Dividing by S^2 proves (iv).

Conclusion. By Lemma C.2, applied to $a = (1, -1-l)$ and $b = (0,1)$, the second differences of $l \mapsto N(1,-1-l)^2$ are $2N(0,1)^2$. So by (iii) the function $\psi(l) = N(1,-1-l)^2 - l^2\theta^2\Lambda_2$ is concave on the integers $l \geq -1$, and consequently $\psi(A) \leq \psi(0) + A(\psi(0) - \psi(-1))$. We insert (iv) for $\psi(A)$, (ii) for $\psi(0)$ and (i) for $\psi(-1)$, divide by A and let $A \to \infty$:

$$\theta\Gamma - O(\theta^2) \leq D^2(-\theta + \theta^2) + \theta^2\Lambda_2.$$

Since $\Lambda_2 \leq (D+|zm|)^2$, dividing by θ and letting $\theta \to 0$ gives $\Gamma \leq -D^2$, which is (C.1). □

C.4. **Price spaces.** Sections D and F use Theorem C.1 in another form: every price is the pull-back of the metric of a complete space of nonpositive curvature, in which geodesics, barycenters and nearest-point projections are available.

A *midpoint* of two points x, v of a metric space is a point at the distance $\frac{1}{2}|xv|$ from both, and it satisfies the *CN inequality* [BT72] if (C.1) holds for it with $D = \frac{1}{2}|xv|$ and with every point z.

Definition. A metric space is *CAT(0)* if it is complete and if any two of its points have a midpoint that satisfies the CN inequality. A *price space* is a CAT(0) space whose metric is a price in the sense of Definition A.4.

This is the definition used by Sturm [Stu03], and for complete geodesic spaces it is equivalent to the more common one by comparison triangles [BH99, Exercise II.1.9(1)(c)]. The midpoint in the definition is the only midpoint of the two points, as the CN inequality shows when z is another midpoint. The following proposition says that every price arises from a price space.

Proposition C.3 (embedding into a price space)**.** *Every price on a set V is the pull-back of the metric of a price space.*

Proof. We add midpoints to V again and again and then pass to the completion, using Proposition A.8 for the midpoints and Theorem C.1 for the CN inequality.

Let $V_0 = V$, and let V_{l+1} be V_l with one new point for every pair of its points. A price on V_l extends to one on V_{l+1} by Proposition A.8, applied to the random map that fixes V_l and sends each new point to either point of its pair with probability $\frac{1}{2}$, and in it each new point is a midpoint of its pair. These prices extend one another, so together they form a price on the union of the V_l, in which any two points have a midpoint and, by Theorem C.1, every midpoint satisfies the CN inequality. We identify points at distance 0 and let Y be the completion, so that the given price on V is the pull-back of the metric of Y. This metric is a price, as a limit of prices, so by Theorem C.1 every midpoint in Y satisfies the CN inequality. Finally, let $x, v \in Y$, choose points $x_n \to x$ and $v_n \to v$ in the union of the V_l, and let m_n be a midpoint of x_n and v_n. The CN inequality for m_n at $z = m_k$ shows that (m_n) is a Cauchy sequence, and its limit is a midpoint of x and v. □

C.5. **Standard facts on CAT(0) spaces.** The next lemma collects the standard facts about CAT(0) spaces that we use [BH99, Stu03, BN08], and Appendix O gives their short proofs from the CN inequality. Part (b) is Reshetnyak's comparison for quadrilaterals [Res68, KS93], whose weak form is Enflo's roundness 2 [Enf69, Sat09], part (c) is in [Stu03], part (d) is in [BH99, II.2], and part (f) is close to the Flat Quadrilateral Theorem [BH99, II.2.11]; our proof of it uses the inequality of part (b) in the form of the quadrilateral cosine of Berg and Nikolaev [BN08]. Section D uses parts (a), (b) and (f), and Section F uses parts (a), (c), (d) and (e).

We call a subset of a CAT(0) space *convex* if it contains the midpoints of its points. The L^2 product Y^n in part (e) carries the metric $\left(\sum_i |x_i y_i|^2\right)^{1/2}$. Multiplying the metric of a

CAT(0) space by a positive constant gives a CAT(0) space again, because completeness, midpoints and the CN inequality are unchanged. When the points in a distance are given by expressions we separate them by a comma, as in $|Py, q|$ below.

Lemma C.4 (standard facts on CAT(0) spaces)**.** *Let Y be a CAT(0) space.*

(a) *For $x, v \in Y$ and $t \in [0,1]$ exactly one point $\gamma(t)$ has the distances $t|xv|$ from x and $(1-t)|xv|$ from v, and $|z\gamma(t)|^2 \le (1-t)|zx|^2 + t|zv|^2 - t(1-t)|xv|^2$ for every $z \in Y$. We write $[x, v]$ for the set of these points, the geodesic segment from x to v, and we call $\gamma(t)$ the point at t on $[x, v]$.*

(b) *For $x, v, y, u \in Y$ we have $|xv|^2 + |yu|^2 \le |xu|^2 + |yv|^2 + 2|xy||uv|$. In particular the squared diagonals of the quadrilateral with the consecutive vertices x, y, v, u add up to at most its squared sides, that is, $|xv|^2 + |yu|^2 \le |xy|^2 + |yv|^2 + |vu|^2 + |ux|^2$, and if there is equality, then opposite sides are equal.*

(c) *Points $x_1, \dots, x_n$ of Y, $n \ge 1$, have a* barycenter *b, the minimizer of $y \mapsto \sum_i |yx_i|^2$, and $\sum_i |yx_i|^2 \ge n|yb|^2 + \sum_i |bx_i|^2$ for all $y \in Y$. If b' is the barycenter of $x'_1, \dots, x'_n$, then $n|bb'| \le \sum_i |x_i x'_i|$.*

(d) *Let $F \subseteq Y$ be nonempty, closed and convex. Every $y \in Y$ has a nearest point Py in F, and $|yq|^2 \ge |y, Py|^2 + |Py, q|^2$ for all $q \in F$.*

(e) *Finite L^2 products of copies of Y are CAT(0).*

(f) Flat rectangles. *Suppose that $|xy| = |uv| = \delta$ and $|xu| = |yv| = h$, that both diagonals of the quadrilateral x, y, v, u have the squared length $\delta^2 + h^2$, and that $a \in [x, u]$ and $b \in [y, v]$ have $|xa| = sh$ and $|yb| = th$. Then $|ab|^2 = \delta^2 + h^2(s-t)^2$.*

Appendix D. The two cases

In this section we formulate two hypotheses on the field $\mathbb{k}$, a flat one and a contracting one, and we show that at least one of them holds (Theorem D.2). Section E proves the lower bound for the permanent under the first hypothesis, and Sections F to K prove it under the second.

D.1. **The two hypotheses.** Let $\mathrm{Mat}_\mu(\mathbb{F}_2)$ be the additive group of the $\mu \times \mu$ matrices over the field with two elements. For a price on it and $r \le \mu$ let $D(r) \ge 0$ be the root mean square distance of two matrices whose difference has rank r, that is, let $D(r)^2$ be the mean of $|a, a+W|^2$ over all matrices a and all matrices W of rank r. We call matrices $W_1, W_2, \dots$ *transversal* if their column spaces are linearly independent and their row spaces are as well. The rank of their sum is then the sum of their ranks. Every price on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ satisfies $D(r) \le \sqrt{r}\, D(1)$, by averaging over the quadrilaterals of Figure 6 (Step 1 of the proof of Theorem D.2).

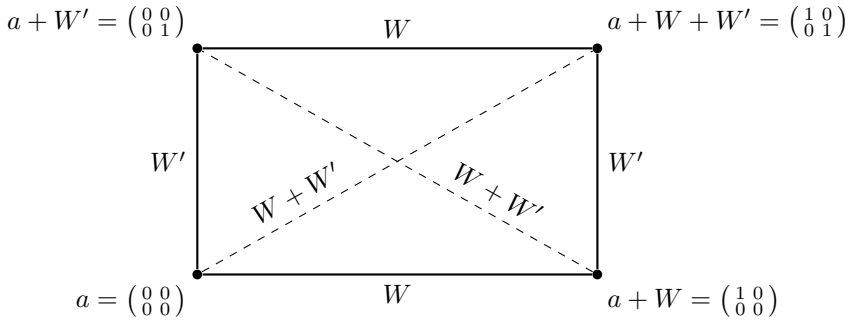


Figure 6. The quadrilateral a, $a + W$, $a + W + W'$, $a + W'$ for $\mu = 2$, $a = 0$ and the diagonal matrices W, W' of rank one. Both diagonals have the difference $W + W'$.

Definition D.1. We consider two hypotheses on the field $\Bbbk$, on which the notion of a price depends.

Flat hypothesis: All finite subsets of all Euclidean spaces, with their metrics, are prices.

Contracting hypothesis: There are an integer $\mu \geq 2$ and a real number s with $1 \leq s$ and $s^2 < \mu$ such that $D(\mu) \leq s\, D(1)$ for all prices on $\mathrm{Mat}_\mu(\mathbb{F}_2)$.

We say that we are in the *flat case* if the flat hypothesis holds, and in the *contracting case* if the contracting hypothesis holds.

Thus the contracting hypothesis says that for some μ the bound $D(\mu) \leq \sqrt{\mu}\, D(1)$ can be improved by a constant factor. The main result of this section relates the two hypotheses.

Theorem D.2 (the two cases)**.** *At least one of the two hypotheses of Definition D.1 holds, that is, one of the two cases occurs.*

D.2. **Proof of Theorem D.2.** We assume that the contracting hypothesis fails and derive the flat one. Steps 1 to 3 produce a price on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ with equality in Lemma C.4(b) for all quadrilaterals as in Figure 6, Steps 4 to 7 deduce that its squared distances are the ranks of the differences, and Step 8 fills a Euclidean cube by Lemma C.4(f).

Proof. Suppose that the contracting hypothesis fails, and fix $\mu \geq 3$. By Proposition C.3 every price is pulled back from the metric of a CAT(0) space, so that Lemma C.4 applies to the points of a price.

Step 1: subadditivity. Let (W, W') be a uniformly random transversal pair of matrices of ranks i and j, where $i + j \leq \mu$, and let a be uniformly random and independent of it. Then W, W' and $W + W'$ are uniformly distributed among the matrices of ranks i, j and $i + j$. Both diagonals of the quadrilateral a, $a + W$, $a + W + W'$, $a + W'$ have the difference $W + W'$, so the mean of the second inequality of Lemma C.4(b) over these quadrilaterals is $D(i + j)^2 \leq D(i)^2 + D(j)^2$.

Step 2: a price with equality. As the contracting hypothesis fails, for every s with $1 \leq s < \sqrt{\mu}$ there is a price on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ with $D(\mu) > s\, D(1)$. Every distance in a price on $\mathrm{Mat}_\mu(\mathbb{F}_2)$ is at most a constant, which depends on μ only, times $D(1)$, so we may scale these prices to $D(1) = 1$ and, by compactness, pass to a limit for $s \to \sqrt{\mu}$, which is a price with $D(1) = 1$ and $D(\mu)^2 \geq \mu$. By Step 1 it has $D(r)^2 \leq r$ and $\mu \leq D(\mu)^2 \leq D(r)^2 + D(\mu - r)^2 \leq \mu$, and therefore $D(r)^2 = r$ for all $r \leq \mu$.

Step 3: equality for all quadrilaterals. In this price the inequality of Step 1, a mean of finitely many inequalities, is an equality, so for every a and every transversal pair (W, W') the quadrilateral of Step 1 has equality in the second statement of Lemma C.4(b). Put $n_a(W) = |a, a + W|^2$. For transversal W, W' opposite sides are equal by the lemma, and the equality then expresses one diagonal by the other and the sides, that is,

$$n_{a+W'}(W) = n_a(W), \tag{D.1}$$

$$|a + W, a + W'|^2 = 2n_a(W) + 2n_a(W') - n_a(W + W'). \tag{D.2}$$

Step 4: additivity. Let $B_a(W, W') = \frac{1}{2}\big(n_a(W + W') - n_a(W) - n_a(W')\big)$, which is symmetric in W and W'. We claim that for transversal W_1, W_2, W_3

$$B_a(W_1, W_2 + W_3) = B_a(W_1, W_2) + B_a(W_1, W_3). \tag{D.3}$$

To see this we apply (D.2) at the matrix $a+W_2$ to the pair (W_1, W_2+W_3), whose quadrilateral has the other vertices $a + W_1 + W_2$, $a + W_1 + W_3$ and $a + W_3$, and we express all its terms by n_a, using (D.1) for $n_{a+W_2}(W_1)$ and (D.2) at a for the others. This gives

$$n_a(W_1{+}W_2{+}W_3) - \sum_{i<j} n_a(W_i{+}W_j) + \sum_i n_a(W_i) = 0,$$

which is (D.3), written out.

Step 5: B_a is constant on pairs of rank one. For column vectors over $\mathbb{F}_2$ we have $u_1x_1^\mathsf{T} + u_2x_2^\mathsf{T} = u_1(x_1+x_2)^\mathsf{T} + (u_1+u_2)x_2^\mathsf{T}$. Let W, $u_1x_1^\mathsf{T}$, $u_2x_2^\mathsf{T}$ be transversal, so that $\operatorname{rk} W + 2 \le \mu$. Then W and the two matrices on the right are transversal as well, and (D.3) gives, for $\chi(u,x) = B_a(W, ux^\mathsf{T})$,

$$\chi(u_1,x_1) - \chi(u_1,x_1+x_2) = \chi(u_1+u_2,x_2) - \chi(u_2,x_2).$$

The right-hand side does not depend on x_1, and the left-hand side changes sign when x_1 becomes x_1+x_2, which leaves the three matrices transversal, so both sides vanish. Thus x may be changed by any vector outside its span with the row space of W, and likewise u. As $\mu \ge 3$, for W of rank one such moves connect all vectors outside its row space, respectively column space, and any two matrices of rank one are both transversal to a third. As B_a is symmetric, it takes one value $\lambda(a)$ on all transversal pairs of matrices of rank one.

Step 6: $\lambda = 0$. For transversal V, W' we have $n_{a+V}(V) = n_a(V)$, we have $n_{a+V}(W') = n_a(W')$ by (D.1), and $n_{a+V}(V+W') = |a+V, a+W'|^2$ is given by (D.2), so that $B_{a+V}(V,W') = -B_a(V,W')$. If V and W' have rank one, this says that $\lambda(a+V) = -\lambda(a)$. If V has rank two and W' has rank one, the two sides are $2\lambda(a+V)$ and $-2\lambda(a)$ by (D.3), and again $\lambda(a+V) = -\lambda(a)$. But a matrix V of rank two is reached by two steps of rank one, which give $\lambda(a+V) = \lambda(a)$. Hence $\lambda = 0$.

Step 7: the squared distance is the rank. By Step 6 and (D.3), B_a vanishes on all transversal pairs, that is, n_a is additive on transversal sums. The exchange of Step 5, with $W = 0$ and n_a in place of B_a, shows that n_a has one value on all matrices of rank one. By (D.1) this value does not depend on a, so it equals $D(1)^2 = 1$. Hence $|a,a'|^2 = \operatorname{rk}(a-a')$ for all $a, a' \in \operatorname{Mat}_\mu(\mathbb{F}_2)$.

Step 8: a solid cube. By Step 7 the diagonal matrices with entries 0 and 1 form the vertices of a Euclidean cube $[0,1]^\mu$ in a price space. We fill it one coordinate at a time, sending each segment in the new direction, whose ends are already mapped, to the geodesic segment between their images. Lemma C.4(f), applied with $h = 1$ to two such segments, shows that the map stays isometric, and thus the metric of $[0,1]^\mu$ is a price. Every finite subset of a Euclidean space of dimension at most μ is, up to translation and scale, a subset of this cube, and hence its metric is a price. As $\mu \ge 3$ was arbitrary, the flat hypothesis holds. □

D.3. **Remarks on the two cases.** We do not show that the two cases exclude each other. Outside the contracting case not every price is Euclidean, because for $\mu \ge 3$ the price $\sqrt{\operatorname{rk}(a-a')}$ of the proof above is not. Theorem D.2 is not effective: we do not know which case occurs, and the contracting case comes without a value for μ.

Appendix E. The flat case

In this section we prove Theorem E.11, the lower bound of Theorem L.1 for the permanent in the flat case of Definition D.1. As in Section 2, we treat the polynomial $f_{k,\ell}$ of the complete network, which is a permanent of size about $2^{k\ell}$, and write it by Lemmas E.4 and A.3 as a sum of networks through two hubs. Only the statements that say so need the flat case. Section E.3 proves a first bound with an unspecified threshold on ℓ, which shows the method, and Sections E.4 to E.6 make the threshold explicit.

E.1. **Sums over bits as permanents.** The next proposition replaces Valiant's completeness theorem [Val79], for which [Bür00] and [Bür24, Section 2.7 of the arXiv version] give the classical constructions.

Proposition E.1 (sums over bits as permanents)**.** *Let R be a commutative ring with an element $\frac{1}{2}$, let $v : I \to V$ and $w : I \to K$ be surjections of finite sets, and for every $n \in I$ let W_n be a 2×2 matrix over R with entries $W_n(x,r)$, $x, r \in \{0,1\}$. Then*

$$\sum_{x \in \{0,1\}^V} \sum_{r \in \{0,1\}^K} \prod_{n \in I} W_n(x_{v(n)}, r_{w(n)})$$

is the permanent of an $M \times M$ matrix, for every $M \geq 3|I|$, whose entries are $0, \pm 1, \pm\frac{1}{2}$ and sums of values W_n with the same $w(n)$.

Proof. We read each row of a matrix as a linear form in variables that stand for the columns, so that the permanent is the coefficient of the product of all these variables in the product of all rows. Let σ and ρ be permutations of I whose cycles are the fibres of v and of w, respectively. For every $n \in I$ the matrix has three columns, with variables A_n, B_n, C_n, and three rows, namely

$$\tfrac{1}{2}(A_n + B_n + A_{\sigma n} - C_n), \qquad A_n + B_n - A_{\sigma n} + C_n,$$
$$W_n(1,1)C_n + W_n(0,1)B_n + W_{\rho n}(1,0)C_{\rho n} + W_{\rho n}(0,0)B_{\rho n}.$$

The squarefree part of the product of the first two is $A_nB_n + A_{\sigma n}C_n$, and we call the choice of the first monomial $x = 1$ at n and that of the second $x = 0$ at n. Every variable A_m is used exactly once if and only if x is constant on every cycle of σ, and then one variable is left at every n, namely C_n if $x = 1$ and B_n if $x = 0$. We say that $r = 1$ at n if the third row at n takes one of its first two terms, and $r = 0$ otherwise. Every variable that was left is used exactly once if and only if r is constant on every cycle of ρ, and then the product of the coefficients is $\prod_n W_n(x, r)$. A diagonal of ones pads the matrix to any larger size. □

We apply the proposition to sums of the following kind.

Definition (clause sums)**.** Let V be a finite set, whose elements we call bits. A *literal* is a condition of the form $x_v = 1$ or $x_v = 0$ on a single bit. A *clause* consists of a nonempty list of literals and a weight. Let K be a finite set of clauses such that every bit occurs in at least one literal, and let w_j be the weight of the clause $j \in K$. The *clause sum* is

$$\sum_{x \in \{0,1\}^V} \prod_{j \in K} \begin{cases} w_j & \text{if } x \text{ satisfies all literals of the clause } j, \\ 1 & \text{otherwise.} \end{cases}$$

Its number of literals is the total length of the lists of all clauses.

Clauses of weight 1 may be added, so that the condition that every bit occurs in a literal is no restriction.

Corollary E.2 (clause sums as permanents)**.** *Let $2 \neq 0$ in $\Bbbk$. A clause sum with q literals in all, whose weights are constants or variables, is the permanent of an $M \times M$ matrix, for every $M \geq 3q$, with entries that are affine in one variable each.*

Proof. We sum over a bit r_j for every clause j, in which $r_j = 0$ contributes 1 and $r_j = 1$ contributes $w_j - 1$ times the indicators of the literals of the clause, and we put the factor $w_j - 1$ on the first literal. Then we apply Proposition E.1 with I the set of all literals, with the map $I \to V$ that sends a literal to its bit, and with the map $I \to K$ that sends it to its clause. A sum of values W_n with the same clause j is then affine in w_j. □

E.2. **The network polynomial.** Let $k \geq 2$ and $\ell \geq 1$, and write $[k] = \{1, \dots, k\}$. The *complete network* on k nodes with ℓ bits on every edge is the complete graph on $[k]$ with a string $\sigma(ab) \in \{0,1\}^\ell$ on every edge ab. Let σ run over all such assignments, and let $\sigma_a = (\sigma(ab))_{b \neq a}$ collect the strings on the edges at the node a. With a variable $x_{a,\rho}$ for every node a and every $\rho \in \{0,1\}^{(k-1)\ell}$, the *network polynomial* is

$$f_{k,\ell} = \sum_{\sigma} \prod_{a=1}^{k} x_{a,\sigma_a}.$$

The variables $x_{a,\rho}$ with a fixed a form the ath *block* of variables, and $f_{k,\ell}$ is linear in each of the k blocks, or set-multilinear in the usual terminology.

Example. The polynomial $f_{3,1} = \sum_{p,q,r \in \{0,1\}} x_{1,pq}\, x_{2,pr}\, x_{3,qr}$ is, up to the names of the variables, the trilinear form of the multiplication of 2×2 matrices.

The coefficients of $f_{k,\ell}$ form a tensor on the k nodes, in which the node a has the index $\rho_a \in \{0,1\}^{(k-1)\ell}$. We give each of the $k-1$ strings of ρ_a to a party of its own, the *end* (a,b) of the edge ab at a (a half-edge in the language of graph theory). On the $k(k-1)$ ends this tensor is

$$T_{k,\ell} = \prod_{a<b} \langle 2^\ell \rangle_{(a,b)(b,a)},$$

which has one wire of dimension 2^ℓ for every edge. A small circuit for a permanent of size about $2^{k\ell}$ gives one for $f_{k,\ell}$.

Lemma E.3 (the network polynomial as a permanent)**.** *Let $2 \neq 0$ in $\Bbbk$, and let $k \geq 2$ and $\ell \geq 1$. If PERM_M has a circuit over $\Bbbk$ with L gates and $M \geq 3k^3\ell\, 2^{k\ell}$, then $f_{k,\ell}$ has one with $5L$ gates.*

Proof. The polynomial $f_{k,\ell}$ is the clause sum over the bits of σ in which, for every node a and every ρ, there is a clause with the weight $x_{a,\rho}$ whose literals say that $\sigma_a = \rho$. It has $k(k-1)\ell\, 2^{(k-1)\ell} \leq M/3$ literals, so by Corollary E.2 it is the permanent of an $M \times M$ matrix whose entries are affine in one variable each. We substitute these entries for the variables of the given circuit, and an affine function of one variable takes five gates. □

The next lemma concerns arbitrary circuits, and its proof is the argument behind depth reduction [Hya79, VSBR83]. We call a set $S \subseteq [k]$, or the set of blocks with numbers in S, *balanced* if $k \leq 3|S| \leq 2k$.

Lemma E.4 (balanced products)**.** *Let a polynomial over a commutative ring in $k \geq 2$ blocks of variables of equal size have a circuit with L gates, and let f be its part that is linear in each block. Then*

$$f = \sum_{S \text{ balanced}} \sum_{j=1}^{L} g_{S,j}\, h_{S,j},$$

where $g_{S,j}$ is linear in each block of S, $h_{S,j}$ is linear in each block outside S, and neither contains other variables.

Some of the products may be zero, so that there are at most L products for every balanced set.

Proof. We argue by induction along the circuit. For a gate γ and a set S of blocks let γ_S be the part of γ of degree one in each block of S and zero in the others. We show that if $3|S| > 2k$, then $\gamma_S = \sum_{\delta,A} \delta_A\, h_{\delta,A}$ over the gates δ and the balanced subsets A of S, where $h_{\delta,A}$, which depends on γ and S, is linear in each block of $S \setminus A$ and free of the other blocks. Variables and constants have $\gamma_S = 0$, because S has at least two elements, and for a sum of two gates we add their representations. For a product $\gamma = \gamma_1\gamma_2$ we have $\gamma_S = \sum (\gamma_1)_A (\gamma_2)_B$ over all $A \sqcup B = S$. In each term the larger of the two sets, say A, has more than $k/3$ elements, so either A is balanced and the term has the required form, or the induction hypothesis applies to $(\gamma_1)_A$ and we multiply by $(\gamma_2)_B$. Finally we take γ to be the output and $S = [k]$, so that $\gamma_S = f$, and collect the terms by (δ, A), of which there are at most L for every balanced A. □

For a circuit for $f_{k,\ell}$ the lemma says that $T_{k,\ell} = \sum_S t_S$ over the balanced sets S, where t_S is a sum of at most L products of a tensor on the ends (a,b) with $a \in S$ and a tensor on the other ends, which is the form to which Lemma A.3 applies.

E.3. **A first lower bound.** We first prove a lower bound for $f_{k,\ell}$ that holds for all sufficiently large ℓ.

Proposition E.5 (a first lower bound for the network polynomial)**.** *In the flat case, for every $k \geq 2$ there is ℓ_0 such that for $\ell \geq \ell_0$ every circuit over $\Bbbk$ for $f_{k,\ell}$ has at least $2^{\ell k^{3/2}/6}$ gates.*

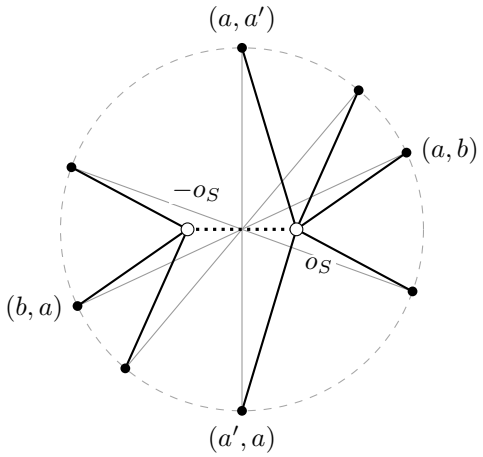


FIGURE 7. The ends of four edges and the two hubs of a balanced set S. The ends (a, b) with $a \in S$ are joined to o_S, the others to $-o_S$, and the hubs by a wire of dimension L (dotted).

Proof. We place the ends, and two hubs for every balanced set, in $\mathbb{R}^k$, and use the flat case to compare Euclidean lengths with the values of a spectral point on the decomposition of $T_{k,\ell}$ given by a circuit.

The ends. Let $e_1, \dots, e_k$ be the standard basis of $\mathbb{R}^k$ and let $u_{ab} = (e_a - e_b)/\sqrt{2}$. We put the end (a, b) at $\frac{1}{2}u_{ab}$ (Figure 7), so that the two ends of each of the $N = \binom{k}{2}$ edges have distance 1.

The hubs. For a balanced set S let $w_S = \sum_{a\in S,\, b\notin S} u_{ab}$, so that $\|w_S\|^2 = \frac{1}{2}k|S|(k-|S|) \geq k^3/9$. Since $u_{ab} + u_{ba} = 0$, the vectors u_{ab} with $a \in S$ and $b \neq a$ add up to w_S, and the others to $-w_S$. Let the hubs of S be $\pm o_S = \pm\eta\, w_S/\|w_S\|$ with $\eta = \frac{1}{6\sqrt{k}}$, regarded as different parties for different S, and join every end (a, b) with $a \in S$ to o_S and every other end to $-o_S$. For a unit vector u and a vector o we have $\|\frac{1}{2}u - o\| \leq \frac{1}{2} - u \cdot o + \|o\|^2$, because $(\frac{1}{2} + t)^2 \geq \frac{1}{4} + t$. Hence the distances from the ends to their hubs add up to at most

$$N - 2\eta\|w_S\| + k^2\eta^2 \leq N - \frac{k}{12}.$$

The spectral point. Let $\delta = \frac{1}{150k}$. Because we are in the flat case, there are a spectral point ϕ on the set of the ends and of all hubs and a factor $c \geq 0$, both depending on k only, such that $c\, d_\phi$ differs from the Euclidean distances of the positions by less than δ. We put $\ell_0 = \lceil 100c \rceil + 1$.

The comparison. Let $\ell \geq \ell_0$ and let L be the size of a circuit for $f_{k,\ell}$. By Lemmas E.4 and A.3, with $\pm o_S$ as the two further parties, $T_{k,\ell} = \sum_S t_S$, where t_S is a restriction of the tensor

$$T_S = \prod_{a\in S,\, b\neq a} \langle 2^\ell \rangle_{(a,b),\, o_S} \cdot \langle L \rangle_{o_S, -o_S} \cdot \prod_{a\notin S,\, b\neq a} \langle 2^\ell \rangle_{(a,b),\, -o_S}$$

of Figure 7. Thus $T_{k,\ell}$ is covered by at most 2^k tensors T_S, and $\phi(T_{k,\ell}) \leq 2^k \max_S \phi(T_S)$ by Definition A.1. We take logarithms, use Lemma A.2 for every wire, multiply by c, and replace each $c\, d_\phi$ by the Euclidean distance, which gives for some S

$$\ell N(1-\delta) \leq ck + \ell\Big(N - \frac{k}{12} + 2N\delta\Big) + (2\eta + \delta)\log L. \tag{E.1}$$

The conclusion. Here $ck \leq \ell k/100$ by the choice of ℓ_0, and $3N\delta \leq k/100$. Hence $\ell k(\frac{1}{12} - \frac{1}{50}) \leq (2\eta + \delta)\log L \leq 0.34\, k^{-1/2} \log L$, and this gives $\log L \geq \ell k^{3/2}/6$. □

With Lemma E.3 this excludes circuits of polynomial size for the permanent.

Theorem E.6 (no circuits of polynomial size in the flat case)**.** *In the flat case, for every C there is m_0 such that for $m \geq m_0$ no circuit over $\mathbb{k}$ with at most m^C gates computes* PERM_m.

Proof. We apply Proposition E.5 with a fixed k for which $\sqrt{k} \geq 30(C+1)$. For large m let ℓ be the largest integer with $3k^3\ell\, 2^{k\ell} \leq m$, so that $m < 3k^3(\ell+1)2^{k(\ell+1)}$, and hence $\log m \leq 2k\ell$ and $\ell \geq \ell_0$. By Lemma E.3 a circuit with at most m^C gates for PERM_m then gives $\ell k^{3/2}/6 \leq 3 + 2Ck\ell$, which is false by the choice of k. □

E.4. **Sums with few terms.** In (E.1) the k bits that name a balanced set are multiplied by the unknown factor c, which is the reason for the unspecified threshold ℓ_0. The following lemma replaces the factor J in the inequality for covers of Definition A.1 by $J^{g_\phi(P)}$ when the J terms arise by applying functionals outside a set P of parties. With P the set of all ends, the term ck becomes k times the scaled exponent $c\,g_\phi(P)$, which Section E.5 bounds by about $\sqrt{k}$.

Lemma E.7 (sums with few terms)**.** *Let $P \subseteq V$ be nonempty, let t be a tensor on P, and let $T_1, \dots, T_J$ be tensors on V that have the same index sets as t at the parties of P. Suppose that $t = \sum_j f_j(T_j)$, where each f_j consists of one linear functional at every party outside P, the trivial one where there is no index, and $f_j(T_j)$ is the tensor on P that results when these functionals are applied to the indices of T_j outside P. Then $\phi(t) \leq J^{g_\phi(P)} \max_j \phi(T_j)$ for every spectral point ϕ on V.*

Proof. We expand a high power t^n by words with n letters in $[J]$, and group the words by their *type* $\tau = (\tau_1, \dots, \tau_J)$, in which τ_j is the number of places with the letter j. We number the $M_\tau \leq J^n$ words of type τ. The sum over these words is a restriction of $\langle M_\tau \rangle_P \cdot \prod_j T_j^{\tau_j}$, because every party of P can order its n indices as the word whose number it holds prescribes, while the parties outside P apply $\prod_j f_j^{\tau_j}$. There are at most $(n+1)^J$ types, so

$$\phi(t)^n \leq (n+1)^J J^{n g_\phi(P)} \max_j \phi(T_j)^n.$$

Taking nth roots and letting n tend to infinity proves the lemma. □

E.5. **The exponent of the set of all ends.** The scaled exponent $c\,g_\phi(P)$ of two points is their distance, and the next lemma allows us to pass to larger sets P by taking midpoints. For a map π of a finite set I into V we write $g_\phi(\pi) = g_\phi(\pi(I))$ for the exponent of its image.

Lemma E.8 (the exponent is almost convex along midpoints)**.** *Let $V \subseteq \mathbb{R}^n$ and $I \neq \emptyset$ be finite. For every $\varepsilon > 0$ there is $\delta > 0$ with the following property. Let ϕ be a spectral point on V and $c \geq 0$ with $\left|c\,d_\phi(x,y) - \|x-y\|\right| < \delta$ for all $x, y \in V$, and let $a, b, m : I \to V$ be maps such that every $m(i)$ is the midpoint of $a(i)$ and $b(i)$. Then*

$$c\,g_\phi(m) \leq \tfrac{1}{2} c\,g_\phi(a) + \tfrac{1}{2} c\,g_\phi(b) + \varepsilon.$$

Proof. We add a new party i' for every $i \in I$, extend ϕ as if all i' were placed at $a(i)$ or all at $b(i)$, each with probability $\frac{1}{2}$, and use Theorem C.1 to show that i' is close to $m(i)$. Let $Q = \{i' : i \in I\}$, and let $\pi_a, \pi_b : V \cup Q \to V$ be the identity on V and map i' to $a(i)$ and to $b(i)$. Proposition A.7, with the weights $\frac{1}{2}, \frac{1}{2}$ and with C the product of the wires of dimension 2 on V and of $\langle 2 \rangle_{m(I)}$, gives a spectral point ϕ' on $V \cup Q$ with $\phi'(t) \leq \phi(\pi_a t)^{1/2} \phi(\pi_b t)^{1/2}$ for all t and with the same values as ϕ on the factors of C, because each can only decrease and the product is kept. Thus $d_{\phi'} = d_\phi$ on V and $g_{\phi'}(m(I)) = g_\phi(m)$, and since $\pi_a \langle q \rangle_Q$ is a restriction of $\langle q \rangle_{a(I)}$, and likewise for b and for wires,

$$g_{\phi'}(Q) \leq \tfrac{1}{2} g_\phi(a) + \tfrac{1}{2} g_\phi(b), \qquad d_{\phi'}(i', x) \leq \tfrac{1}{2} d_\phi(a(i), x) + \tfrac{1}{2} d_\phi(b(i), x) \quad (x \in V).$$

Contracting $\langle q \rangle_Q$ with the wires $\langle q \rangle_{i'\, m(i)}$ gives $\langle q \rangle_{m(I)}$, and hence $g_\phi(m) \leq g_{\phi'}(Q) + \sum_i d_{\phi'}(i', m(i))$. Now fix i, measure distances in the price $c\,d_{\phi'}$, and let λ be the distance of $a(i)$ and $b(i)$. Then i' has distance at most $\lambda/2$ from both, so it is an exact

midpoint, and $m(i)$ has distance less than $\frac{\lambda}{2} + \frac{3}{2}\delta$ from both. Theorem C.1, with i' as the midpoint, $D = \lambda/2$ and $z = m(i)$, bounds the square of the distance of i' and $m(i)$ by $\frac{3}{2}\delta\lambda + (\frac{3}{2}\delta)^2$, which tends to 0 with δ, uniformly in ϕ, c and the maps a, b, m, because λ is at most the diameter of V plus δ. Inserting this bound for every i into the inequality for $g_\phi(m)$, multiplied by c, gives the claim for small δ. □

We apply the lemma to sets of facet centres of the cube $[-\frac{1}{2}, \frac{1}{2}]^k$, that is, of the $2k$ points $\pm\frac{1}{2}e_i$ (Figure 8).

Proposition E.9 (facet centres of a cube)**.** *In the flat case let $V \subseteq \mathbb{R}^k$ be finite, let $P \subseteq V \cap \{\pm\frac{1}{2}e_1, \dots, \pm\frac{1}{2}e_k\}$ be nonempty, and let $\varepsilon > 0$. There are a spectral point ϕ on V and $c \geq 0$ with $\big|c\, d_\phi(x, y) - \|x - y\|\big| < \varepsilon$ for all $x, y \in V$ and $c\, g_\phi(P) \leq \sqrt{k} + \varepsilon$.*

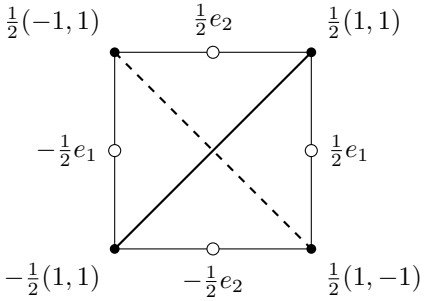


FIGURE 8. Proposition E.9 for $k = 2$. Every facet centre (circles) is the midpoint of an end of the solid diagonal and an end of the dashed one, so by Lemma E.8 the scaled exponent of the four centres is at most about $\sqrt{2}$.

Proof. We pass from the two ends of a main diagonal of the cube to the set P by taking midpoints k times, as in Lemma E.8. We add to V the points $\frac{1}{2}u$ with $u \in \{-1, 0, 1\}^k$, and restrict ϕ to the given set in the end. For a level $0 \leq j \leq k$ and signs $\chi = (\chi_{j+1}, \dots, \chi_k)$ let $\alpha_\chi : P \to V$ map a centre $\pm\frac{1}{2}e_i$ in P to itself if $i \leq j$, and to $\pm\frac{1}{2}\chi_i\,(0, \dots, 0, \chi_{j+1}, \dots, \chi_k)$, with j zeros, if $i > j$. At level k the map is the identity, with image P, and at level 0 its image is contained in the two ends $\pm\frac{1}{2}\chi$ of a main diagonal, so that $g_\phi(\alpha_\chi)$ is at most the exponent of one wire of length $\sqrt{k}$. For $j \geq 1$ the map α_χ of level j is the pointwise midpoint of the two maps $\alpha_{(1,\chi)}$ and $\alpha_{(-1,\chi)}$ of level $j - 1$, as one sees by comparing the three cases $i < j$, $i = j$ and $i > j$. We use the flat case to choose ϕ and c such that $c\, d_\phi$ differs from the Euclidean metric on V by less than $\varepsilon/(k + 1)$, and by so little that Lemma E.8 applies to each of these finitely many triples of maps with $\varepsilon/(k+1)$ in place of ε. Then $c\, g_\phi(\alpha_\chi) \leq \sqrt{k} + (j + 1)\varepsilon/(k + 1)$ at level j by induction on j, which for $j = k$ is the claim. □

E.6. **The lower bound for $\ell \geq 100\sqrt{k}$.** We repeat the proof of Proposition E.5 with the ends at facet centres of a cube and with Lemma E.7 in place of the inequality for covers.

Proposition E.10 (lower bound for the network polynomial)**.** *In the flat case, for $k \geq 2$ and $\ell \geq 100\sqrt{k}$ every circuit over $\Bbbk$ for $f_{k,\ell}$ has at least $2^{\ell k^{3/2}/16}$ gates.*

Proof. The steps and the notation are those of the proof of Proposition E.5.

The ends. For $a < b$ we put the end (a, b) at $\frac{1}{2}e_a$ and the end (b, a) at $-\frac{1}{2}e_a$. The ends thus occupy facet centres of the cube $[-\frac{1}{2}, \frac{1}{2}]^k$, the two ends of an edge have distance 1, and all positions add up to 0.

The hubs. For a balanced set S let w_S be twice the sum of the positions of the ends (a, b), $b \neq a$, with $a \in S$. If c_a is the number of $b > a$ such that exactly one of a, b lies in S, then

$w_S = \sum_a \pm c_a e_a$ and $\sum_a c_a = |S|(k-|S|) \ge 2k^2/9$, so that $\|w_S\| \ge \frac{2}{9}k^{3/2}$ by the Cauchy–Schwarz inequality. We choose the hubs as $\pm o_S = \pm\eta\, w_S/\|w_S\|$ with $\eta = \frac{2}{9\sqrt{k}}$, and join every end to the hub of its side as before. As $\|r-o\| \le \frac{1}{2} - 2\,r\cdot o + \eta^2$ for $\|r\| = \frac{1}{2}$ and $\|o\| = \eta$, the distances from the ends to their hubs add up to at most

$$N - 2\eta\|w_S\| + k^2\eta^2 \le N - \frac{4}{81}k.$$

The spectral point. Let $\delta = 10^{-3}/k$, and let ϕ and c be as in Proposition E.9 for the set V of the positions of all ends and hubs and the set P of the positions of the ends, with δ as ε, pulled back to the ends and hubs regarded as different parties. Then $c\,d_\phi$ differs from the Euclidean distance of the positions by less than δ, and $c\,g_\phi(\mathcal{E}) \le \sqrt{k} + \delta$ for the set $\mathcal{E}$ of all ends, because the push-forward of a unit tensor on $\mathcal{E}$ is the unit tensor on P.

The comparison. For a circuit with L gates we have, as before, $T_{k,\ell} = \sum_S t_S$ over the at most 2^k balanced sets, and t_S is a restriction of T_S. By the proof of Lemma A.3 this restriction applies a functional at o_S and one at $-o_S$ and nothing at the ends, and T_S has no index at the hubs of the other sets. Lemma E.7, with $\mathcal{E}$ as the set P of the lemma and $J \le 2^k$, therefore gives $\phi(T_{k,\ell}) \le 2^{k\,g_\phi(\mathcal{E})} \max_S \phi(T_S)$, and as before we find for some S

$$\ell N(1-\delta) \le k(\sqrt{k}+\delta) + \ell\Big(N - \frac{4}{81}k + \delta k^2\Big) + (2\eta+\delta)\log L. \tag{E.2}$$

The conclusion. Here $k(\sqrt{k}+\delta) \le 1.5\,k^{3/2} \le 0.015\,\ell k$, because $\ell \ge 100\sqrt{k}$, and the two terms with δ add up to at most $\frac{3}{2}\delta k^2\ell = 0.0015\,\ell k$. Hence $(2\eta+\delta)\log L \ge \ell k/31$, and since $2\eta+\delta \le 0.446\,k^{-1/2}$, it follows that $\log L \ge \ell k^{3/2}/14$, which implies the claim. □

For the permanent we take $\ell = 100\sqrt{k}$, the smallest value that Proposition E.10 admits, which makes $\log m$ of order $k^{3/2}$ and the bound of order k^2.

Theorem E.11 (lower bound for the permanent in the flat case)**.** *In the flat case there is m_0 such that for $m \ge m_0$ every circuit over $\Bbbk$ for PERM_m has more than $2^{(\log m)^{4/3}/1600^2}$ gates.*

Proof. We apply Proposition E.10 to a square $k = r^2$. Let r be the largest integer with $200r^3 \le \lfloor \log m \rfloor$, and let $k = r^2$ and $\ell = 100r$. For large m we have $r \ge 2$, $3k^3\ell\,2^{k\ell} \le 2^{200r^3} \le m$ and $\log m \le 1600r^3$. By Lemma E.3 and Proposition E.10, a circuit with L gates for PERM_m gives

$$\log(5L) \ge \frac{\ell k^{3/2}}{16} = \frac{25}{4}r^4.$$

Hence $\log L > r^4 \ge (\log m/1600)^{4/3} \ge (\log m)^{4/3}/1600^2$. □

Appendix F. A Poincaré inequality on unitriangular groups

In this section and in Sections G to K we are in the contracting case, and we fix μ and s as in Definition D.1. Let U_n be the group of the upper unitriangular $n \times n$ matrices over $\mathbb{F}_2$. We number the coordinates by $0, \dots, n-1$ and use the generators $x_j = 1 + E_{j,j+1}$, $0 \le j < n-1$, where E_{jk} is the matrix whose only nonzero entry is a 1 in row j and column k. The generators x_i and x_j commute if $|i-j| \ge 2$.

Only Theorem F.6 is used later. It bounds the mean squared distance of two random elements in a price on U_h by C_0 times the sum, over the generators, of the mean squared distance of two neighbours in the Cayley graph, where C_0 does not depend on h. Section F.1 gives a weaker bound for each U_n separately. In Sections F.2 and F.3 we let U_h act on a CAT(0) space, we show that the sets of points fixed by two overlapping subgroups $\cong U_{2u}$, called windows, meet almost at a right angle, and we reach a common fixed point by averaging projections.

F.1. **Contracting length functions.** A *length function* on a group is a function $g \mapsto |g| \geq 0$ with $|1| = 0$, $|g^{-1}| = |g|$ and $|gg'| \leq |g| + |g'|$. We say that it *contracts* if for every homomorphism θ from the additive group $\mathrm{Mat}_\mu(\mathbb{F}_2)$ into the group, the mean of $|\theta W|^2$ over the matrices W of rank μ is at most s^2 times the mean over the matrices of rank 1.

Remark (length functions from maps into price spaces)**.** Let f be a map of a finite group G into a price space. Then $|g| = \left(\mathbb{E}_a |f(a)f(ga)|^2\right)^{1/2}$, with a uniformly distributed in G, is a length function on G that contracts. Indeed, for a homomorphism $\theta : \mathrm{Mat}_\mu(\mathbb{F}_2) \to G$ and $a \in G$, the pull-back of the metric along $W \mapsto f(\theta(W)a)$ is a price on $\mathrm{Mat}_\mu(\mathbb{F}_2)$, so it has $D(\mu)^2 \leq s^2 D(1)^2$ by the contracting hypothesis, and the mean over a is the claim.

The next lemma extends the contraction from the rank μ to all ranks, with the letter D used as in Section D.

Lemma F.1 (growth of the mean length with the rank)**.** *Let $1 \leq r \leq k$, let θ be a homomorphism from the additive group $\mathrm{Mat}_{k\times k}(\mathbb{F}_2)$ into a group with a contracting length function, and let $D(r)$ be the root mean square of $|\theta W|$ over the matrices W of rank r. Then $D(r) \leq \mu(j+1)s^j D(1)$, where $j = \lfloor \log_\mu r \rfloor$.*

Proof. We show that $D(\mu i) \leq s\, D(i)$ if $\mu i \leq k$ and that $D(i+i') \leq D(i) + D(i')$ if $i + i' \leq k$, and then we write r in base μ. For the first claim we compose θ with $Z \mapsto \alpha(Z \otimes 1_i)\alpha'$, where 1_i is the identity matrix of size i and α and α' are uniformly random $k \times \mu i$ and $\mu i \times k$ matrices of full rank. This homomorphism from $\mathrm{Mat}_\mu(\mathbb{F}_2)$ multiplies ranks by i, and the image of a fixed Z is uniformly distributed among the matrices of its rank, so the mean of the contraction inequality over α, α' is $D(\mu i)^2 \leq s^2 D(i)^2$. The second claim follows by Minkowski's inequality from $|\theta(W + W')| \leq |\theta W| + |\theta W'|$ for a random transversal pair (W, W') as in Step 1 of the proof of Theorem D.2. Finally, if $r = \sum_{t \leq j} d_t \mu^t$ with digits $0 \leq d_t < \mu$, then the two claims give $D(r) \leq \sum_t d_t s^t D(1) \leq \mu(j+1)s^j D(1)$, because $s \geq 1$. □

The next lemma bounds $D(1)$, with U_n divided into four blocks between the coordinates $k-1$ and k.

Lemma F.2 (length of the elements of rank one)**.** *Let $1 \leq k < n$, and let $\xi \in \mathbb{F}_2^k$ and $\omega \in \mathbb{F}_2^{n-k}$ be nonzero. The element of U_n with the upper right block $\xi\omega^\mathsf{T}$ and with unit diagonal blocks has, in every length function on U_n, length at most $18\sum_j |x_j|$.*

Proof. We write the element as a word in which each generator occurs at most 16 times. Suppose first that the last entry of ξ and the first entry of ω are 1, let e be the last unit vector of $\mathbb{F}_2^k$, and let $i_1 < \cdots < i_l$ be the places where ξ has a 0. The product $x_0 \cdots x_{k-2}$ maps e to $(1, \ldots, 1)$, so $A = x_{i_l} \cdots x_{i_1} x_0 \cdots x_{k-2}$ satisfies $Ae = \xi$ and uses each generator at most twice. Likewise a word B in the x_j with $j \geq k$ uses each of them at most twice and has $e'^\mathsf{T} B = \omega^\mathsf{T}$ for the first unit vector e' of $\mathbb{F}_2^{n-k}$. As x_{k-1} has the upper right block ee'^T and unit diagonal blocks, the element in question is $AB^{-1}x_{k-1}BA^{-1}$, a word that uses each generator at most four times.

In general ξ and ω are sums of at most two such vectors each, so our element is a product of at most four elements of the kind just treated. □

We now combine the two lemmas. The rate $\vartheta_n^2 n = o(n^2)$ is what allows the hypothesis of Lemma F.4 below to be met with a small ζ.

Lemma F.3 (mean length on U_n)**.** *There are numbers ϑ_n with $\vartheta_n^2 n = o(n^2)$ such that every contracting length function on U_n satisfies $(\mathbb{E}_g |g|^2)^{1/2} \leq \vartheta_n \sum_j |x_j|$.*

Proof. The proof is by induction along the powers of two. This case suffices, because U_n is a quotient of every larger $U_{n'}$ in such a way that generators go to generators or to 1, so that $\vartheta_{n'}$ also serves as ϑ_n. So let $n = 2^{i+1}$, and divide the matrices in U_n into four blocks

of size $n/2$. A uniformly distributed element of U_n is the product of a uniformly distributed element of the abelian upper right block, which is a copy of $\mathrm{Mat}_{n/2\times n/2}(\mathbb{F}_2)$, and of uniformly distributed elements of the two diagonal blocks, which are copies of $U_{n/2}$. By Lemmas F.1 and F.2 the first factor has root mean square length at most $18\mu(d+1)s^d\sum_j|x_j|$, where $d=\lfloor\log_\mu n\rfloor$. The induction hypothesis applies to the two other factors, which use disjoint sets of generators. By Minkowski's inequality we may therefore take $\vartheta_n=\vartheta_{n/2}+18\mu(d+1)s^d$ and $\vartheta_1=0$, which gives $\vartheta_n=O\big((\log n+1)^2 s^{\log_\mu n}\big)$, and $\vartheta_n^2 n=o(n^2)$ because $s^2<\mu$. □

F.2. **Windows and their fixed points.** Let U_h act by isometries on a CAT(0) space Z, and suppose that for every $z\in Z$ the length function $g\mapsto|z,gz|$ on U_h contracts. For $p+n\le h$ the *window* $[p,p+n)$ is the subgroup of the matrices that differ from the identity only in the rows and columns $p,\dots,p+n-1$, a copy of U_n generated by $x_p,\dots,x_{p+n-2}$.

The points of Z that a window fixes form a closed convex set, which contains the barycenter of every orbit of the window. We write $P_{[p,p+n)}$ for the nearest-point projection to this set (Lemma C.4(d)). By Lemma C.4(c) for the orbit of a point z we have $|z,Pz|^2\le\mathbb{E}_g|z,gz|^2$ for $P=P_{[p,p+n)}$, with g uniformly distributed in the window. Let $\delta_j(z)=|z,x_jz|$. Lemma F.3 for the window and the Cauchy–Schwarz inequality now show that

$$|z,P_{[p,p+n)}z|^2\le C(n)\sum_{p\le j<p+n-1}\delta_j(z)^2,\qquad C(n)=\vartheta_n^2 n=o(n^2). \tag{F.1}$$

If two windows have no coordinate in common, then they commute, and therefore the projection of one of them maps the fixed points of the other to fixed points of the other.

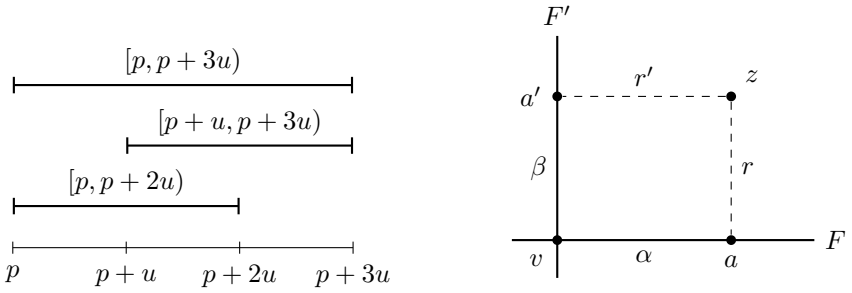


FIGURE 9. Left: two overlapping windows and the window that they generate. Right: the sets F and F' of the points that the two short windows fix, and the points of Lemma F.5.

The windows $[p,p+2u)$ and $[p+u,p+3u)$ generate $[p,p+3u)$, which therefore fixes exactly their common fixed points (Figure 9). By the next lemma a point that the first window fixes is not much closer to the fixed points of the second than to the common ones.

Lemma F.4 (projections for overlapping windows)**.** *Let $u\ge2$, $p+3u\le h$, $\zeta\ge0$ and $4C(3u)\le\zeta^2(u-1)^2$. If a is fixed by $[p,p+2u)$, then $|a,P_{[p+u,p+3u)}a|\ge(1-\zeta)R$, where $R=|a,P_{[p,p+3u)}a|$.*

Proof. We let $p=0$ and pass from the window $[u,3u)$ to $[0,3u)$ through the windows $[k,3u)$. Let $b_k=P_{[k,3u)}a$ for $u\le k<2u$, so that $|ab_k|\le|ab_u|\le R$.

Let $u<k<2u$. Every generator of $[0,3u)$ fixes b_k, with the possible exception of x_{k-1}. Indeed, those after it fix b_k by definition, and those before it do because they fix a and commute with $[k,3u)$. The generator x_{k-1} fixes b_{k-1}, so it moves b_k by at most $2|b_{k-1}b_k|$, and Lemma C.4(d) gives $|b_{k-1}b_k|^2\le|ab_{k-1}|^2-|ab_k|^2$, since b_{k-1} is fixed by $[k,3u)$.

Let $\bar b$ be the barycenter of the $u-1$ points b_k, $u<k<2u$. By Lemma C.4(c) each generator moves $\bar b$ by at most $1/(u-1)$ times the distance by which it moves the one point

b_k that it moves, so that

$$\sum_j \delta_j(\bar b)^2 \le \frac{4}{(u-1)^2}\sum_{u<k<2u}(|ab_{k-1}|^2-|ab_k|^2) \le \frac{4|ab_u|^2}{(u-1)^2} \le \frac{4R^2}{(u-1)^2}.$$

By (F.1) for the window $[0,3u)$ and by the hypothesis on ζ, the point $\bar b$ is within ζR of a point fixed by $[0,3u)$, and $|a\bar b| \le |ab_u|$ by Lemma C.4(c). Hence $R \le |a\bar b| + \zeta R \le |ab_u| + \zeta R$. □

The following lemma turns this into a law of cosines for the angle between the two sets of fixed points.

Lemma F.5 (almost right angles)**.** *Let u, p, ζ be as in Lemma F.4, and let $0 \le \zeta \le 1$, $c > 0$ and $1 - c^2 \le (1-\zeta)^2$. If a and a' are the projections of a point z for the windows $[p, p+2u)$ and $[p+u, p+3u)$, then $|aa'|^2 \le |za|^2 + |za'|^2 + 2c|za||za'|$.*

Proof. We first bound the angle at a common fixed point v, and then we compare the quadrilateral z, a, v, a' with a Euclidean one (Figure 9). Let $v = P_{[p,p+3u)}a$, and put $r = |za|$, $r' = |za'|$, $\alpha = |av|$ and $\beta = |va'|$. We may assume that these four numbers are positive, as the other cases are immediate. The first step is the inequality

$$\alpha^2 + \beta^2 - 2c\alpha\beta \le |aa'|^2. \tag{F.2}$$

To prove it, let q and q' be the points at the distances $\sigma\alpha$ and $\sigma'\beta$ from v on $[v,a]$ and $[v,a']$, where $\sigma, \sigma' \in (0,1]$ satisfy $\sigma'\beta = c\,\sigma\alpha$. The point q is fixed by $[p, p+2u)$ and its projection for $[p, p+3u)$ is again v, so Lemma F.4 and the hypothesis on c give $|qq'|^2 \ge (1-c^2)\sigma^2\alpha^2$. Lemma C.4(a), applied to q' on $[v,a']$ as seen from q and to q on $[v,a]$ as seen from a', gives the upper bound $|qq'|^2 \le \sigma^2\alpha^2 + \sigma'^2\beta^2 - \sigma\sigma'(\alpha^2+\beta^2-|aa'|^2)$. The two bounds together, with $\sigma'\beta = c\,\sigma\alpha$, are (F.2).

For the second step note that $r^2+\alpha^2$ and $r'^2+\beta^2$ are at most $|zv|^2$, by Lemma C.4(d). Let $\eta, t \in [0,1]$. Lemma C.4(a), applied twice, to the points at η on $[a,a']$ and at t on $[z,v]$, gives

$$t(1-t)|zv|^2 + \eta(1-\eta)|aa'|^2 \le (1-t)(1-\eta)r^2 + (1-t)\eta r'^2 + t(1-\eta)\alpha^2 + t\eta\beta^2.$$

On the left we insert $|zv|^2 \ge (1-\eta)(r^2+\alpha^2) + \eta(r'^2+\beta^2)$, and we choose $\eta = r\alpha/(r\alpha + r'\beta)$ and $t = rr'/(rr'+\alpha\beta)$, which gives

$$(rr'+\alpha\beta)|aa'|^2 \le \alpha\beta(r^2+r'^2) + rr'(\alpha^2+\beta^2).$$

Inserting $\alpha^2+\beta^2 \le |aa'|^2 + 2c\alpha\beta$ from (F.2) on the right gives the claim. □

F.3. **Proof of the Poincaré inequality.** In the main result of this section g and a are independent and uniformly distributed in U_h.

Theorem F.6 (Poincaré inequality)**.** *There are an integer $u \ge 2$ and a real number $C_0 \ge 0$ such that for every $l \ge 2$, $h = lu$, and every price on U_h*

$$\mathbb{E}_{g,a}|a, ga|^2 \le C_0 \sum_{j<h-1} \mathbb{E}_a|a, x_j a|^2.$$

Proof. We let U_h act on the maps z from U_h into a price space, and we show that the sum of the squared distances from z to the fixed points of the windows $[iu, iu+2u)$ decreases by a fixed factor when z is replaced by the barycenter of its projections.

Since $C(n) = o(n^2)$, there is $u \ge 2$ with $4C(3u) \le (u-1)^2/32^2$. We fix it and put $\zeta = 1/32$ and $c = 1/4$. By Proposition C.3 the given price is pulled back from the metric of a price space Y along a map $f : U_h \to Y$. Let Z be the set of all maps $z : U_h \to Y$ with the metric $|zz'| = (\mathbb{E}_a|z(a)z'(a)|^2)^{1/2}$. This is the L^2 product of $|U_h|$ copies of Y with its metric divided by $|U_h|^{1/2}$, so it is CAT(0) by Lemma C.4(e). Let U_h act on it by $(gz)(a) = z(g^{-1}a)$. Then $|z, gz|^2 = \mathbb{E}_a|z(ga)z(a)|^2$, so these length functions contract by the remark in Section F.1, and Section F.2 applies to Z.

Distances between projections. Let $k = l - 1$, and for $0 \le i < k$ let Q_i be the projection for the window $[iu, iu + 2u)$. For $z \in Z$ let $r_i = |z, Q_i z|$ and $\Phi(z) = \sum_i r_i^2$. We claim that

$$|Q_i z, Q_{i'} z|^2 \le r_i^2 + r_{i'}^2 \text{ if } |i - i'| \ge 2, \qquad |Q_i z, Q_{i+1} z|^2 \le (1 + c)(r_i^2 + r_{i+1}^2). \tag{F.3}$$

The second inequality follows from Lemma F.5, as $2r_i r_{i+1} \le r_i^2 + r_{i+1}^2$. For the first, the windows of i and i' have no coordinate in common, so if y is fixed by the window of i, then the point nearest to y among those fixed by the window of i' is fixed by both. This point can therefore serve as v in the proof of Lemma F.5, with $\zeta = 0$ and every $c > 0$, which gives the first inequality.

One step of averaging. Let b be the barycenter of the k points $Q_i z$. Lemma C.4(c) at each of these points and (F.3) give

$$2k \sum_i |b, Q_i z|^2 \le \sum_{i,i'} |Q_i z, Q_{i'} z|^2 \le 2(k - 1 + 2c)\Phi(z).$$

As $\Phi(b) \le \sum_i |b, Q_i z|^2$, and as Lemma C.4(c) at the point z gives $k|zb|^2 \le \Phi(z) - \sum_i |b, Q_i z|^2$, it follows that

$$\Phi(b) \le \Big(1 - \frac{1 - 2c}{k}\Big)\Phi(z), \qquad k|zb|^2 \le \Phi(z) - \Phi(b).$$

If $\Phi(z) > 0$, these two inequalities give

$$|zb| \le \frac{\Phi(z) - \Phi(b)}{\sqrt{(1 - 2c)\Phi(z)}} \le \frac{2}{\sqrt{1 - 2c}}\Big(\sqrt{\Phi(z)} - \sqrt{\Phi(b)}\Big).$$

Iteration. We iterate the map $z \mapsto b$. The values of Φ decrease geometrically and the steps add up to at most $2\sqrt{\Phi(z)/(1 - 2c)}$, so the points converge to a point fixed by every window and hence by U_h, and therefore $|z, P_{[0,h)} z|^2 \le \frac{4}{1-2c}\Phi(z)$. Each generator lies in at most two of the windows, so (F.1) for $n = 2u$ bounds $\Phi(z)$ by $2C(2u) \sum_j \delta_j(z)^2$, and $|z, gz| \le 2|z, P_{[0,h)} z|$ because g fixes $P_{[0,h)} z$. For $z = f$ we have $|f, gf|^2 = \mathbb{E}_a |a, ga|^2$ and $\delta_j(f)^2 = \mathbb{E}_a |a, x_j a|^2$ in the given price, and the theorem follows with $C_0 = 32C(2u)/(1 - 2c)$. □

F.4. **Related work.** For real-valued functions the inequality of Theorem F.6 is a spectral gap for a walk on unitriangular matrices, as in Stong's eigenvalue bounds [Sto95]; for the mixing time of this walk see [PS13, NS23]. Almost right angles between commuting or overlapping subgroups are used in a similar way in proofs of property (T) [DJ02, EJ10, Kas11, Opp17]. Poincaré constants for nonlinear targets, and for CAT(0) targets in particular, are studied in [Wan00, IN05, Gro03, MN13].

Appendix G. Rerouting laws

We are still in the contracting case. In this section the duality theorem (Theorem A.6) turns the Poincaré inequality of Theorem F.6 into identities between tensors, which we call *rerouting laws*, or simply *laws* (Definition G.1). A law writes a product of wires between arbitrary pairs of elements of U_h as a sum of tensor networks whose wires run along edges of the Cayley graph. Theorem G.2 provides finitely many laws whose size is controlled by a constant C_1 that does not depend on h. That their coefficients lie in a finitely generated ring matters only in Section K.

G.1. **Edges, loads and the choice of h.** Let $b = |U_h| = 2^{h(h-1)/2}$. By an *edge* we always mean an edge $\{g, x_j g\}$ of the Cayley graph of U_h for the generators $x_j = 1 + E_{j,j+1}$, $0 \le j < h - 1$, and there are $b/2$ edges for each j. A *load* is a vector $n = (n_e)$ of natural numbers, indexed by all pairs e of distinct elements of U_h or by the edges only, and we measure loads by the norms

$$\|n\| = \Big(\tfrac{b(b-1)}{2} \sum_e n_e^2\Big)^{1/2}, \qquad \|n\|_* = \Big(\tfrac{b}{2} \sum_{\text{edges } e} n_e^2\Big)^{1/2}.$$

For a price on U_h and a pair e we write $|e|$ for the distance between the two elements of e. Let u and C_0 be as in Theorem F.6, let h be a multiple of u with $h \geq 2u$, and fix an integer $C_0' \geq 2C_0 + 1$. Then Theorem F.6 implies that for every price on U_h

$$\frac{2}{b(b-1)} \sum_e |e|^2 \;\leq\; C_0' \cdot \frac{2}{b} \sum_{\text{edges } e} |e|^2,$$

because the left side of its inequality is at least half the mean of $|e|^2$ over all pairs, and the right side is $C_0 \cdot \frac{2}{b} \sum |e|^2$ over the edges. Let $C_1 = 4C_0'$. As C_1 does not depend on h, we can choose h so large that

$$h \geq 3, \qquad 1 + 8C_1 \leq h/4. \tag{G.1}$$

From now on h and b are fixed, and $b \geq 8$.

Formalization note. Of the second condition in (G.1), the proof of Theorem H.2 uses only $1 + 4C_1 \leq h/4$. The Lean proof is coarser at this point and uses (G.1) as it stands.

G.2. **Laws.** Informally, a law is a recipe for replacing wires between arbitrary pairs of elements of U_h, the *slots*, by wires along edges, which in this role are called *channels*. Figure 10 shows the simplest case, a single slot that is rerouted along a path. A law does the same for a whole family of slots at once, and it may use a sum of several networks, numbered by an *index* i.

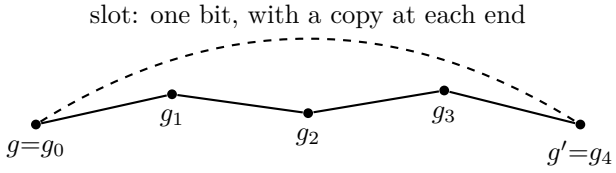


FIGURE 10. A single slot (dashed) rerouted along a path of four channels, each of which carries one bit. Every g_k tests that the bits that it sees agree, and the product of these tests, summed over the bits on the channels, is the wire $\langle 2 \rangle_{gg'}$.

Definition G.1 (law)**.** Let A be a commutative ring. A *law* on U_h over A consists of the following data.

- *Slots.* A finite family of pairs of distinct elements of U_h, the *slots*. They are counted by a load σ on pairs: σ_e is the number of slots equal to e. Every slot carries one bit, and each of its two ends has its own copy of this bit.
- *Indices.* A finite nonempty set $\mathcal{I}$ of *indices*, with an injective *code* that assigns to every index a string of λ bits, where $\lambda \geq 1$.
- *Channels.* For every $i \in \mathcal{I}$ a load len_i on edges. In this role an edge e is called a *channel*, and it carries $\mathrm{len}_{i,e}$ bits.
- *Node tensors.* For every $i \in \mathcal{I}$ and every element g of U_h a function with values in A, the *node tensor* of (i, g). Its arguments are the bits at g of the slots that contain g, and the bits of the channels that contain g.

These data are required to satisfy, for all values of the bits at the ends of the slots,

$$\prod_e [\text{the } \sigma_e \text{ bits agree at both ends of } e] = \sum_{i \in \mathcal{I}} \sum_y \prod_g (\text{node tensor of } (i, g)), \tag{G.2}$$

where e runs over all pairs, y over the assignments of $\mathrm{len}_{i,e}$ bits to every channel e, and g over U_h.

Convention. When a law is applied, every element has to know the index i, so we let every channel begin with the λ bits of the code of i, and give the node tensor 0 to an element whose channels do not all begin with the same code, or begin with a string that is no code. Since the edges connect U_h, codes that agree along all channels agree everywhere. The channel e now carries $\lambda + \mathrm{len}_{i,e}$ bits, and this is the load that Theorem G.2(b) bounds.

G.3. **Existence of laws.** The next theorem provides the laws that we use.

Theorem G.2 (existence of laws)**.** *There are a finitely generated subring A of $\Bbbk$, a positive integer a_1, and finitely many laws on U_h over A such that*

(a) *for every load n on pairs and every $t \in \mathbb{N}$ with $\|n\| \le a_1 t$ some law has $n \le t\sigma$;*
(b) *for every law and every index i of it, $\|\lambda + \mathrm{len}_i\|_* \le \sqrt{C_1}\, a_1$.*

Here $n \le t\sigma$ is meant entry by entry, and $\lambda + \mathrm{len}_i$ is the load of the convention above.

Proof. We apply the duality theorem to an inequality between spectral points that follows from the Poincaré inequality, and read the resulting cover as a law.

Let a be a large integer. The loads $\sigma_x = \lceil ax \rceil$, for real vectors $x \ge 0$ on pairs with $\|x\| \le 1$, are finitely many, and $\|\sigma_x\| \le a + O(1)$. Let $\mathcal{B}$ be the finite set of loads $\lceil y \rceil$ on edges, where $y \ge 0$ is a real vector with $\|y\|_* \le \sqrt{C_0'}\,(a + O(1))$, with the same $O(1)$. For every price on U_h and every x, the Cauchy–Schwarz inequality, applied twice, and the Poincaré inequality in the form given above yield

$$\sum_e \sigma_{x,e}|e| \le \max_{\beta \in \mathcal{B}} \sum_{\text{edges } e} \beta_e |e|.$$

The second application is to the vector y on edges that is proportional to $(|e|)$ and has $\|y\|_* = \sqrt{C_0'}\,\|\sigma_x\|$, for which it is an equality, and $y \le \lceil y \rceil \in \mathcal{B}$. (If $|e| = 0$ on all edges, the left side vanishes as well, by the Poincaré inequality.)

For a load σ, on pairs or on edges, let $\mathsf{W}^\sigma = \prod_e \langle 2 \rangle_e^{\sigma_e}$, a tensor on the set of parties U_h. For a spectral point ϕ on U_h we have $\phi(\mathsf{W}^\sigma) = 2^{\sum_e \sigma_e d_\phi(e)}$ by Lemma A.2, and d_ϕ is a price. Hence $\phi(\mathsf{W}^{\sigma_x}) \le \max_{\beta \in \mathcal{B}} \phi(\mathsf{W}^\beta)$, which is the hypothesis of Theorem A.6. For every $\varepsilon > 0$ and every large q the theorem covers $\mathsf{W}^{q\sigma_x}$ by at most $2^{\varepsilon q}$ tensors $\mathsf{W}^{\beta'}$ in which β' is a sum of q elements of $\mathcal{B}$, and one q serves for the finitely many σ_x.

For every x this cover, with its linear maps written out, is a law: the left side of (G.2) is $\mathsf{W}^{q\sigma_x}$, so the slots are counted by $q\sigma_x$, the indices number the tensors of the cover, len_i is the sum β' of the ith of them, and the node tensor of (i, g) is the matrix of the linear map that the party g applies to $\mathsf{W}^{\beta'}$. The indices get an injective code of λ bits with $1 \le \lambda \le \varepsilon q + 2$. Let $a_1 = aq$, and let A be the subring of $\Bbbk$ generated by -1 and the entries of the node tensors of these finitely many laws.

By the triangle inequality the norm of $\lambda + \mathrm{len}_i$ is at most $q\big(\sqrt{C_0'}\,a + O(1)\big) + (\varepsilon q + 2)\|1\|_*$, where 1 is the load that is 1 on every edge. This is at most $2\sqrt{C_0'}\,aq = \sqrt{C_1}\,a_1$ if we choose first a large, then ε small, and then q large, which proves (b). For (a), let $\|n\| \le a_1 t$. If $t > 0$, then $x = n/(a_1 t)$ has $\|x\| \le 1$ and $n \le t\, q\sigma_x$, so the law that belongs to x will do. If $t = 0$, then $n = 0$ and every law will do. □

The laws are finitely many polynomial identities with coefficients in A, so they hold over every commutative A-algebra. From now on A, a_1 and the laws, in some fixed order, are kept, and all constants below depend only on these choices.

Appendix H. Every network is a sum of tree networks

In this section we apply the laws of Theorem G.2 again and again, along a tower of finite sets, to an arbitrary tensor network. By Theorem H.2, the value of a network on M nodes is a sum of values of networks whose nodes form a tree of small depth, in which the bound

for the packets at any one node is smaller by a factor M^γ than the bound for all packets together. In Sections I and J the network will be a computation, and the tree will show where to cut it. Here $\gamma = 1/\log b$.

H.1. **A tower of finite groups.** To apply the laws repeatedly we use groups with h types of edges, arranged in a cycle, such that leaving out any one type leaves copies of the Cayley graph of U_h. Let

$$R_n = \mathbb{F}_2[s]/(s^{n+1}), \qquad G_n = \mathrm{SL}_h(R_n),$$

number the coordinates by $\mathbb{Z}/h$, and put $\varsigma = 1 + s$, $X_k = 1 + E_{k-1,k}$ for $k \neq 0$ and $X_0 = 1 + \varsigma E_{h-1,0}$, where E_{jk} is a matrix unit. The letters s and ς have this meaning in the present subsection only. Let K_n be the kernel of the reduction $s \mapsto 0$ from G_n to $\mathrm{SL}_h(\mathbb{F}_2)$, a 2-group of order $2^{(h^2-1)n}$.

Proposition H.1 (tower)**.** *Let $n \geq 1$ and $M = |G_n|$. There are G_n-sets $V_0 = G_n, V_1, \dots, V_J$, with equivariant projections $\pi : V_j \to V_{j+1}$ that are b to one, such that*

(a) *for every $i \in \mathbb{Z}/h$ the X_k, $k \neq i$, generate a copy of U_h (with $X_{i+1}, \dots, X_{i-1}$ as the generators in order), which acts freely on every V_j; we call its orbits* cosets of type i*;*

(b) *c and $X_i c$ always lie in different cosets of type i, and π maps every coset of type i bijectively onto one, exactly b of them onto each;*

(c) *$|V_J| < b\,|\mathrm{SL}_h(\mathbb{F}_2)|$ and $|G_{n+1}| = 2^{h^2-1}|G_n|$; thus $|V_J| \leq$ const, and M runs through a sequence with bounded quotients as n varies;*

(d) *every element of G_n is a word of at most $\mathsf{L} = O((n+1)^7)$ letters X_k (the proof gives the exponent 3; we state 7 as in the formal development).*

Proof. We take the sets V_j to be quotients of G_n by a chain of subgroups of K_n. Choose subgroups $\{1\} = \Gamma_0 \subset \cdots \subset \Gamma_J$ of K_n, each of index b in the next, with $[K_n : \Gamma_J] < b$, which is possible in a 2-group. Let $V_j = G_n/\Gamma_j$, with G_n acting from the left, and let π be the natural projections.

(a): Let $Pe_k = e_{k+1}$ for $k \neq h-1$ and $Pe_{h-1} = \varsigma^{-1}e_0$. Then $g \mapsto PgP^{-1}$ is an automorphism of G_n that maps K_n to itself and X_k to X_{k+1} for every $k \in \mathbb{Z}/h$, so each of these groups is the image of U_h, the group generated by the X_k with $k \neq 0$, under a power of it. Now U_h meets the normal subgroup K_n trivially, hence so do its images, and they act freely because the stabilizers of the points of V_j lie in K_n.

(b): Modulo s the element X_i is not in the group generated by the others, because of its entry $(i-1, i)$, so, as stabilizers lie in K_n, the elements c and $X_i c$ lie in different cosets of type i. As π is equivariant and both actions are free, it maps a coset of type i bijectively onto one, and the preimage of a coset of type i has b^2 elements and consists of cosets of type i, hence of b of them.

(c): We have $|V_J| = |\mathrm{SL}_h(\mathbb{F}_2)|\,[K_n : \Gamma_J]$ with $[K_n : \Gamma_J] < b$, and $|G_n| = |\mathrm{SL}_h(\mathbb{F}_2)|\,2^{(h^2-1)n}$.

(d): Write $e_{jk}(r) = 1 + rE_{jk}$. The $e_{jk}(1)$ with $j < k$ belong to U_h, the group generated without $X_0 = e_{h-1,0}(\varsigma)$. As $h \geq 3$, the rule $[e_{jk}(r), e_{kl}(r')] = e_{jl}(rr')$ gives from these and X_0 all $e_{jk}(\varsigma)$ by words of bounded length, and then, by halving a, all $e_{jk}(\varsigma^a)$ by words of length $O(a^2)$. The powers $\varsigma, \dots, \varsigma^{n+1}$ span R_n, and Gaussian elimination over this local ring writes every matrix of G_n as a product of boundedly many elementary matrices $e_{jk}(r)$. This gives words of length $O((n+1)^3)$, which is stronger than (d). □

Formalization note. The exponent 7 in part (d) is the one that is proved in the Lean development; the argument above gives the exponent 3.

An *edge of type i* of V_j is a pair $\{c, X_i c\}$, and the edges inside a coset of type i are those of the other types. For each coset ω of type i of V_{j+1} we fix a bijection $\chi \mapsto \hat{\chi}$ from the b cosets of type i of V_j over ω to the b elements of ω. These groups, with the subgroups generated by all but one of the X_k, are close to those of the coset complexes of Kaufman and Oppenheim [KO18, KO23].

H.2. **Packets, networks and the two moves.** From now on wires are moved in *packets*, strings of ℓ bits for a fixed $\ell \geq 1$. A string w is written on packets as $w10\cdots0$, so it takes $\lfloor |w|/\ell \rfloor + 1$ of them, and w can be recovered from them. A *network* on a finite set of nodes assigns a number of packets to every pair of nodes, and to each node a *node tensor*, a function, with values in a ring, of the bits of the packets at this node. Its *value* is the sum, over all contents of all packets, of the product of the node tensors. This is a tensor network in the usual sense [MS08b, Orús14], and tensor networks as a model of algebraic computation are studied in [DHM02, CDM16, AKK22]. There are two kinds of moves, neither of which changes the value.

Definition (relay)**.** A *relay* through a node w replaces a packet between two other nodes x and y by a packet between x and w and one between w and y, with a test at w that these two are equal. It leaves no copy of the old packet, as the node tensors of x and y read the new packets instead.

Definition (law move)**.** Let ω be a coset of type i in some V_j whose elements are nodes of a network, identified with U_h by $g \mapsto gz_0$ for one of its elements z_0, so that the edges of U_h become edges $\{c, X_k c\}$. Let a law and a number $t \in \mathbb{N}$ be given, and let t packets of the network be given to every slot of the law, where slots that are not full get packets of zeros. The *law move* uses up these packets and applies (G.2) to each of the $t\ell$ *layers* of bits. By the convention of Section G, every channel then holds a string of $t\ell$ blocks, each of which is the code of an index followed by its bits. The string of every channel e is written on $o_e + 1$ new packets, where $o_e = \lfloor (\text{length of the string})/\ell \rfloor$, and the vector $o = (o_e)$ is called the *outcome*.

A packet that is used up leaves one copy at each of its two ends, and each of the two nodes sums over its own copy. The node tensor of the law move at an element of ω is the sum, over one index for each layer and over the bits of its channels in each layer, of the products of the node tensors of the law, restricted to those choices whose strings are the contents of the new packets.

Thus a law move replaces a network by a sum, over outcomes, of networks with $o_e + 1$ packets on the channel e, whose node tensors check the codes, and the value does not change because each choice of indices and bits belongs to exactly one outcome. We call an outcome of a law move with the number t *good* if $\|o\|_* \leq \sqrt{C_1}\, a_1 t$ and $o_e \geq t$ for all channels e. Every outcome that can occur is good, by Theorem G.2(b), the triangle inequality over the $t\ell$ layers, and $\lambda \geq 1$, so the sum may be restricted to good outcomes. The numbers of packets in the new networks depend only on the outcomes, never on the contents of packets, and all bounds below follow from such numbers.

H.3. **The decomposition theorem.** Of the h types of edges of V_j, only the packets on the least loaded one travel on to V_{j+1}, whereas all others end at a parent (Figure 11). Rerouting the former multiplies their load by a factor that does not depend on h, so by (G.1) the load, in a quadratic norm, is halved at every stage, while the number of elements falls by the constant factor b, and this gives the factor $M^{-\gamma}$. Different outcomes of the law moves give different trees. Parts (c), (d) and (e) of the theorem are needed only in Section J and can be skipped at first reading.

Theorem H.2 (decomposition into tree networks)**.** *There is a constant C with the following property. Let $M \geq 2$, let $\ell, N \geq 1$, and let a network over a commutative A-algebra, with packets of ℓ bits, have at most M nodes, no packet from a node to itself, and at most N packets at each node. Its value is the sum, over a set of choices $\mathbf{o}$, each of which is a sequence of good outcomes, of the values of networks with the following properties. Put $\Lambda = (\lg M)^C$, where $\lg M = \lfloor \log M \rfloor + 1$, and $\gamma = 1/\log b$.*

(a) *The nodes form a rooted tree of depth at most Λ, by stages: the given nodes are of stage 0, every node but the root has its parent in the next stage and at most b children, and the root has at most C children.*

(b) *A set of at most M^C named packets, the* dead *ones, may join any two nodes, and not all of them need be present. All other packets join a node and its parent. Their number is at most $\Lambda M^{1-\gamma}(N+M^2)$ for one node, its children included, and at most $\Lambda M(N+M^2)$ in all.*

(c) *The node tensor of a node is the given one (or 1) times the node tensors of the moves at it, summed over the packets that they have used up. At stage 0 these are tests only. Elsewhere they are tests and entries of node tensors of laws, and the numbers of layers of the law moves at a node add up to at most the number of bits of all packets that it ever holds.*

(d) *The parents, the packets of the new network, the packets and numbers of the moves, and which outcomes are good are computed from $\mathbf{o}$ and the given numbers of packets by a straight-line program with at most $(MN)^C$ steps on numbers below $(MN)^C$. A step is an input, a constant, or an arbitrary function of two earlier results, given by its table (a* binary operation*). The program does not depend on the numbers of packets, the outcomes or the node tensors. The numbers of nodes, of moves and of packets that can occur are at most $(MN)^C$ as well.*

(e) *The program also computes for every node p a number u_p that bounds the number of packets, other than dead ones, between p and its parent. The bounds of (b) hold with these numbers in place of those packets: u_p and the numbers u_q of the children q of p add up to at most $\Lambda M^{1-\gamma}(N+M^2)$, and all u_p together to at most $\Lambda M(N+M^2)$. The number of all packets that a node p ever holds is at most a constant times the sum of u_p, of the numbers u_q of its children q, and of the number of dead packets at p, present or not, and ΛN more if p is a given node. Here the packets that are used up at p, by relays as well as by law moves, and the zeros in slots are counted too.*

Formalization note. For part (d), and for the numbers u_p of part (e), the proof below only describes the straight-line program and accounts for its size. That the program computes exactly what is claimed is proved in the Lean development (Appendix Q), and so are the details of the count in part (e).

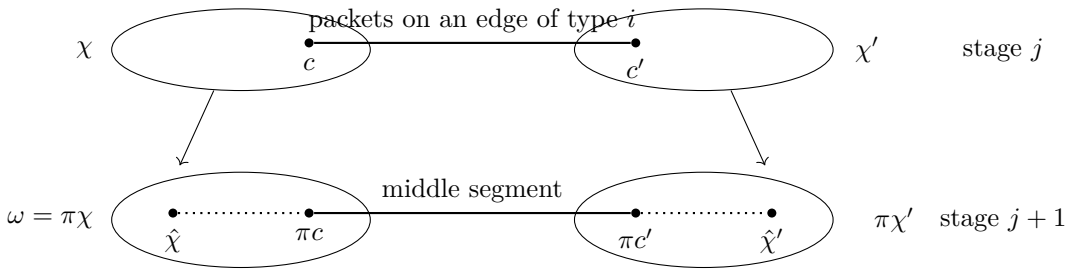


FIGURE 11. One stage. Every element of the coset χ becomes a child of $\hat{\chi}$. A law is applied to the dotted segments inside each coset of stage $j+1$.

Proof. We spread the packets over the edges of G_n, describe one stage and the recursion for the loads, and then verify (a) to (e). Take the least $n \geq 1$ with $M \leq |G_n|$, so that $|G_n| = O(M)$ by Proposition H.1(c). Nodes with node tensor 1 and no packets may be added, and bounded factors are absorbed in Λ, so we may assume that $M = |G_n|$. The nodes of the new networks will be the elements of all V_j, those of V_j forming the stage j, and a root, and we place the given nodes at the elements of $V_0 = G_n$.

Start. We divide the packets of each pair (α, β) of given nodes into M batches, of equal size up to one, one for every $g \in G_n$. We lead the batch of g by relays from α to g along the word for $g\alpha^{-1}$ and on to β along the word for βg^{-1}, and we stop at the first visit of β, as a relay needs three different nodes. The batches that leave one α for one g have together at most

$N/M + M$ packets. There are M words $g\alpha^{-1}$, of at most L letters each, and for a fixed word all its translates use each edge at most twice for each of its letters. Hence the first halves of the paths put at most $2\mathsf{L}(N + M^2)$ packets on an edge, and likewise the second halves, so that every edge gets at most $n_0 = 4\mathsf{L}(N + M^2)$ packets.

Loads. For numbers n_e of packets on the edges of V_j let

$$N_i^2 = \frac{|V_j|}{2} \sum_{e \text{ of type } i} n_e^2, \qquad Q_j^2 = \sum_{i \in \mathbb{Z}/h} N_i^2,$$

so that $Q_0 \le \sqrt{h}\, M n_0$ after the start.

One stage (Figure 11). Let i be the first type for which N_i is least, so that $N_i^2 \le Q_j^2/h$. Every element c of a coset χ of type i gets the parent $\hat{\chi}$. A packet on an edge inside χ is relayed through $\hat{\chi}$, which gives two packets between nodes and their parent. A packet on an edge cc' of type i, with cosets χ, χ', is relayed along $c, \hat{\chi}, \pi c, \pi c', \hat{\chi}', c'$, where trivial steps are omitted. The middle segments $\pi c\, \pi c'$ lie on edges of type i of V_{j+1}, and as b edges of V_j lie over one edge of V_{j+1}, the new N_i is at most the old one.

The outer segments $\hat{\chi}\, \pi c$ lie inside the cosets $\omega = \pi\chi$ of type i of V_{j+1}, and they give each ω a load n_ω on its pairs. As $\hat{\chi}$ determines χ, and πc then determines c, every pair serves at most two elements c, so that $\sum_\omega \sum_e n_{\omega,e}^2 \le 4 \sum n_e^2$, where the last sum is over the edges of type i of V_j. With $|V_j| = b|V_{j+1}|$ this gives

$$\frac{|V_{j+1}|}{b} \sum_\omega \|n_\omega\|^2 \le 4N_i^2.$$

In each ω we apply a law move to the outer segments, with the first law that has $n_\omega \le t\sigma$ for the least possible t, so that $t \le \|n_\omega\|/a_1 + 1$ by Theorem G.2(a). We do this also if $t = 0$, which produces only the end packets described next. For a good outcome o_ω the new load on the channels of ω satisfies $\|o_\omega\|_* \le \sqrt{C_1}\,(\|n_\omega\| + a_1)$. The last of the $o_e + 1$ packets on a channel is not counted here, because it is never moved again, and these end packets are the dead packets of (b). The channels of ω are its edges of the types $k \ne i$, so the new loads of these types have

$$\sum_{k \ne i} N_k^2 = \frac{|V_{j+1}|}{b} \sum_\omega \|o_\omega\|_*^2,$$

a sum over $|V_{j+1}|/b$ cosets. By Minkowski's inequality and (G.1), with the constant $C_2 = \sqrt{C_1}\, a_1/b$,

$$Q_{j+1} \le (1 + 4C_1)^{1/2} N_i + C_2 |V_{j+1}| \le \tfrac{1}{2} Q_j + C_2 |V_{j+1}|.$$

As $b \ge 4$, this gives $Q_j \le 2^{-j} \bar{Q}$ with $\bar{Q} = O(M n_0)$. After stage J we relay every packet on an edge of V_J that is not dead through the root.

Bounds. The tree has the stages $0, \dots, J$ and the root, and $J = O(\log M)$; the children of $\hat{\chi}$ are the b elements of χ, and those of the root are the elements of V_J, which gives (a). Every packet on an edge at an element becomes one packet between this element and its parent. Hence, by the Cauchy–Schwarz inequality over its h edges, an element of stage j has towards its parent at most

$$\Big(\frac{2h}{|V_j|}\Big)^{1/2} Q_j \le C \Big(\frac{\sqrt{b}}{2}\Big)^j M^{-1/2} \bar{Q}$$

packets. This is largest for $j = J$, and $2^{-J} = O(M^{-\gamma})$ since $M = O(b^J)$, which gives (b) for one node and its boundedly many children, as L is at most a power of $\lg M$. In the same way all elements of stage j together have at most $2\sqrt{h}\, Q_j$ packets towards their parents, which gives the total. There is one law move for each coset of each stage, whatever N is, so there are at most hM law moves, with boundedly many channels and hence boundedly many dead packets in each.

For (c), the elements of V_0 take part in relays only, and $t \le o_e$ for every channel e at a node, so the number of layers that a node tensor handles is bounded by the node's own bits. For (e), let u_p be the sum of the loads on the edges at p, and 0 for the root, so that the bounds just shown are bounds for these numbers. A relay puts two packets at its middle node and replaces a packet by one packet at each of its ends. In a stage a packet is relayed boundedly often, and the slots at an element take boundedly many times $t \le o_e$ packets, zeros included. Hence what a node ever holds in the stages is at most a constant times the loads at it and at its children, and its dead packets. In the start the end β of a given packet holds one packet after another, at most $2\mathsf{L}$ of them, and this is the term ΛN at a given node. The copies that a node sums over are among these packets.

For (d), and for computing the numbers u_p, the groups, the words, the cosets and the maps $\chi \mapsto \hat{\chi}$ are constants of the program, and all loads are $(MN)^{O(1)}$. Finding the least N_i and the least t, and testing whether an outcome is good, amounts to comparing integers of that size, because C_0' is an integer. Packets are assigned to batches, slots and channels by their number in a fixed order. There are fewer than $2M$ nodes besides the root and at most hM law moves. In the start each of the at most MN given packets is relayed at most $2\mathsf{L}$ times. Afterwards every packet on an edge is relayed boundedly many times in its stage, there are $O(\log M)$ stages with at most hM edges each, and each edge carries $(MN)^{O(1)}$ packets by the bounds for the loads. Hence the number of relays is $(MN)^{O(1)}$. So is the number of packets: a packet is a given one, or is made by a relay, or is put by a law move on a channel, which is an edge of the next stage, or is one of the packets of zeros of a law move, of which there are boundedly many times $t \le o_e$. □

Appendix I. Programs, automata and networks

To apply Theorem H.2 to computations, we fix in this section what a computation is and show that it is a network. Programs on binary trees (Section I.1) can run other programs, as Section K requires, and the values of tape automata (Section I.2) are sums of values of networks (Lemma I.2). Proposition I.1 connects the two models, and its proof, which follows classical ideas, can be skipped at first reading. We keep the ring A and the laws of Section G, and $\lg v = \lfloor \log v \rfloor + 1$ for an integer $v \ge 1$.

I.1. **Programs and sums over runs.** All data are binary trees, where a tree is either nil or a pair of two trees, and the *size* $|x|$ of a tree x is its number of nodes. A *program* is an expression in the constructs of the following table, in which p_1, p_2, p_3 stand for subprograms and x for the argument. Programs are nondeterministic, and a run carries a weight in A, which is 1 at the start, and a *tally*, which is a list of names.

construct	output, and effect on the run
a fixed tree; the argument	that tree; x
first, second part of p_1	that part of the output of p_1, and nil for nil
pair of p_1, p_2	the pair of the two outputs
if p_1 then p_2 else p_3	the output of p_2 if that of p_1 is not nil, otherwise that of p_3
`run` p_1 p_2	the output of the program coded by the output of p_1 at the output of p_2
`guess`; `fail`	two runs, with outputs nil and (nil, nil); no accepting run
`weight` p_1	nil; multiplies the weight by the generator of A named by the output of p_1
`use` p_1	nil; appends the output of p_1, a name, to the tally

Subprograms are evaluated from left to right, their weights multiply and their tallies are concatenated. Every construct costs one step, and the *cost* of a run is its number of steps. Finitely many generators of A, one of which is -1, have names, and every other name has the weight 0. A run is *accepting* if it does not fail, and its output is then the output of the program.

A program is coded as the pair of the number of its outermost construct, in unary, and the codes of its parts. Conversely every tree is read as a program, with nil read as the fixed

tree nil and a pair whose first component is not the number of a construct as `fail`. A loop is a program that receives its own code in its argument and calls `run` on it.

An *instance* z is a pair of a program and an argument, and the tree nil, of size 1, also counts as one, with program and argument nil. That *all runs* of an instance *halt* is defined inductively along the evaluation: in a pair and in `run` both subprograms must halt, also when the first has no accepting run, and a chosen branch, and the program that is run, must halt for every accepting run of what precedes them.

Definition. An instance z is *τ-plain* if all its runs halt and every accepting run has an empty tally and costs at most τ. The *sum* $\mathrm{sum}(z) \in A$ of an instance is the sum of the weights of its accepting runs.

Any machine model with a self-interpreter of constant overhead should do, and polylogarithmic factors are always absorbed by powers of the parameter v of Sections J and K. The construct `use` and the tally are needed only in Section K, where a program that simulates a circuit over $\mathbb{R}$ cannot contain the real constants of the circuit and names them in the tally instead (Definition K.1). Finally, the *code* of an instance *as a word*, or briefly its *word*, lists one symbol for each node of the tree, in preorder.

I.2. **Tape automata.** The second model is a weighted finite automaton that also writes on tapes.

Definition. A *tape automaton* over A has finite sets of states, of tapes and of symbols, two fixed states q^- and q^+, and for each *letter* a matrix over A whose rows and columns are indexed by the states. Each tape has two sides. A letter is either *idle*, or it writes one symbol on one of the two sides of a tape, or it writes one symbol on the input. A sequence of t letters is *matched* on an input word if on every tape the two sides receive the same word and the input receives the given word. The *value* of the automaton on an input word in t steps is the sum, over all sequences of t letters that are matched on this word, of the entry (q^-, q^+) of the product of their matrices.

One side of a tape is for writing and the other for reading: a letter on the reading side guesses the symbol that is read, and being matched forces the guesses to agree with what is written. So a tape is like a queue, except that a symbol may be read before it is written. One tape automaton, the *universal automaton*, evaluates every plain instance with a small loss of time.

Proposition I.1 (a universal tape automaton)**.** *There are a tape automaton and a constant C_{univ} such that the value of the automaton on the code of an instance z as a word, in t steps, is* $\mathrm{sum}(z)$ *whenever z is τ-plain and $t \geq C_{\mathrm{univ}}(|z| + \tau)\big(\lg(|z| + \tau)\big)^2$.*

Formalization note. The proof describes the automaton and counts its steps, and that the automaton is exact is proved only in the Lean development (Appendix Q).

Proof. We describe the construction, which follows classical ideas [Sch78, Sch80, BEGKN94, Rob91]: a pointer machine evaluates the program, and a tape automaton simulates it by guessing what every load returns and checking the guesses at the end.

The pointer machine. The machine has a finite control and a fixed number of registers that hold addresses, where the address 0 stands for nil. In one step it may load the pair at the address in a register, allocate a pair of two registers at the next free address, test a register for 0, guess a bit, or multiply by one of finitely many weights. It first reads the code of z, which is the list of the brackets of the tree in preorder, and builds the tree, keeping a stack of pairs in the heap. Then it evaluates the program on the argument, with a stack of frames (tag, program, argument, outputs found so far) in the heap. On `use` it fails, since a plain instance uses no name in an accepting run. The number of a construct is found by at most 11 loads, every construct takes boundedly many steps, and the name of a generator is read

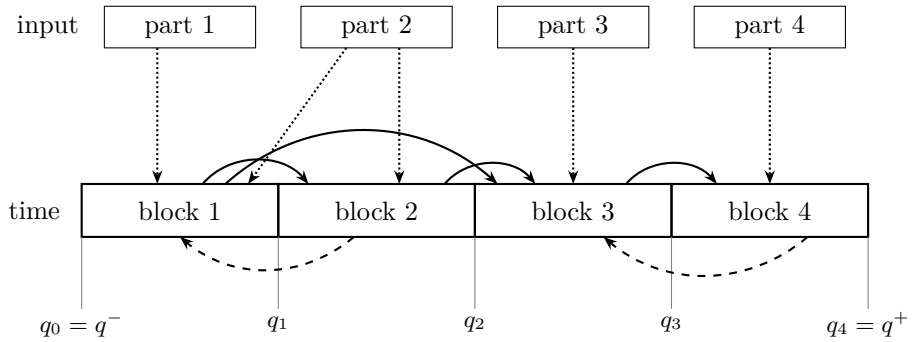


FIGURE 12. A run of a tape automaton over KT steps as a network, for $K = 4$. An arrow from p to p' stands for the packets with the symbols that p writes and p' reads, solid and dashed for two tapes and dotted for the input. Dashed arrows point backwards in time.

only to the bounded depth beyond which there are no names. Hence an accepting run of z of cost n gives exactly one accepting run of the machine, with the same weight and $O(|z| + n)$ steps, and there are no others.

The automaton. The automaton guesses the number β of bits of an address and writes it in unary on a tape that serves as a counter. The registers and the next free address are words of β bits that circulate on one tape: in each step of the machine they are read and written again, and the free address is increased on the way by the carry rule, where a carry out of the β bits fails. For each load it guesses the pair and writes the record (address, load, pair), and for each allocation, the input tree included, it writes (address, allocation, pair). When the machine has accepted, it checks that the leading bit of the free address is 1, which determines β, and it sorts the records by the key (address, allocations first). A pass of the sort reads the list and writes each record on one of two tapes according to the first bit of its key, with that bit moved to the end of the key, and then copies the first tape and after it the second back to the list. The counter counts the $\beta + 1$ passes, so the same tapes serve for all of them. As the key is written with its least significant bit first, after $\beta + 1$ passes it is in its old order and the sort, which is stable, is complete. A last pass checks that each address begins with an allocation and that all loads of it repeat its pair. After that the automaton stays in its final state on idle letters, and this is the only place where idle letters are allowed.

Exactness and number of steps. Both sides of a tape carry the same word, and each block on a tape ends with a mark and is written before it is read, so what is read is what was written, in the same order. By induction on time all addresses in registers are below the free one, so a load refers to the only allocation at its address, which is earlier. Hence the last pass forces every guessed load, β is forced, and all other letters are determined by the guesses of the machine. Thus accepting runs correspond to matched sequences of length t, provided that t is large enough. They need $O(\beta)$ letters for each of the $O(|z| + \tau)$ steps and records in each of the $\beta + 1$ passes, and $\beta = O(\lg(|z| + \tau))$. This is the classical conversion of random access into sequential checking [GS89]. □

I.3. **The runs of a tape automaton as a network.** Cutting a computation into time blocks and recording which block reads what another has written goes back to Hopcroft, Paul and Valiant [HPV77] (see also [PPST83, Wil25]). For a tape automaton the result is a network in the sense of Section H.2, whose nodes are the time blocks and the parts of the input word (Figure 12). As a tape is a queue, the readers of what a node writes follow each other in order, and the data ρ below record the first and the last.

Lemma I.2 (the runs of a tape automaton as networks)**.** *Let $K, T, \ell \geq 1$ and $T' \leq T$. Let an input word of at most KT' symbols be cut into K consecutive* parts *of T' symbols, the last parts possibly shorter or empty, and let the KT steps be cut into K consecutive* time

blocks *of T steps. Order the $2K$ nodes as follows: first the K time blocks, then the K parts. Regard the input as one more tape, which is written by the parts and read by the blocks. Let S be the set of tapes and Q the set of states, and let the symbols and an end mark be coded by c bits each.*

- Data ρ *say, for every tape and every node, whether the node writes nothing on the tape, and otherwise which is the first and which is the last node that reads its symbols.*
- *The data ρ are* valid *if on every tape the first reader of a node is not before the last reader of the node before it, and not after the last reader of the node itself. Here a node that writes nothing repeats the last reader of the node before it, so that ρ is determined by the sequence of letters, and for the first node, the first node itself takes the place of that last reader.*

Then the value of the automaton in KT steps is the sum of the values of networks on these $2K$ nodes, one network for every choice of

- *boundary states $q \in Q^{K+1}$ with $q_0 = q^-$ and $q_K = q^+$,*
- *valid data ρ, and*
- *numbers of packets of ℓ bits between any two nodes, each number at most $(T + |S| + 1)c/\ell + 1$.*

In each of these networks a node whose node tensor is not 0 has at most $2\big((T + 2K(|S| + 1))c/\ell + 2K\big)$ packets.

Proof. We describe the network of one choice and then compare its terms with the matched sequences. For an ordered pair (p, p') of nodes, the symbols that p writes and p' reads on all $|S|+1$ tapes, those of each tape closed by the end mark, form one string, which is written on the packets from p to p'. On a tape, two nodes are *partners* if one reads a symbol that the other writes. The node tensor of a part tests that the contents of its packets are its symbols. The node tensor of the ith block is the sum, over all sequences of T letters that have the partners prescribed by ρ and whose symbols, coded on packets, are the given contents, of the entry (q_{i-1}, q_i) of the product of their matrices. It vanishes unless the block writes at most T symbols in all, which bounds its packets.

In a matched sequence the two sides of a tape carry the same word, so the jth symbol of one side meets the jth symbol of the other. Hence the partners of a node on a side follow each other in order, from a first to a last, and every matched sequence belongs to one choice of a valid ρ, of the boundary states and of the numbers of packets.

Conversely, let ρ be valid, and on some tape let $w_{pp'}$ be the symbols that p writes for p', in a term that is not 0. If $w_{pp'}$ and $w_{rr'}$ are not empty and $p < r$, then $p' \le r'$, because ρ is valid. So the $w_{pp'}$ give the same word in the order of (p, p'), which is the writing side, and in that of (p', p), the reading side, and the sequence is matched. □

Appendix J. Cutting a computation into pieces

Combining Theorem H.2 with Lemma I.2, we prove in this section that a long computation can be cut into many short and almost independent ones, the *pieces*, and that a fast program lists them (Theorem J.4). All quantities are measured in powers of a large integer v, which we call the *scale*. Proposition J.2 is the combinatorial part, Theorem J.3 turns it into a program that makes one cut, and Theorem J.4 repeats such cuts until the pieces take v^H steps, for a constant H. Table 2 lists the powers of v used from here to the end of Section K; the exponents are not optimized, and apart from the few inequalities noted in the table only their order matters.

Convention. Throughout this section Λ denotes a power of $\lg v$ whose exponent may grow a bounded number of times. We write C_{tree} for the constant C of Theorem H.2, and C_{univ} is the constant of Proposition I.1.

power	where it occurs	why it suffices
v^{300}	the cut packets of a piece have at most $KT\Lambda/v^{300}$ bits (Proposition J.2)	more than 256, so the next line absorbs these bits
v^{256}	one cut adds τ/v^{256} to the inputs and $\tau k/v^{256}$ to the time of cutting (Theorem J.3)	$48 + 64 \le 256$, so inputs stay shorter than times by v^{48}
v^{64}	largest number k for one cut	less than 300, so a single node of the tree is small enough for a piece
v^{63}	k for every cut but the last	the time stays at least v^{H+1}
τ/v^{48}	inputs of a cut	exponent vectors and programs of a round fit (Proposition K.5)
τ/v^{24}	time of listing the pieces	less than what a round saves
v^{H}	time of the sums to be evaluated	a multiple of 16 with $H \ge 272$ and $H \ge a_0{+}16$
v^{H-32}	their inputs	programs of size $v^{H/4}$ fit
v^{H-16}	time of an accelerator (Definition K.1)	a round saves v^8, with room for logarithms

TABLE 2. The exponents in Sections J and K, for a computation of τ steps.

J.1. **Programs that cut.**

Definition J.1. A program G *cuts* (τ, s) into (τ', s') with P pieces in t steps if the following holds for every instance z with $|z| \le s$.

- All runs of G on z halt. The accepting ones take at most t steps, have empty tallies, and output lists of at most P instances of size at most s'.
- If z is τ-plain, then all these instances are τ'-plain, and

$$\operatorname{sum}(z) = \sum_{\text{runs}} \text{weight} \cdot \prod_{y \text{ in the list}} \operatorname{sum}(y),$$

  where the sum is over the accepting runs of G on z.

Thus every accepting run of G outputs a list of instances, the *pieces* of z. The first condition is required of every instance of size at most s, plain or not, which Proposition K.5 uses.

J.2. **The pieces of a tree network.** By Lemma I.2 and Theorem H.2 the value of a tape automaton in KT steps is a sum of values of tree networks. The next proposition cuts each tree into subtrees, the *pieces*, each with about the fraction $1/k$ of the nodes and of the packets, for a prescribed k. Once the packets between different pieces are fixed, the sum over all others is a product of one factor for each piece, its *value*.

Proposition J.2 (the pieces of a computation)**.** *For every tape automaton over A there are constants a, a_K and C, where $a_K \ge 300$ and where a and a_K do not depend on the automaton, with the following property. For all large v there is K with $v^{a_K} \le K \le Cv^{a_K}$ such that the statements below hold for all $T \ge v^a$ with $\lg T \le (\lg v)^2$ and all $2 \le k \le v^{64}$.*

Cut an input word of at most KT' symbols, where $T' \le T$, into K consecutive parts of T' symbols, the last parts possibly shorter or empty.

- *The value of the automaton in KT steps is the sum, over boundary states, valid data ρ as in Lemma I.2, numbers of packets, good outcomes, and the contents of* cut packets*, of products of the values of at most $k\Lambda$* pieces*.*
- *A piece has at most CK/k blocks and parts, at most $KT\Lambda/v^{300}$ bits in cut packets, and at most $KT\Lambda/k$ bits in all.*
- *The packets, the pieces and what each node tensor consists of are computed from the numbers of packets and the outcomes by v^C binary operations on numbers below v^C.*

Formalization note. The last claim rests on Theorem H.2(d) and (e), whose printed proofs only describe the program in question.

Proof. We apply Lemma I.2 and then Theorem H.2, and cut each tree into subtrees of about equal size. Let $K = v^{a_K}$, with $a_K \geq 300$ so large that $M^{-\gamma} \leq v^{-300}$ for $M = 2K$, so that every polylogarithm of M and of T is at most Λ. Let packets have $\ell = \lfloor T/v^{a'} \rfloor$ bits, with a constant $a' < a$ so large that all $M^{C_{\text{tree}}}$ dead packets together have at most T/v^{300} bits, and that $N = v^{a'}\Lambda$, a bound for the packets at a node, exceeds M^2. Apply Lemma I.2, keep only numbers that give every node at most N packets, and sum every packet from a node to itself into the node tensor of that node, which the evaluator below treats as an inner packet of the piece of that node. Then apply Theorem H.2.

Heavy nodes and pieces. Let Y be the bound for the total in Theorem H.2(b), and let the given nodes be the blocks and parts. A node is *below* the nodes in whose subtrees it lies, itself included, and the *packets below* a node are those between a node below it and the parent of that node, counted by the numbers u_p of Theorem H.2(e). Call a node of the tree *heavy* if it is the root or if

$$\frac{\#(\text{given nodes below})}{4M} + \frac{\#(\text{packets below})}{2Y} > \frac{1}{k},$$

and *light* otherwise. The sets below different nodes of one stage are disjoint, so over a stage the left sides add up to at most $\frac{1}{4} + \frac{1}{2}$, and each stage has fewer than k heavy nodes. Parents of heavy nodes are heavy. A *piece* is a heavy node, or the subtree of a light node with a heavy parent, so there are at most $k\Lambda$ pieces. The packets between different pieces are the *cut packets*, and the value of a piece sums over its other packets, the *inner* ones. Every assignment of packets and letters is counted once, since the contents of absent or padded places are forced to be zero.

The bounds. By part (b) of the theorem a piece has at most $\Lambda M^{1-\gamma} N\ell \leq KT\Lambda/v^{300}$ bits in cut packets, besides the dead ones. A subtree that is a piece has at most $4M/k$ given nodes and $2Y/k$ packets, because its root is light, and a single node has fewer still, as $k \leq v^{64}$. Packets used up and zeros in slots change this by a constant factor only, apart from ΛN at each given node, by Theorem H.2(e). The packets from a node to itself are at most N at each of the at most $4M/k$ given nodes, and the dead packets have at most T/v^{300} bits. This bounds the bits in all. Parts (d) and (e) of the theorem give the last claim, since the heavy nodes and the pieces are computed from the parents and the numbers u_p. □

J.3. **One cut.** The next theorem turns Proposition J.2 into a program. Each piece becomes an instance again, whose program is a fixed *evaluator* and whose argument is a *script* that describes the piece.

Theorem J.3 (one cut)**.** *There are constants a_0, C, v_0 such that the following holds with $\Lambda = (\lg v)^C$, for all $v \geq v_0$, all $\tau \geq v^{a_0}$ with $4\lg\tau \leq (\lg v)^2$, all $s \leq \tau$ and all $2 \leq k \leq v^{64}$. Some program of size at most v^{a_0} cuts (τ, s) into*

$$(\tau\Lambda/k,\ (s/k + \tau/v^{256})\Lambda)$$

with at most $k\Lambda$ pieces, in $(s + \tau k/v^{256})\Lambda$ steps.

Formalization note. The proof describes the evaluator and the cutting program G but does not print them, and that they are exact and keep the stated bounds is proved only in the Lean development (Appendix Q).

Thus a cut by k divides the time and the input by k, up to the factor Λ, and the term τ/v^{256} accounts for the cut packets and the description of the tree, which the input of a piece also contains.

Proof. We obtain a formula for $\text{sum}(z)$ from Proposition J.2 and then describe the two programs. Let K, a be as in that proposition for the universal automaton, M, N, ℓ as in its proof, and $T = \lceil \tau\Lambda/K \rceil$. For a τ-plain z with $|z| \leq s \leq \tau$ we have $KT \geq C_{\text{univ}}(|z| + \tau)(\lg(|z| + \tau))^2$, so $\text{sum}(z)$ is the value of the universal automaton on the word of z in KT steps. If a_0 is large, then $v^a \leq T \leq \tau$, and K, M and all tables of size $(MN)^{C_{\text{tree}}}$, those

of the tower among them, are smaller than v^{a_0} and than τ/v^{256}. The proposition therefore gives the formula for $\mathrm{sum}(z)$. A piece reads at most $(s/k + M/k)\Lambda$ symbols of the input, and $M \le \tau/v^{256}$.

The evaluator. Call the tree, the packets and the pieces of one choice its *plan*. We name its packets, and the places of letters in blocks, by strings of $O((\lg v)^2)$ bits, and a binary trie finds an object in that many steps. A *script* of a piece contains the numbers T and ℓ in binary, the contents of its cut packets, the names of its inner packets, and a list of instructions of three kinds.

- BLOCK: the boundary states of a block, its entries of ρ, the numbers of packets, and the names of the packets that carry them (or, for the input, the symbols themselves).
- ROLE, which is the part of one node in one law move: the law, the number t, the outcome, and for every slot and channel at this node the names of the packets that supply or receive its bits. A slot may be supplied by zeros. The bits of a segment that a law has replaced are those of the packet on the other side of the test at the node.
- TEST: two packets are equal, or one packet is zero.

The evaluator guesses the contents of all inner packets and stores them. For BLOCK it guesses T letters and the states between them, multiplies for each by the matrix entry, collects the symbols of every side, checks that they fit these partners and numbers, and compares them, coded on packets, with the stored contents. For ROLE it guesses the index of each of the $t\ell$ layers and the bits of its channels, reads the bits of the slots in order, multiplies by the entry of the node tensor, and compares the coded strings with the stored packets. An entry of a matrix or of a node tensor is an element of A, hence a fixed sum of products of the generators, so multiplying by it is guessing a summand and multiplying by its factors. The contents fix the symbols and the indices, since codes are injective, and what is still summed over is exactly what the node tensors of Lemma I.2 sum over. Hence the sum of the instance (evaluator, script) is the value of the piece. All its loops have lengths that stand in the script, so all its runs halt, and within

$$(\#\text{blocks} \cdot T + \#\text{bits of packets} + \#\text{instructions} \cdot v^{O(1)})(\lg v)^{O(1)} \le \tau\Lambda/k$$

steps, because a_0 is large and because $t \le o_e$, which holds for every channel e of a good outcome o (Section H.2), bounds the layers by the bits of the packets.

The program G. The program contains the tables of the tower. On z it writes the word of z and cuts it into K parts of equal length, the last ones shorter. It guesses q, ρ, the numbers of packets and the outcomes, each in binary with a fixed number of digits, and it rejects unless ρ is valid, the numbers give every node at most N packets, and all outcomes are good. With these it computes by the binary operations, in $(MN)^{C_{\mathrm{tree}}}\Lambda$ steps, the tree, the packets, the heavy nodes and the pieces, where rules such as ranks in a fixed enumeration decide which packet goes where. It then guesses the contents of the cut packets, writes for each piece its script, and outputs the list of the instances (evaluator, script). All of this is deterministic except for the guesses, and every choice in the formula is guessed exactly once, so the sum formula is the one obtained above.

A script has $(s/k + M/k)\Lambda$ input symbols, $\tau\Lambda/v^{300}$ bits of cut packets, and $v^{O(1)}$ names and instructions, which is at most $(s/k + \tau/v^{256})\Lambda$ in all. The program G takes $s\Lambda$ steps for the word, $(MN)^{C_{\mathrm{tree}}}\Lambda \le \tau\Lambda/v^{256}$ steps for the plan, and $k\Lambda \cdot \tau\Lambda/v^{300}$ steps for the cut packets, which gives the stated size of the pieces and number of steps. □

J.4. **Cuts at all scales.** We now repeat the cut of Theorem J.3 until the pieces take v^H steps, and the point to watch is that the inputs remain shorter than the times by the factor v^{48}, up to a factor 4 for each cut.

Theorem J.4 (cuts at all scales)**.** *There are a multiple $H \ge 272$ of 16 and constants C_{cut}, v_0 such that the following holds whenever $v \ge v_0$ and α satisfy $(\lg v)^{C_{\mathrm{cut}}\alpha} \le v$. For all τ and*

s with $v^H \le \tau \le v^{H\alpha}$ and $sv^{48} \le \tau$, some program of size at most v^H cuts (τ, s) into (v^H, v^{H-32}) with at most $\tau v^2/v^H$ pieces in at most τ/v^{24} steps.

Proof. We compose cuts of Theorem J.3 and follow their parameters by induction. Let H be a multiple of 16 with $H \ge 272$ and $H \ge a_0 + 16$. We cut by $k = v^{63}$ as long as $\tau\Lambda \ge v^{H+64}$, and at last once by $k = \lfloor \tau\Lambda/v^H \rfloor + 1$, where τ is the current bound on the time, and we compose the programs, so that the next one is run on every piece in the list. The condition on v gives $4 \lg \tau \le (\lg v)^2$, and there are $r = O(\alpha)$ cuts.

Let (τ_i, s_i) be the parameters after i cuts and P_i the number of pieces, and put $\tau_r = v^H$ after the last cut, for which $k \le 2\tau_{r-1}\Lambda/v^H$. Three bounds hold by induction on i, starting from $sv^{48} \le \tau$. First, $s_i v^{48} \le 4^i \tau_i$, because $k \le v^{64}$ and $48 + 64 \le 256$, so that Theorem J.3 applies again. Second, $P_i\tau_i \le \tau(4\Lambda^2)^i$. Third, the next cut takes at most $(s_i + \tau_i k/v^{256})\Lambda \le 2 \cdot 4^i \tau_i \Lambda/v^{48}$ steps on each piece. Moreover, the sum formulas compose.

It follows that the final pieces have size at most $v^{H-48}(8\Lambda)^r$ and that there are at most $(\tau/v^H)(4\Lambda^2)^r$ of them. The number of steps is at most $(\tau/v^{48})(100\Lambda^3)^r$, and the program has size at most $4^r(v^{a_0} + O(1))$. All these powers of Λ are below v if C_{cut} is large, which gives the bounds of the theorem. □

Remark. For a fixed α the condition $(\lg v)^{C_{\text{cut}}\alpha} \le v$ holds for all large v, and this is all that Theorem L.2 needs.

Appendix K. Small circuits would speed up every computation

We are still in the contracting case, and we keep the ring A of Theorem G.2 and the constant H of Theorem J.4, and put $h_0 = H/16$. In this section we prove Theorem K.7, the lower bound for the permanent in the contracting case. A small circuit for the permanent would give a program, called an accelerator, that evaluates sums over the runs of other programs in fewer steps than those runs take, and we show that accelerators do not exist.

K.1. **Accelerators.** A circuit for PERM_m with m^C gates is advice that is longer than the running times at its own scale, so we cannot diagonalize against it there. By Theorem J.4, however, a device that speeds up sums over runs of v^H steps also speeds up much longer sums, piece by piece. The same advice therefore serves at the scale $v' = v^\alpha$, where it is short (about $v'^{H/4}$, against inputs of length v'^{H-32}) and a diagonal argument applies.

A run of a program never multiplies by the constants of a circuit but only *names* them in its tally (Section I). The powers ξ^{2^i} of a constant ξ also get names, so that a program can multiply by ξ^e by naming ξ^{2^i} for the binary digits of e that are 1.

Definition K.1. Let R be a commutative A-algebra. An *accelerator* at scale v with constants $\xi_1, \xi_2, \dots$ from R and exponent bound B is a program D with the following two properties.

(i) On every instance z with $|z| \le v^{H-32}$ all runs of D halt, the accepting runs halt within v^{H-16} steps, and every name in the tally of an accepting run stands for a power $\xi_l^{2^i}$ with $i < B$.

(ii) If moreover z is v^H-plain, then the *value* of D on z is $\text{sum}(z)$, where the value is the sum, over the accepting runs, of the weight of the run times the powers that its tally names, each taken as often as it is named.

The constants sit at the leaves of a binary trie with Q nodes, and names are paths: the name of $\xi_l^{2^i}$ is the path from the root to the leaf of ξ_l, followed by 1^i0. The accelerator is *of size s* if Q and the size of D are at most s and, with $\beta = \lg(\max(v^H, s))$, the height of the trie is at most β^2 and $B \le \beta^3$.

An instance of size at most v^{H-32} is called *short*, and Table 2 explains the margins between the exponents H, $H - 16$ and $H - 32$.

K.2. **Accelerators from small circuits.** The first lemma evaluates the part of degree at most d of a polynomial with a small circuit in about $d + |z|$ steps, up to a polylogarithmic factor. The polynomials have one variable for every pair (place, symbol), where an end mark counts as a symbol. The *indicator assignment* of a word of fewer than n symbols sets to 1 the variable (place, symbol) for each of its symbols and the variable of the end mark at the place after the last symbol, and it sets all other variables to 0.

Lemma K.2 (evaluation of the part of low degree)**.** *Let R be a commutative A-algebra, let $d \geq 0$, and let a polynomial over R in the variables (place, symbol), with n places, have a circuit with L gates. There is a program of size $O(\log L + \log n + d)$, with $O((L+n)^3)$ constants, which on every instance z with $|z| < n$ has as value, in $(d + |z| + 1)(\log(L + n + d))^{O(1)}$ steps, the sum of the homogeneous parts of degree at most d of the polynomial, evaluated at the indicator assignment of the word of z. All its runs halt, it names only constants, not powers, and the trie that holds the constants has height $O(\log L + \log n)$.*

Proof. We unfold a recursion for the homogeneous parts at the gates whose coefficients do not depend on the degree. Let $g^{(i)}$ be the homogeneous part of degree i of the polynomial at the gate g. For $i \geq 1$ we have

$$(g' + g'')^{(i)} = g'^{(i)} + g''^{(i)}, \qquad (g'g'')^{(i)} = g'^{(0)}g''^{(i)} + g''^{(0)}g'^{(i)} + \sum_{0<j<i} g'^{(j)}g''^{(i-j)}.$$

A *chain* goes from a gate to one of the two gates below it (in two ways if these are the same gate), and so on. A step has the factor 1 at a sum gate, and at a product gate the part of degree zero of the other gate below. Let $c_{g,\gamma} \in R$ be the sum, over all chains from g to γ, of the products of their factors, with $c_{g,g} = 1$. Let $a_{g,l}$ be the sum of the $c_{g,\gamma}$ over the input gates γ with the variable x_l, and $b_{g,g',g''}$ that over the product gates $\gamma = g'g''$. By induction on g we obtain

$$g^{(1)} = \sum_l a_{g,l}x_l, \qquad g^{(i)} = \sum_{g',g''} b_{g,g',g''} \sum_{0<j<i} g'^{(j)}g''^{(i-j)} \quad (i \geq 2).$$

These constants do not depend on i, and as they are indexed by a gate, a place and a symbol, or by three gates, names of $O(\log L + \log n)$ bits suffice. One more constant is the part of degree zero of the output gate.

The program writes the word of z as a balanced tree, so that the symbol at a place can be read in $O(\log n)$ steps. If $d = 0$, it names the last constant and accepts. Otherwise it guesses a bit, and for one value it again names that constant and accepts. For the other value it keeps a stack of gates, at first the output gate only, and a counter, at first $d - 1$ in unary. In each round it takes a gate g from the stack and guesses either a variable x_l, in which case it names $a_{g,l}$ and rejects if $x_l = 0$ at the indicator assignment, or, if the counter is not 0, two gates g', g'', in which case it names $b_{g,g',g''}$, puts both gates on the stack and decreases the counter. It accepts when the stack is empty. Every round removes a gate for good or decreases the counter, so all runs halt after at most $2d - 1$ rounds of $(\log(L + n))^{O(1)}$ steps. The accepting runs are, in preorder, the parse trees with at most d leaves [JS82, MP08] (compare [VSBR83, Vin91, AJMV98]), and the products of the constants that they name sum to $\sum_{1 \leq i \leq d} g^{(i)}$ at the indicator assignment, where g is the output gate. □

We apply Lemma K.2 to a polynomial Φ whose values at indicator assignments are the sums $\mathrm{sum}(z)$ of the short plain instances, and which is a clause sum, so that by Corollary E.2 it has a small circuit if the permanent has one.

Lemma K.3 (accelerators from circuits for the permanent)**.** *There are constants B_0, C, v_0 such that the following holds for all integers $a \geq 0$ and $v \geq v_0$ with $(\lg v)^{Ca} \leq v$. Let R be a field that is an A-algebra and in which $2 \neq 0$. If, for some $m \geq v^{B_0}$, the polynomial PERM_m has a circuit over R with at most v^a gates, then R has an accelerator of size $v^{C(a+1)}$ at scale v.*

Proof. Let $n = v^{H-32} + 1$, so that the word of a short instance has fewer than n symbols. Let Φ be the sum, over all words w of fewer than n symbols, of the value of the universal automaton on w in $t = v^{H+1}$ steps times the product of the variables that are 1 at the indicator assignment of w. It has degree at most n, and at the indicator assignment of a word it takes the value of the automaton on that word, because no indicator assignment contains another.

It is a clause sum, whose bits are the indicator assignment of w, for every step a one-hot code of the state and of the letter, for every step and every side of a tape a one-hot counter of the symbols written so far, and for every tape and place a one-hot code of a symbol. Clauses of weight 0 enforce one-hotness, the first and the last state, and the single end mark of w. They also forbid a wrong count, a symbol written at count i that differs from the symbol at place i, different final counts on the two sides of a tape, and symbols beyond the final count. The other clauses carry the entries of the matrices of the automaton and the variables. So there is exactly one assignment for each matched sequence and each sequence of states. The clause sum has $v^{O(1)}$ literals, and we choose B_0 so that v^{B_0} is at least three times their number. By Corollary E.2 a circuit for PERM_m with at most v^a gates then gives one for Φ with $O(v^a)$ gates.

The word of an instance uses only two symbols, so we set the variables of all other symbols, the end mark excepted, to 0. By Proposition I.1 the polynomial Φ then takes the value $\mathrm{sum}(z)$ at the indicator assignment of the word of every short v^H-plain instance z. As n bounds the degree of Φ, Lemma K.2 with $d = n$ turns the circuit into a program with $v^{O(a+1)}$ constants that has these values and takes $v^{H-32}((a+1)\lg v)^{O(1)} \le v^{H-16}$ steps. All its runs halt, its names have $O((a+1)\lg v)$ bits, and $B = 1$, so it is an accelerator of size $v^{C(a+1)}$ if C is large. The proof uses of the permanent only that Corollary E.2 holds for it, so the lemma holds for every family that is complete for grid sums (Appendix M), with constants that depend on the factor of the family, which takes the place of 3 in Corollary E.2. □

K.3. **Transferring accelerators between scales.** The tool for passing to larger scales is Theorem J.4, but it applies only to plain instances, whose accepting runs name no constants, whereas the programs to be sped up contain the accelerator. The following lemma therefore collects the constants in front of the sum over the runs.

Lemma K.4 (postponing the constants)**.** *Let p be a program and let an argument be given on which all runs of p halt, the accepting ones within $T < 2^J$ steps and with names among the leaves of a binary trie with Q' nodes and height η. Let the leaves of the trie stand for powers $\lambda_1, \lambda_2, \dots$ Then the value of p on the argument is*

$$\sum_e \lambda^e \,\mathrm{sum}(z_e), \qquad \lambda^e = \prod_j \lambda_j^{e_j},$$

where e runs over the vectors with one entry in $\{0, \dots, 2^J - 1\}$ for every leaf, and the instance z_e accepts exactly the runs of p whose tally uses each name the prescribed number of times. The instance z_e is plain within $O(T(J + \eta + 1) + Q'(J + 1))$ steps, and its size is that of p and the argument plus $O(Q'(J + 1))$.

Proof. We take $z_e = (I, (p, \mathrm{argument}, e))$, where I is a fixed program, the *interpreter*. It evaluates p construct by construct, also inside `run`, with constant overhead, and it carries e as a binary trie of counters indexed by the names. At a `use` it follows the name and decreases that counter, and it fails if there is none or if it is 0, which takes $O(J + \eta + 1)$ steps. The interpreter accepts if p accepts and all counters are 0 at the end. Hence it halts on every run and has empty tallies. An accepting run of p is accepted for exactly one e, because the numbers of times that it uses the names are at most $T < 2^J$. □

The next proposition is stated with explicit bounds, so that it can be applied repeatedly.

Proposition K.5 (transfer to the next scale)**.** *Let v be large, let $\alpha \ge 5$ be an integer, and suppose that $(\lg v)^{C\alpha} \le v$, where C is a suitable constant that depends on the fixed power*

below. Let D be a program with the two properties of an accelerator at scale v (Definition K.1), whose trie has $Q \le v^{h_0\alpha}$ nodes and whose size is at most $v^{h_0\alpha}$. Suppose that its exponent bound B and the height of its trie are at most a fixed power of $\alpha \lg v$. Then there is a program with the two properties of an accelerator at scale $v' = v^\alpha$, with the same constants, with exponent bound $B + 51h_0\alpha^2 \lg v$ and with program size at most v'^{4h_0}. Its accepting runs take at most v'^{H-17} steps. For the new program we claim only this, and not that it is an accelerator of a given size in the sense of Definition K.1, which would require the bounds of that definition on the height of the trie and on the exponent bound.

Proof. We speed up the universal program `run` in rounds, each of which lowers the exponent of v in the running time by 8. Let $J = (H\alpha+1)\lg v$, so that 2^J exceeds all running times, and let X bound J, the lengths of the names and all exponent bounds below. By the hypothesis every fixed power of X is less than v. We construct programs p that have the values required at scale v' in v^{f+8} steps, starting with the program `run` and $f + 8 = H\alpha + 1$.

One round, from $f + 8$ to f, for $f \ge \alpha(H - 32) + 40$. The powers that p may name are the $\xi_l^{2^i}$ with $i < B_p$, where B_p is its exponent bound, and their trie has $Q \cdot O(X)$ nodes and height at most X. On an instance z the new program guesses a vector e' of exponents for these powers, as in Lemma K.4, and multiplies by $\lambda^{e'}$ by naming powers $\xi_l^{2^{i+j}}$ with $i < B_p$ and $j < J$. As it also runs D, the new exponent bound is $\max(B_p + J, B)$. Then it forms $z_{e'}$ for p on z, cuts it to scale v by the program of Theorem J.4, with 2α for α, and runs D on every piece. By Lemma K.4 and Definitions J.1 and K.1 the new program has the same value as p.

Here $z_{e'}$ is plain within $\tau' = Xv^{f+8}$ steps, up to a constant factor. Its size is at most $v'^{H-32} + v^{4h_0\alpha} + v^{h_0\alpha+2}$, where the three terms bound z, p and e', and this is at most τ'/v^{48}. So Theorem J.4 gives at most $\tau' v^2/v^H$ pieces, each of which takes v^{H-16} steps under D, and listing them takes τ'/v^{24} steps. Altogether this is less than v^f, and the program grows by $v^{h_0\alpha+2} + v^{H+1}$.

The number of rounds. As $15\alpha - 8 \ge 40$ and $24\alpha \ge 17\alpha + 1$, after at most 3α rounds the time is at most v'^{H-17}, and the exponent bound is then at most $B + 3\alpha J \le B + 51h_0\alpha^2 \lg v$, as $H = 16h_0$. The final program contains the last p inside $z_{e'}$, the cutting program, and D, so its size is at most $3\alpha(v^{h_0\alpha+2} + v^{H+1}) \le v'^{4h_0}$, as $H + 1 \le 5H/4$.

Halting. A new program only writes $z_{e'}$ down, and the cutting program and D halt on all short instances, plain or not, by the first condition of Definition J.1 and property (i) of Definition K.1. Thus the new program halts whether or not p does. The construction is not circular, because D is one fixed program that is right on all short instances. □

K.4. **Nonexistence of accelerators.** We now show, by a diagonal argument, that accelerators of moderate size do not exist.

Theorem K.6 (nonexistence of small accelerators)**.** *There are C and v_1 such that for all integers $a \ge 0$ and $v \ge v_1$ with $(\lg v)^{Ca} \le v$ no nonzero commutative A-algebra has an accelerator of size v^a at scale v.*

Proof. Suppose that there is such an accelerator. We reduce its constants into a finite field, pass to a large scale by Proposition K.5, and diagonalize there. As A is finitely generated (Theorem G.2), so is the ring that its constants generate over A. Modulo a maximal ideal this ring is a field that is finitely generated as a ring, hence a finite field E [Eis95, Theorem 4.19], over which the program is still an accelerator, but we know nothing about the size of E.

All scales. Proposition K.5 with $\alpha = \max(5, a)$, and then again and again with $\alpha = 5$, gives programs with the two properties of an accelerator at arbitrarily large scales v_i, with the same constants, of size at most $v_i^{4h_0}$ and with exponent bounds $O((\lg v_i)^3)$. Its hypotheses hold again, since $Q \le v_i^{h_0}$, the trie is the same, and $(\lg v_i)^{5C'} \le v_i$, where C' is the constant of the proposition.

The diagonal. Let $v = v_i$ be large compared with the size of E and with the number of constants, let D be the program there and B its exponent bound, and let $2^J > v^{H-15}$. As E is finite, there is m_E such that for every $j \geq 0$ there is an $r(j) < m_E$ with $c^{2^j} = c^{2^{r(j)}}$ for all $c \in E$, and this is where finiteness is needed. Let $B' = \max(B, m_E)$, and let λ be the list of the powers $\xi_l^{2^i}$, $i < B'$, indexed by their names. Then every λ^e can be named with exponent bound B', because $(\xi_l^{2^i})^{2^j} = \xi_l^{2^{r(i+j)}}$. The program P contains the table of $r(j)$ for $j < B' + J$, and on an argument x it guesses a bit. For one value of the bit it accepts with weight 1, and for the other it guesses an exponent vector e for λ, multiplies by λ^e by naming, and runs D on the instance z_e that Lemma K.4 provides for the program x on the argument x.

The size of P is at most $v^{H/4}$ plus lower-order terms, so for $x = \mathsf{P}$ every z_e is short. Hence D halts on z_e, accepting within v^{H-16} steps, so all runs of P on P halt, the accepting ones within v^{H-15} steps, and the names in their tallies are among those of λ. By Lemma K.4 with $T = v^{H-15}$, the z_e are v^H-plain and the value V of P on P is $\sum_e \lambda^e \operatorname{sum}(z_e)$. On the other hand, D has the value $\operatorname{sum}(z_e)$ on z_e by property (ii), so the definition of P gives

$$V = 1 + \sum_e \lambda^e \operatorname{sum}(z_e) = 1 + V,$$

which is impossible in the field E. □

Remark. This is an indirect diagonalization in the sense of [FLMV05, vM07, SW14, MW17]. Compare the lower bounds for the permanent against uniform threshold circuits of Allender [All99] and of Koiran and Perifel [KP09]. The assumed speed-up is used at one scale only.

K.5. **The lower bound in the contracting case.** The lower bound now follows from Lemma K.3 and Theorem K.6 by a calculation.

Theorem K.7 (lower bound in the contracting case)**.** *In the contracting case there are $c > 0$ and m_0 such that for $m \geq m_0$ every circuit over $\Bbbk$ for PERM_m has more than $2^{c(\log m)^2/\log\log m}$ gates.*

Proof. Suppose, for a small $c > 0$ and some large m, that a circuit over $\Bbbk$ for PERM_m has at most $2^{c(\log m)^2/\log\log m}$ gates. Let $v = \lfloor m^{1/B_0} \rfloor$, so that $m \geq v^{B_0}$, and let a be the least integer such that the circuit has at most v^a gates, so that $a = O(c \log m/\log\log m) + 1$. For any fixed C' we have $(\lg v)^{C'(a+1)} \leq v$ if c is small enough and m is large. Hence Lemma K.3 gives an accelerator of size $v^{C(a+1)}$ at scale v over $\Bbbk$, which is a nonzero A-algebra, and this contradicts Theorem K.6 with $C(a+1)$ in place of a. □

Appendix L. Proofs of the main theorems

Sections A to K work over the real numbers, and this section proves the following three statements over $\mathbb{R}$. Theorem 1.1 and Corollary 1.2, which hold over every field, are then proved in Appendix M.

Theorem L.1 (main theorem)**.** *There are $c > 0$ and m_0 such that for all $m \geq m_0$ every circuit over $\mathbb{R}$ that computes PERM_m has more than*

$$m^{c(\log m)^{1/3}} = 2^{c(\log m)^{4/3}}$$

gates.

Theorem L.2 (no circuits of polynomial size)**.** *For every integer $C \geq 0$ there is m_0 such that for all $m \geq m_0$ no circuit over $\mathbb{R}$ with at most m^C gates computes PERM_m.*

Corollary L.3. *Over $\mathbb{R}$, VP is a proper subset of VNP.*

The flat case was treated in Section E and the contracting case in Sections F to K, and by Theorem D.2 at least one of them occurs.

Proof of Theorem L.1. By Theorem D.2 we are in the flat case or in the contracting case. In the flat case Theorem E.11 gives the bound $2^{c''(\log m)^{4/3}}$ for all large m, with an explicit constant $c'' > 0$. In the contracting case Theorem K.7 gives the bound $2^{c'(\log m)^2/\log\log m}$ for some $c' > 0$ and all large m, and $(\log m)^2/\log\log m \geq (\log m)^{4/3}$ for large m. Hence the smaller of c' and c'' serves as the constant c of the theorem. □

Theorem L.2 follows at once from Theorem L.1, since $m^{c(\log m)^{1/3}}$ exceeds m^C for large m. The following direct proof uses of Section E only the part that precedes Section E.4.

Proof of Theorem L.2. By Theorem D.2 we are in the flat case or in the contracting case. In the flat case the claim is Theorem E.6. In the contracting case it follows from Theorem K.7, whose bound exceeds m^C for large m. □

Proof of Corollary L.3. The inclusion VP $\subseteq$ VNP is the case $r(n) = 0$ of the definition. By Ryser's formula [Rys63], see also [Gly10],

$$\mathrm{PERM}_m = (-1)^m \sum_{e\in\{0,1\}^m} \prod_{j=1}^{m} (1-2e_j) \prod_{i=1}^{m} \sum_{j=1}^{m} e_j x_{ij}.$$

The summand is a polynomial in the x_{ij} and the e_j of degree at most $3m$ that has a circuit of size polynomial in m, so the family (PERM_m) is in VNP. By Theorem L.2 it is not in VP. □

Remarks and open questions. The constant m_0 in our theorems is not effective, and we do not claim that the exponent $1/3$ is optimal. We do not know whether we are in the flat case or in the contracting case, and we have not shown that the two cases exclude each other (see the end of Section D). To make m_0 effective along the lines of this proof we would have to decide which case holds, and in the contracting case we would also need values for the numbers μ and s of Definition D.1, which the proof of Theorem D.2 does not provide.

The exponent $1/3$ comes from the flat case alone, since the contracting case gives the stronger bound of Theorem K.7. Two quantities limit the argument in the flat case: the k bits that name a balanced set, and the scaled exponent $c\,g_\phi(P)$ of the set P of all ends, which Proposition E.9 bounds by about $\sqrt{k}$. We do not know whether the bound in the flat case can be improved.

Appendix M. All fields and the Hamiltonian cycle polynomial

This appendix proves the lower bound of Theorem L.1 over every field, with a constant that does not depend on the field (Theorem M.3), and deduces that VP $\neq$ VNP over every field (Corollary M.4). Sections A to L use nothing about $\Bbbk$ except that it is a field, apart from the element $\frac{1}{2}$ in Proposition E.1, and for uniform constants we apply them to a single field in each characteristic. Since the permanent is the determinant in characteristic two, we also treat the Hamiltonian cycle polynomial HC_M, for which Proposition M.2 replaces Proposition E.1 and needs no $\frac{1}{2}$.

M.1. Reduction to the algebraic closure of the prime field.

Let p be zero or a prime, and let $\mathbb{K}$ be the algebraic closure of the prime field of characteristic p.

Lemma M.1 (circuits with constants in $\mathbb{K}$)**.** *Let f be a polynomial with integer coefficients. If f has a circuit with L gates over some field of characteristic p, then it has a circuit with L gates over $\mathbb{K}$.*

Proof. We embed the given field into an algebraically closed field, which contains a copy of $\mathbb{K}$, and adjoin the constants of the circuit to this copy. That the circuit computes f is an identity in the resulting finitely generated $\mathbb{K}$-algebra, and we reduce it modulo a maximal ideal, whose residue field is $\mathbb{K}$ by the Nullstellensatz. □

M.2. **Sums over bits as Hamiltonian cycle polynomials.** The next proposition is the analogue of Proposition E.1 for the Hamiltonian cycle polynomial.

Proposition M.2 (sums over bits as Hamiltonian cycle polynomials)**.** *Let R be a commutative ring, and let the surjections $v : I \to V$ and $w : I \to K$ and the 2×2 matrices W_n over R be as in Proposition E.1. Then the sum of that proposition is* HC_M *of an $M \times M$ matrix whose entries are* 0, 1 *and single values $W_n(x, r)$, for every $M \geq 40|I|$.*

Proof. We construct a directed graph with weighted edges in which the products of the weights along the Hamiltonian cycles add up to the given sum, and we take its weighted adjacency matrix.

Segments and XORs. The graph is a cyclic row of *segments*, each of which offers a few parallel *options*, which are directed paths from one *junction* to the next. An *XOR* between two options consists of three new vertices a, b, c with edges in both directions along a–b–c, and the first option is led through it from a to c and the second from c to a. A Hamiltonian cycle must pass b one way or the other, and if it runs a, b, c then it follows the first option from junction to junction, also through several XORs in series. Hence it takes exactly one of the two options, and every choice with this property gives exactly one cycle. The weight of an option is put on its first edge, and in each segment at most one option passes no XOR, so that there are no parallel edges.

The graph. For every element of V and of K there is a segment with the two options T and F, which stand for the values 1 and 0 of the bits of x and of r. For every $n \in I$ there are seven further segments, namely the *selector* with an option s_t of weight $W_n(t)$ for each $t \in \{T, F\}^2$, for each t a *flag* with the options y_t and $\bar{y}_t$, the *x-check* with the options $\xi_{TT}, \xi_{TF}, \xi_0$, and the *$r$-check* with the options $\rho_{TT}, \rho_{FT}, \rho_0$. The ten XORs at n join $\bar{y}_t$ with s_t and, where they exist, with ξ_t and ρ_t, and they join ξ_0 and ρ_0 with the option T of the segments of $v(n)$ and of $w(n)$, respectively. Thus y_t is taken exactly for the one t for which s_t is taken, and the two checks then force this t to be $(x_{v(n)}, r_{w(n)})$. The Hamiltonian cycles thus correspond to the pairs (x, r), and the cycle of (x, r) has the weight $\prod_n W_n(x_{v(n)}, r_{w(n)})$.

The size. There are $|V| + |K| + 7|I|$ junctions and $10|I|$ XORs, which makes at most $39|I|$ vertices, and segments with a single option of weight 1 give every larger M. □

With HC_M in place of PERM_M and $40q$ in place of $3q$, Corollary E.2 therefore holds in every characteristic, and its entries are again affine in one variable each. We call a family (G_M) of polynomials over a field $\Bbbk$ in $M \times M$ variables *complete for grid sums with factor θ* if the following holds: whenever in Proposition E.1 the ring R is a polynomial ring over $\Bbbk$, with or without the element $\frac{1}{2}$, and the entries of every matrix W_n are affine in one variable, which depends on $w(n)$ only, the sum there is, for every $M \geq \theta|I|$, the value of G_M at a matrix whose entries are affine in one variable each. Thus (PERM_M) has this property with factor 3 if $2 \neq 0$, and (HC_M) with factor 40. Corollary E.2, Lemma E.3 and with it Theorem E.11 hold for every such family, with θ in place of 3. These constructions are in the tradition of Valiant and Malod [Val79, Mal03], and a completeness proof for the Hamiltonian cycle polynomial along these lines can be found in [FPV13].

M.3. **The lower bound over all fields.** We can now state the main theorem in its general form.

Theorem M.3 (lower bound over all fields)**.** *There is an absolute constant $c > 0$ such that for every p, zero or prime, there is m_0 with the following property. Over every field of characteristic p and for every $m \geq m_0$, every circuit for* HC_m*, and every circuit for* PERM_m *if $p \neq 2$, has more than $m^{c(\log m)^{1/3}}$ gates.*

Proof. By Lemma M.1 it suffices to consider circuits over $\mathbb{K}$, where Theorem D.2 gives the flat or the contracting case, depending on p. The constant of Theorem E.11 in the flat case is absolute, and the bound of Theorem K.7 in the contracting case, although its constant

depends on p, exceeds that of the flat case for all large m, as in the proof of Theorem L.1. For HC_m, Proposition M.2 replaces Proposition E.1, which enters through Corollary E.2 in Lemmas E.3 and K.3. There the factor 3 becomes 40, and the proofs only used that $3k^3\ell\, 2^{k\ell} \le 2^{2k\ell}$ for large values of the parameters. □

The separation of the classes follows as in Corollary L.3.

Corollary M.4. *Over every field,* VP *is a proper subset of* VNP*.*

Proof. The family (HC_m) is in VNP [Val79], as one sees by summing over one-hot codes of the places that the vertices take on the cycle, with vertex 1 at place 1. By Theorem M.3 it is not in VP. □

Appendix N. Approximate computation

In this appendix we extend Theorem M.3 to circuits that only approximate the polynomial, in the sense of border complexity (Theorem N.5), so that VNP is not contained in the closure $\overline{\mathrm{VP}}$ of VP. We first reduce to constants that are Laurent polynomials in a formal variable ε over $\mathbb{K}$, and in the flat case we then only have to adapt Lemma E.7. In the contracting case an approximating circuit gives a program that is correct only up to multiples of ε, and Proposition N.4, whose long proof may be skipped at first reading, excludes such programs.

N.1. **Approximating circuits.**

Definition N.1. Let $\Bbbk$ be a field, and let $\Bbbk((\varepsilon))$ be the field of formal Laurent series over $\Bbbk$. A circuit over $\Bbbk((\varepsilon))$ *approximates* a polynomial f over $\Bbbk$ if it computes a polynomial of the form $f + \varepsilon g$, where g is a polynomial all of whose coefficients lie in $\Bbbk[[\varepsilon]]$. We then fix one such polynomial and call it the *output* of the circuit. The class $\overline{\mathrm{VP}}$ is defined like VP, with approximating circuits in place of circuits.

Over an algebraically closed field, f is approximated with L gates if and only if it lies in the Zariski closure of the polynomials that have circuits with L gates [Ald84, Bür04], but we do not use this fact.

Lemma N.2 (constants that are Laurent polynomials)**.** *Let f be a polynomial with integer coefficients, and let $\Bbbk$ be a field of characteristic p. If a circuit with L gates over $\Bbbk((\varepsilon))$ approximates f, then some circuit with L gates over the ring $\mathbb{K}[\varepsilon, \varepsilon^{-1}]$ computes a polynomial that lies in $f + \varepsilon\, \mathbb{K}[\varepsilon][x]$.*

Proof. We cut every constant off above ε^N with $N = 2^L\Delta$, where no constant contains a power of ε below $\varepsilon^{-\Delta}$. Since the output has degree at most 2^L in the constants, the cut changes it only in the positive powers of ε. We then apply the argument of Lemma M.1 to the coefficients of the constants, which are now Laurent polynomials. □

N.2. **Approximating circuits in the flat case.**

The flat case needs Lemma E.7 for functionals that are Laurent polynomials.

Lemma N.3 (sums with few terms, approximate version)**.** *Lemma E.7 remains true if the functionals of which the f_j consist have coefficients in $\Bbbk[\varepsilon, \varepsilon^{-1}]$, no negative power of ε occurs in $\sum_j f_j(T_j)$, and t is the coefficient of ε^0 in this sum.*

Proof. We follow the proof of Lemma E.7, and let all exponents of ε in the functionals lie in $[-\Delta, \Delta]$. As no negative powers occur, t^n is the coefficient of ε^0 in the nth power of the sum, which is again a sum over the types. For each type the functional at a party outside P is a sum of at most $2n\Delta + 1$ functionals over $\Bbbk$, one for each power of ε, so that this coefficient is covered by $(2n\Delta + 1)^{|V|}$ copies of the same tensor as before, a subexponential number. □

It follows that Proposition E.10, and with it Theorem E.11, holds for approximating circuits. Indeed, we apply Lemma E.4, which holds over every commutative ring, over $\Bbbk((\varepsilon))$, cut the functionals off at a power of ε that is high against their negative exponents times the number of parties, and use Lemma N.3 in place of Lemma E.7.

N.3. **Approximate accelerators in the contracting case.** For the contracting case we weaken the notion of an accelerator over a commutative A-algebra R.

Definition. An *approximate accelerator* over R is defined as in Definition K.1, except that its constants lie in $R[\varepsilon, \varepsilon^{-1}]$ and that in property (ii) its value on a v^H-plain instance z is only required to lie in $\operatorname{sum}(z) + \varepsilon R[\varepsilon]$.

Approximate accelerators of moderate size do not exist either.

Proposition N.4 (no approximate accelerators)**.** *Theorem K.6 holds for approximate accelerators.*

Formalization note. The proof below describes the programs and accounts for their sizes and running times to the level of exponents. That the programs do what is claimed is proved only in the Lean development.

Proof. We carry the given program to larger scales by the rounds of Proposition K.5, keeping the error a multiple of ε, and at a large scale we extract the coefficient of ε^0.

Suppose that there is an approximate accelerator of size v^a at scale v over a nonzero A-algebra E. Let Δ bound the absolute values of the exponents of ε in its constants, and for $r \geq 1$ let σ_r be the substitution $\varepsilon \mapsto \varepsilon^r$, an endomorphism of $E[\varepsilon, \varepsilon^{-1}]$. All constants below arise from the given ones by such substitutions, so that the nth power of each of them has at most $2n\Delta + 1$ terms. We carry the program to the scales $u = v^{a+5}$, u^5, u^{25}, and so on, keeping at the scale u a program of size at most u^{4h_0} and a trie of at most u such constants. The height of the trie and the exponent bound are $b_0 + O(\lg u)$, where $b_0 \leq ((a+1)\lg v)^{O(1)}$ depends on the first step only, and the exponents of ε in the constants are at most $2^{(\lg u)^{O(1)}}$ in absolute value, with a constant in $O(1)$ that does not depend on Δ for large u.

One round. We follow the proof of Proposition K.5 from a program D, with constants ξ', to the next scale. Let p be a program whose values on plain instances z lie in $\operatorname{sum}(z) + \varepsilon E[\varepsilon]$. By Lemma K.4 its value is exactly $\sum_e \lambda^e \operatorname{sum}(z_e)$, where λ lists the powers that p names and $\operatorname{sum}(z_e) \in A$. Let r exceed the absolute values of the exponents of ε in all λ^e. The new program guesses e, multiplies by λ^e, and cuts z_e into pieces. For the list of pieces it guesses once a vector e'' of exponents for the constants $\sigma_r(\xi')$, multiplies by these powers, and lets the interpreter of Lemma K.4 run D on all pieces, counting the names against e''. By that lemma the result is the product, over the pieces y, of σ_r applied to the value of D at y, which lies in $\prod_y \operatorname{sum}(y) + \varepsilon^r E[\varepsilon]$ if the pieces are plain. Summed over the runs of the cutting program, whose weights lie in A, the products $\prod_y \operatorname{sum}(y)$ give exactly $\operatorname{sum}(z_e)$ by Definition J.1. The new value therefore lies in

$$\sum_e \lambda^e \big(\operatorname{sum}(z_e) + \varepsilon^r E[\varepsilon]\big) \subseteq (\text{value of } p) + \varepsilon E[\varepsilon] \subseteq \operatorname{sum}(z) + \varepsilon E[\varepsilon].$$

The new trie has the old one and the trie of the constants $\sigma_r(\xi')$ as its two halves, so that a round adds one to the height, at most the number of constants of D to the number of constants, and J to the exponent bound B of the names, where J is as in the proof of Proposition K.5. The logarithm of the largest exponent of ε becomes at most the old one, plus that of the constants of D, plus $O(J + B + \lg u)$. From u to u^5 there are at most 15 rounds, all with the same D, so that this logarithm is multiplied by at most 16 and increased by $O(b_0 + \lg u)$, while $\lg u$ is multiplied by 5. Hence it remains $(\lg u)^{O(1)}$. In general there are $O(\alpha)$ rounds from a scale to the next, so that the invariant is kept, and the running time again has a factor u to spare.

Back to exact values. Let u be one of these scales, large compared with v, Δ and b_0, let D be the program there, and let $2^J > u^H$. We choose $\beta = O(\lg u)$ and $W = (\lg u)^{O(1)}$ such that every relevant power of a constant has at most 2^β terms and every product of such powers has exponents of ε below 2^W in absolute value. The new program does the following for every named power $c = \xi_l^{2^i}$ with $i < B$. It guesses the binary digits of the number $n < 2^J$

of times that this name is to be used, and a number $k < 2^\beta$. It names the coefficient of the kth term of c^n, an element of E that is 0 if there is no such term, and adds the exponent of this term to a counter modulo 2^W. These exponents form a table of $u^{O(1)}$ numbers in the program. Then the program runs the interpreter on D, counting each name against its number n, and it accepts if the interpreter does and the counter is 0.

Expanding the powers c^n into their terms shows, as the true sum of the exponents is below 2^W in absolute value, that the value of the new program is the coefficient of ε^0 in the value of D, which is $\mathrm{sum}(z)$ on plain instances. The coefficients are at most $u^{O(1)}$ new constants with the exponent bound $B = 1$, since n and k have $O(\lg u)$ digits. The guesses and the interpreter increase the running time by a polylogarithmic factor, so that the accepting runs take at most u^{H-16} steps. This is an accelerator over E of size $u^{O(1)}$ with a fixed exponent, which contradicts Theorem K.6. □

N.4. **The lower bound for approximating circuits.** We can now prove the lower bound for approximating circuits.

Theorem N.5 (approximate computation)**.** *Theorem M.3 holds for circuits that approximate* HC_m*, respectively* PERM_m*. In particular* $\mathrm{VNP} \not\subseteq \overline{\mathrm{VP}}$ *over every field.*

Proof. By Lemma N.2 we may assume that the constants of the circuit lie in $\mathbb{K}[\varepsilon, \varepsilon^{-1}]$. The flat case was treated after Lemma N.3. In the contracting case the proofs of Lemmas K.2 and K.3 work over this ring and give approximate accelerators. Indeed, the output $\Phi + \varepsilon g$ of the circuit may have a large degree, but the program of Lemma K.2 gives its part of degree at most d, which is Φ plus ε times that part of g. Proposition N.4 excludes them, with the calculation of Theorems K.7 and M.3. For the last claim, (HC_m) is in VNP. □

Remark N.6 (determinantal complexity)**.** This remark is the one numbered statement of the paper that has not been formalized in Lean. Since the determinant has circuits of polynomial size [Ber84], Theorem N.5 has the following consequence. Over fields of characteristic different from two there is $c > 0$ such that, for all large m, the polynomial PERM_m is not approximated, in the sense of Definition N.1, by determinants of affine matrices of size $m^{c(\log m)^{1/3}}$. Over $\mathbb{C}$, and for matrices of polynomial size, this was conjectured by Mulmuley and Sohoni [MS01] in the language of orbit closures.

Appendix O. Proofs of the standard facts on CAT(0) spaces

In this appendix we prove Lemma C.4 from the definition of a CAT(0) space in Section C.4. None of the arguments is new. Note that midpoints are unique, since the right-hand side of the CN inequality vanishes if z is a second midpoint of x and v.

Proof of Lemma C.4. Minimization. Let $F \subseteq Y$ be nonempty, closed and convex, let $c > 0$, and let $G \geq 0$ be a continuous function on F with $G(m) \leq \frac{1}{2}G(q) + \frac{1}{2}G(q') - \frac{c}{4}|qq'|^2$ for the midpoint m of any two points q, q' of F. Then minimizing sequences in F are Cauchy sequences, so that G has a minimizer b on F. If $G(y) \geq G(b) + c'|yb|^2$ for all $y \in F$, then this bound at the midpoint of b and y, together with the hypothesis on G, gives the same with $\frac{c+c'}{2}$ in place of c'. Repeating this from $c' = 0$ we get in the limit

$$G(y) \geq G(b) + c|yb|^2 \qquad \text{for all } y \in F. \tag{O.1}$$

By the CN inequality this applies to $G(y) = \sum_i w_i |yx_i|^2$ on $F = Y$, for weights $w_i \geq 0$ with $c = \sum_i w_i > 0$, and to $G(q) = |yq|^2$ on the set F of (d), with $c = 1$.

Part (a). Let γ be the minimizer for the points x, v with the weights $1 - t, t$. By (O.1) we have $|z\gamma|^2 \leq (1-t)|zx|^2 + t|zv|^2 - (1-t)|\gamma x|^2 - t|\gamma v|^2$, and $(1-t)|\gamma x|^2 + t|\gamma v|^2 \geq t(1-t)(|\gamma x| + |\gamma v|)^2 \geq t(1-t)|xv|^2$, which together are the inequality. At $z = x$ and at $z = v$ it gives $|x\gamma| \leq t|xv|$ and $|\gamma v| \leq (1-t)|xv|$, and hence equality in both. The uniqueness is the inequality at a second point z with the same distances, for which the right-hand side is 0.

Part (b). Let $0 < \eta < 1$ and let z be the point at η on $[y, u]$. By the triangle inequality and the Cauchy–Schwarz inequality we have $|xv|^2 \le |xz|^2/\eta + |zv|^2/(1-\eta)$. We bound $|xz|^2$ and $|zv|^2$ by (a) and choose $\eta = |xy|/(|xy| + |uv|)$, taking a limit if this is 0 or 1, which gives the first statement. The second follows because $2|xy||uv| \le |xy|^2 + |uv|^2$. Equality in the second statement forces $|xy| = |uv|$, and likewise $|xu| = |yv|$ when the roles of y and u are exchanged.

Parts (c) and (d). The barycenter of (c), with all weights equal to 1, and the nearest point of (d) are minimizers as in the first step, and the inequalities of (c) and (d) are (O.1), which also shows that there is only one minimizer. For the last statement of (c) we add the inequality of (c) for the points x_i at $y = b'$ and for the points x_i' at $y = b$, and we apply (b) to the four points b', x_i, b, x_i' for every i, which together give $n|bb'|^2 \le |bb'| \sum_i |x_i x_i'|$.

Part (e). The product is complete, midpoints are taken coordinate by coordinate, and the CN inequality for the product is the sum of those for the coordinates.

Part (f). Let $\Phi(p, p'; r, r') = |pr'|^2 + |p'r|^2 - |pr|^2 - |p'r'|^2$, so that $\Phi(p, p'; r, r') \le 2|pp'||rr'|$ by (b). We have $\Phi(x, u; y, v) = \Phi(x, a; y, b) + \Phi(x, a; b, v) + \Phi(a, u; y, b) + \Phi(a, u; b, v)$, and the bounds for the four terms add up to $2|xu||yv| = 2h^2$, which by the hypotheses is the value of $\Phi(x, u; y, v)$. So all four bounds are equalities, and the first one says that

$$|xb|^2 + |ya|^2 - \delta^2 - |ab|^2 = 2sth^2. \tag{O.2}$$

Let first $s = t$. By (b) we have $|xb|^2 + |ya|^2 \le |xa|^2 + |yb|^2 + 2|xy||ab| = 2s^2h^2 + 2\delta|ab|$, so that $(|ab| - \delta)^2 \le 0$ and $|xb|^2 + |ya|^2 = 2\delta^2 + 2s^2h^2$. By (a) both $|xb|^2$ and $|ya|^2$ are at most $\delta^2 + s^2h^2$, so both are equal to it. As b does not depend on s and a does not depend on t, it follows that $|xb|^2 = \delta^2 + t^2h^2$ and $|ya|^2 = \delta^2 + s^2h^2$ for all s and t, and inserting this into (O.2) gives the claim. □

Appendix P. Glossary of coined terms

This appendix lists the main terms that are introduced in this paper or used in a special sense and says where each is defined.

term	meaning	defined in
wire $\langle q \rangle_{uv}$	the unit tensor on two parties $u \ne v$; it has dimension q and carries $\log q$ bits	§A
spectral point ϕ	a nonnegative function on tensors with $\phi(1) = 1$ that is multiplicative, does not increase under restriction and is subadditive	Def. A.1
$g_\phi(P)$, $d_\phi(u, v)$	the exponents in $\phi(\langle q \rangle_P) = q^{g_\phi(P)}$; $d_\phi(u, v) = g_\phi(\{u, v\})$	Lemma A.2
price; $\lvert xy \rvert$	a limit of multiples $c\, d_\phi$; the distance of x and y in it	Def. A.4
hub	one of two additional parties through which a tensor of small rank is routed	Lemma A.3
price space	a complete CAT(0) space whose metric is a price	§C
clause sum	a sum over bit vectors of a product of weights of satisfied clauses	§E
complete network; $f_{k,\ell}$	the tensor network on the complete graph on k nodes with ℓ bits on every edge; its polynomial	§E
load	a vector of natural numbers on pairs of elements, or on edges	§G
law; slot; channel	an identity that rewrites bits on arbitrary pairs (slots) as a sum of networks with bits on edges (channels)	Def. G.1
packet	a string of ℓ bits carried between two nodes	§H
network; node tensor; value; outcome	numbers of packets on pairs of nodes and a tensor at each node; the sum over all contents of the packets of the product of the node tensors; the numbers of packets that a law puts on its channels	§H
run; weight; tally	a computation of a nondeterministic program; the element of A that it accumulates; the list of names of constants that it has used	§I
instance z; sum(z)	a program with an argument; the sum of the weights of its accepting runs	§I
τ-plain	all runs halt, and the accepting ones have empty tallies and take at most τ steps	§I
tape automaton	a finite automaton with a matrix over A for each letter, whose letters write on the two sides of tapes	§I

term	meaning	defined in
accelerator	a program that evaluates sums over v^H steps within v^{H-16} steps	Def. K.1

Appendix Q. The Lean development

This appendix describes the formalization on which Section 3 relies. The file of definitions and statements is printed in Section 3. Here we say how it can be checked, list the Lean theorem that belongs to each numbered statement of the paper, and say where the Lean development differs from the text.

The development uses Lean 4.33.1 [Lean] and Mathlib [Mathlib] at commit

`0df444a360eaa60ab8c11dca51a86af692955474`.

The file printed in Section 3 imports Mathlib and nothing else, and it is printed as it stands in the development, without the license comment of five lines at its head. It defines circuits, the two polynomials, approximating circuits and Valiant's classes, and it states four propositions: `MainTheorem` is Theorem M.3, `MainTheoremApprox` is the first assertion of Theorem N.5, `Separation` is Corollary M.4, and `SeparationClosure` is the second assertion of Theorem N.5. In the namespace `Solution` the theorems `mainTheorem`, `mainTheoremApprox`, `separation` and `separationClosure` have exactly the types `MainTheorem`, `MainTheoremApprox`, `Separation` and `SeparationClosure`, and `permanent_over_real` is Theorem L.1.

Q.1. **Checks of the development.** The folder `https://github.com/anthropics/formal-math/tree/main/vp-vnp` builds with Lean 4.33.1 and Mathlib's compiled files. It uses no `native_decide` and declares no axioms, and its only uses of `sorry` are in `Challenge.lean`, which states the theorems of `Solution` without proof so that the tool Comparator can compare against them. The theorems of `Solution` depend on no axioms other than `propext`, `Classical.choice` and `Quot.sound`, as `scripts/PrintAxioms.lean` prints. Comparator accepted them against `Challenge.lean`, rechecking the proofs with Lean's kernel and with the independently written checker nanoda. The file printed in Section 3 is `Statement.lean`, without the license comment at its head.

Whether the statements say what is meant was judged only by AI readers. No human mathematician has checked the development, the statement file or the proofs.

Q.2. **The numbered statements in Lean.** Every numbered statement of the paper except Remark N.6 has a Lean theorem. Table 3 lists them in the order of the paper, and for some statements it names a second Lean theorem; it also lists the display (F.1), and `eq_Q` is the display (B.2).

Q.3. **Differences between the text and the Lean development.** The theorems with prefix `Paper.` state the numbered statements, and each of them is proved once. The theorems of `Solution` are proved through them, with these exceptions, which are not on that route: Theorem L.2, Corollary L.3, the qualitative bound in the flat case (Proposition E.5 and Theorem E.6) and the definitions. Many unnumbered steps of the printed proofs have named theorems too. Lean does not identify equivalent tensors (`def_spectral_congr` shows that spectral points respect the identifications), and it writes the standing hypotheses out. Some theorems hold there for every field, or under weaker hypotheses. The theorems with prefix `Paper.` are stated for fields in the lowest universe, and so are their rings, algebras, groups, spaces, point sets and index sets. The exception is the given field in Theorems M.3 and N.5, Corollary M.4, Definition N.1 and Lemmas M.1 and N.2, which may lie in any universe. This restricts the printed statements to the lowest universe. The theorems of `Solution` hold for fields in every universe.

The Lean development uses some other words. It says *gauge* for an upper functional, *unit* for a packet, *table* for a node tensor, *bill* for a tally, *near pair* for an edge of the Cayley graph

Statement	Lean (prefix `Paper.`)
Section A	
Definition A.1	`def_spectral`, `def_spectral_congr`
Lemma A.2	`lem_exponents`
Lemma A.3	`lem_hubs`, `lem_hubs_further`
Definition A.4	`def_price`
Proposition A.5	`prop_gauge`
Theorem A.6	`thm_duality`
Proposition A.7	`prop_extend`
Proposition A.8	`prop_random`
Section B	
Lemma B.1	`lem_count`
Proposition B.2	`prop_identity`
Proposition B.3	`prop_crossing`, `eq_Q`
Section C	
Theorem C.1	`thm_cn`
Lemma C.2	`lem_means`
Proposition C.3	`prop_space`
Lemma C.4	`lem_toolkit`, `lem_toolkit_a` to `_f`, `lem_toolkit_c_minimizer`
Section D	
Definition D.1	`def_cases`
Theorem D.2	`thm_alt`
Section E	
Proposition E.1	`prop_grid`
Corollary E.2	`cor_clauses`
Lemma E.3	`lem_fkl_perm`, `lem_fkl_perm_printed`
Lemma E.4	`lem_balanced`
Proposition E.5	`prop_flat_lower`, `prop_flat_lower_printed`
Theorem E.6	`thm_flat_simple`
Lemma E.7	`lem_few`
Lemma E.8	`lem_convex`
Proposition E.9	`prop_cube`
Proposition E.10	`prop_flat_rate`, `prop_flat_rate_printed`
Theorem E.11	`thm_flat_rate`

Statement	Lean (prefix `Paper.`)
Section F	
Lemma F.1	`lem_growth`
Lemma F.2	`lem_words`
Lemma F.3	`lem_first`
Display (F.1)	`eq_window`, `projection_nearest`
Lemma F.4	`lem_tele`
Lemma F.5	`lem_angle`
Theorem F.6	`thm_poincare`
Section G	
Definition G.1	`def_law`
Theorem G.2	`thm_laws`
Section H	
Proposition H.1	`prop_tower`, `prop_tower_representatives`
Theorem H.2	`thm_tree`, `thm_tree_good`
Section I	
Proposition I.1	`prop_universal`
Lemma I.2	`lem_runnet`
Section J	
Definition J.1	`def_cut`
Proposition J.2	`prop_pieces`, `prop_pieces_identity`
Theorem J.3	`thm_onecut`
Theorem J.4	`thm_cut`, `exists_setting`
Section K	
Definition K.1	`def_acc`
Lemma K.2	`lem_lowdeg`
Lemma K.3	`lem_circuit`, `lem_circuit_perm`
Lemma K.4	`lem_postpone`
Proposition K.5	`prop_next`, `scales_setting`
Theorem K.6	`thm_noacc`, `scales_setting`
Theorem K.7	`thm_contracting`
Section L	
Theorem L.1	`thm_main`, `thm_main_form`
Theorem L.2	`thm_simple`
Corollary L.3	`cor_classes`
Appendix M	
Lemma M.1	`lem_transfer`
Proposition M.2	`prop_hc_grid`, `cor_clauses_hc`
Theorem M.3	`thm_allfields`
Corollary M.4	`cor_allclasses`
Appendix N	
Definition N.1	`def_approx`
Lemma N.2	`lem_transfer_approx`
Lemma N.3	`lem_few_approx`
Proposition N.4	`prop_approx_acc`, `prop_approx_acc_setting`
Theorem N.5	`thm_approx`
Remark N.6	not formalized

TABLE 3. The Lean theorems that belong to the numbered statements.

of U_h, *window* for the boxes of Section C as well as for the windows of Section F, and *state* for the symbols of Section B. The other differences that are known are listed below.

Statements. The statements differ in the following points.

- The numbers C, a and α are natural numbers.
- In the file printed in Section 3 circuits and the classes VP and VNP are defined over commutative semirings. The permanent is written there as $\sum_\sigma \prod_i x_{\sigma(i),i}$, where Section 1 has $x_{i,\sigma(i)}$; the two sums agree, as one sees by replacing σ with its inverse.
- Lean defines the projection to the fixed points of a window (Section F) as the barycenter of the orbit, and `projection_nearest` shows that this is the nearest point. In the proof of Theorem F.6 the space of functions carries the sum where the text has the mean, which is one scaling factor. Lean states Lemma F.3 for explicit numbers, $\vartheta_n = 18\lceil \log_2 n \rceil\, \mu(L+1)s^L$ with $L = \lfloor \log_\mu 2^{\lceil \log_2 n \rceil} \rfloor$ (`kappa`), which are not those of the printed proof, and $C(n)$ in (F.1) and in Lemmas F.4 and F.5 is formed with them.
- The families of tensors in Theorem A.6 and of maps in Proposition A.7 are indexed by any finite set.
- A variable of block a of $f_{k,\ell}$ is indexed by k strings, and the one at the place a is 0 in the variables that occur. The circuit sizes are the same, and the theorems with suffix `_printed` state Lemma E.3 and Propositions E.5 and E.10 for the printed polynomial.
- Lemmas A.3, E.4, E.7, F.1, F.2, I.2, K.3, M.1, N.2 and N.3 and Proposition B.2 are proved in stronger or more general forms. For example, `lem_hubs` pushes the tensor t forward along an arbitrary map from its parties to a finite set of parties, of which o_1, o_2 are any two, and `lem_hubs_further` is the form with two further parties. Proposition B.2 is proved without the assumption that Σ_1 and Σ_2 are nonempty. Lemma E.4 is proved over every commutative semiring, and Lemmas E.7 and N.3 whenever t is covered by $T_1, \dots, T_J$, so that linear maps may also be applied at the parties of P. In Section F Lemma F.1 is proved without the assumption $s^2 < \mu$ and also for $r = 0$, and Lemma F.2 without the conditions $1 \le k < n$ and $\xi, \omega \neq 0$. Lemma K.3 is proved for every family that is complete for grid sums (Appendix M) and for every $H \ge 32$. Lemmas M.1 and N.2 are proved with any algebraically closed field of characteristic p in the place of $\mathbb{K}$. The flat bound for approximating circuits has 14 where Proposition E.10 has 16.
- Laws may have any finite sets of elements, slots and channels. A law in Lean also records that its channels connect its elements, in the form that indices which agree along all channels agree everywhere; on U_h, where the channels are all the edges, this holds by itself. The conditions on λ, on $\mathcal{I}$ and on U_h come with Theorem G.2. Its Lean form also states the choices of Section G: u and C_0 are as in Theorem F.6, $C_0' \ge 2C_0 + 1$ is an integer and $C_1 = 4C_0'$. It asserts that some multiple $h \ge 2u$ of u satisfies (G.1), and it asserts (a) and (b) for every such h. That the channels there are the edges, each once, on which the norm in Theorem G.2(b) depends, follows from the definitions but is stated by no theorem of the development. The theorem `def_law` lists only three of the fields of the structure, in one direction, so that it rests on reading the structure that it is the printed notion.
- In Proposition H.1, the set V_0 is only stated to have as many elements as G_n, bijectivity on cosets is stated as equivariance and freeness, and the cosets over a coset are given by representatives (`prop_tower_representatives`: there is exactly one in each coset). Lean states the proposition for every $h \ge 3$ and every $n \ge 0$.
- Theorem H.2 and Proposition J.2 state their clauses as the fields of structures. A relay is a law with one slot, two channels, one index and $\lambda = 0$, applied to a batch of packets. Thus relays too have outcomes and leave copies and dead packets at the ends of their strings, an outcome of a relay is good if it equals the number of packets of the batch on both channels, and there is a law move for every stage, type, coset and law, with

$t = 0$ for those not chosen. The value of a tree is written as a sum, over the contents of named packets, present or absent, of products of the values of its nodes, and no theorem restates this as the value of a network as defined in Section H. As printed in (b) and (e), the dead packets are a set of names that are not all stated to be present, so that the bound on what a node ever holds may count absent ones. The end packets of relays count as dead, and in networks without these packets they would count among the absent ones. In part (e) Lean has the constant C of the theorem, and it counts for every move at p, with the number t, also t times one plus the number of its slots at p, which covers the zeros in slots. Proposition J.2 uses only the total M^C. Packets have a direction in Lean, and packets from a node to itself are allowed and stay at their node. The programs may depend on M, N, ℓ, the ring of the network and the bound on the number of given nodes, respectively on v, T, k, the automaton and the standing data. In Proposition J.2 the outcomes range over tuples of numbers below a bound that is part of the data (the field `U1`); no clause says that every good outcome lies below it, and the identity sums over those that do. Whether an outcome is good is tested with the integer $\lfloor 2C_1 a_1^2 \rfloor$ (`thm_tree_good`). This implies the printed notion and is equal to it when $2C_1 a_1^2$ is an integer, as the laws provide.

- Proposition J.2 is proved without the bound on $\lg T$, and its number of bits in all counts every copy ever held and the zeros in slots. Its sum runs over the numbers of packets with at most N packets at each node, as in the printed proof, where Lean takes for N the last bound of Lemma I.2, it is stated for every pair of a first and a last state, and the exponent of Λ is the constant C of the statement. It also states what the program needs besides: the packets have at most T bits, at most $T\Lambda$ bits start at a node, and the numbers of the program have room for packets, outcomes and pieces.
- Theorem H.2, Proposition J.2 and Theorems J.3 and J.4 have the laws, and the last two the generators of A, as explicit standing hypotheses. The laws may be any finite family of laws, over any commutative ring, that has the properties of Theorem G.2, with $h \geq 3$ and (G.1) as hypotheses; property (b) of Theorem G.2 enters as $b \sum_e (\lambda + \mathrm{len}_{i,e})^2 \leq G_1$ for an integer $G_1 \leq 2C_1 a_1^2$. In the contracting case `exists_laws` and `exists_generates` supply these hypotheses, and `exists_setting` states the conclusion of Theorem J.4 without them. The programs of Theorem J.3 are written out in full, where we describe them (`thm_onecut_evaluator`, `thm_onecut_program`). The quotients in the bounds of Theorems J.3 and J.4 are rounded down in Lean.
- Proposition K.5, Theorem K.6 and Proposition N.4 do not assume the contracting case. They take the conclusion of Theorem J.4 as a hypothesis (`CutHyp`), for a commutative ring A, which has to be finitely generated only in Theorem K.6 and Proposition N.4, and $H = 16h$ with $h \geq 17$, and `scales_setting` and `prop_approx_acc_setting` are their instances for the A and H of Section J in the contracting case.
- In Section C Lean fixes one particular free ultrafilter, the one that Mathlib calls `hyperfilter`, and counts the prices W_n from $n = 0$; the Lean form of Lemma C.2 also asserts that the limit that defines N exists and equals the infimum. In Proposition M.2 Lean has $\mathrm{HC}_0 = 1$, which matters only for empty I. The theorem `def_approx` states when a circuit approximates a polynomial, as in the second sentence of Definition N.1; the last sentence of the definition is the definition of `Paper.VPBar`, and the match rests on reading it.
- In Section I.1, Lean reads the construct of a pair from the length of its first component as a list, so that this component need not be the number of a construct, and such a pair is read as `fail` only if the length is above 10.
- In Proposition I.1, that the two symbols of the word of an instance are different is a separate theorem (`universal_symbols_ne`). Lemma K.2 is stated for the alphabet of these two symbols and the end mark and with explicit constants, in which the time is $O((d+1)\log(L+n) + (|z|+1)\log n)$; its bound on the number of steps is for accepting

runs, and halting is stated for the z with $|z| < n$. In Lemma K.3, `lem_circuit_phi` states what is used of Φ, not that it equals the printed sum over words.

Routes of proof. The theorems of the main route follow the references between the printed proofs, with these exceptions.

- In Proposition K.5, Theorem K.6 and Proposition N.4 some printed references are to definitions or enter as standing hypotheses. Proposition N.4 repeats the iteration of Proposition K.5, as printed.
- Lemma K.3, Theorem M.3 and Theorem N.5 use the general forms for families that are complete for grid sums, of which Corollary E.2 and the statements about the permanent are instances.
- Proposition E.10 uses the step `prop_flat_lower_sum` of the proof of Proposition E.5, not the proposition. Lemma N.2 repeats the argument of Lemma M.1 on coefficients.
- Theorem N.5 does not cite Lemma K.3, Theorem K.7 and Theorem M.3. Instead it uses the step `thm_allfields_exceeds` of the proof of the last of them and approximate analogues of the first two, which share their steps with them; the analogues are `lem_circuit_approx` and `contracting_approx_complete`. On the way the constants are cut off twice.
- Of Lemma A.2 only the clauses that say that d_ϕ is a pseudometric are taken from the lemma: the power law and $d_\phi \leq 1$ come from its named parts, and the bound by a tree is used nowhere. The clause of Lemma K.4 on the size is never cited and is proved again where it is needed. Part (d) of Lemma C.4 is redundant where it is used. The form of the constants in Lemma N.2 is derived again in the proof of Theorem N.5.
- Theorem H.2 uses parts (a) and (b) of Proposition H.1 through the named theorems that also prove the proposition.

The proofs of Proposition A.5 and Theorem A.6 are carried out in a commutative monoid with a covering relation, which is then taken to be the classes of tensors. In the proof of Theorem A.6 the certificate is first raised by one level, and the number of terms is bounded by $(c+1)^{k+1}\eta^{-N}$. Some steps exist only under names of the underlying library and not as theorems with prefix `Paper.` These are the sorting pass, the tape encoding and the time count of Proposition I.1, the constructions of Propositions E.1 and M.2, the induction of Lemma E.4, the word in Lemma F.2, the averaging in Lemma F.1, the properties of the mean $\mathbb{E}$ in Section C, and the semantics of scripts inside the evaluator. Lean proofs can be coarser or more general at single steps. The proof of Theorem G.2, for example, uses all loads of bounded norm where the text uses the rounded ones, and the proof of Theorem H.2 uses (G.1) as it stands where the printed proof uses only $1 + 4C_1 \leq h/4$.

Not in Lean. Remark N.6, the example of flattening ranks after Definition A.1, the remarks that follow the proof of Theorem D.2, the fact that r in the proof of Proposition A.5 is also a limit, and the factor three between the two ways of counting gates in Section 1 are not formalized.

Comments in the Lean files are informal: some paraphrase the paper, say a little more than the step that they name proves, or describe a bound or a definition loosely. The theorems follow the present text, up to the differences listed.