# Taming the Wild Stratum: Class-2 Groups of Bounded Rank-Width

**Faried Abu Zaid**
Independent Researcher, Germany

## Abstract

Finite $p$-groups of nilpotency class 2 already account for the leading term in the enumeration of all finite $p$-groups, while their classification contains tensor isomorphism and graph interpretation. We isolate a tractable part of this class by bounding the rank-width of the commutation tensor. Over $R = \mathbb{Z}/p^d$, the width of a cut is the number of generators required by six crossing modules. For fixed $p, d, r$, central extensions with distinguished kernel over $R$ and a site-tree layout of width at most $r$ form a uniformly tree-automatic class; chain layouts give a uniformly word-automatic class. Finite local certificates make the advice language exact.

Two phenomena absent over fields drive the proof: sibling interactions need not be bilinear in chosen interface representatives, and distributed central targets create obligations flowing into a subtree. Finite pairing tables and bounded claim modules handle these effects. From certified advice, every fixed first-order query is decidable in linear time, and a sentence in time $O(\log |G|)$; the uniform theories are decidable. The class is closed under direct products and contains several familiar structured families. Yet, for fixed parameters, it represents only $2^{O(m)}$ isomorphism types of order $p^m$, and is therefore asymptotically negligible among class-2 $p$-groups.

**2012 ACM Subject Classification** Theory of computation → Finite Model Theory; Theory of computation → Regular languages; Computing methodologies → Symbolic and algebraic algorithms

**Keywords and phrases** automatic structures, advice automata, finite $p$-groups, rank-width, chain rings, group isomorphism, first-order model checking

**Supplementary Material** *Software (Source Code)*: `https://github.com/fariedabuzaid/AutStr`

## 1 Introduction

Two research programmes meet in this paper.

**Algorithmic group theory.** Among finite $p$-groups, nilpotent groups of class 2 already account for the leading term in the enumeration problem. The Higman–Sims bounds show that the number of groups of order $p^m$ is $p^{\frac{2}{27}m^3+O(m^{8/3})}$, and class-2 groups already attain the lower bound; for odd $p$ they may be chosen to have exponent $p$ [13, 24, 23]. In this odd-prime, exponent-$p$ setting, the Baer correspondence [1] encodes the group, up to the standard bookkeeping for central direct factors, by an alternating bilinear commutator map $B\colon \mathbb{F}_p^n \times \mathbb{F}_p^n \to \mathbb{F}_p^k$. The resulting classification problem is pseudo-isometry of alternating maps, equivalently isometry of alternating matrix spaces, and is complete for the tensor-isomorphism class [10]. Polynomial-time algorithms are known for important restricted cases, including groups of genus two [4] and groups whose commutator subgroup is sufficiently small [16]. Model theory gives a complementary measure of generality: for every odd prime $p$, Mekler constructs class-2, exponent-$p$ groups in which suitable graphs are interpretable, transferring stability questions from graphs to groups [19]. Characteristic 2 has a different boundary: exponent-2 groups are abelian, and general class-2 2-groups require additional power-map or quadratic data. The explicit triangular cocycle presentations studied here form a specified subclass, but they and our automaticity theorem are valid for $p = 2$ as well.

**Algorithmic model theory.** Automatic structures [18, 3, 2, 9] present a structure by finite automata: elements are words, and each relation is recognised by a synchronous multi-tape automaton. Automaticity yields decidable first-order theories and linear-time query evaluation. For a *class* of finite structures the right notion is *uniform* automaticity with advice [27, 25]: one tuple of automata presents every member of the class, and an additional advice string, read synchronously with the element encodings, selects the member. Uniformly automatic classes admit algorithmic meta-theorems in the style of Courcelle's theorem [6, 8]: every first-order property can be compiled, once and for all, into an automaton that runs in linear time on each member's advice. Known uniformly automatic classes include all finite abelian groups, all finite Boolean algebras, and the graphs of bounded tree-depth [25]. On the group-theoretic side, automatic presentations remain close to commutativity: finitely generated word-automatic groups are virtually abelian [22], and arbitrary word-automatic groups are locally virtually abelian [21]. The recent positive result of Nies and Stephan, for odd $p$, concerns a *single* infinite structure—the infinite extraspecial $p$-group $E_p$ is word-automatic—while the relatively free class-2 exponent-$p$ group on infinitely many generators is not [20]. These are presentations of individual infinite groups; our concern is instead one regular advice language that presents an exact class of finite groups.

**From one group to a class, and from a matching to a tree.** The positive Nies–Stephan result already points toward a family. For odd $p$, $E_p$ is the direct limit of the finite extraspecial groups of order $p^{1+2m}$. Advice turns this limit into a family: a single automaton with *two* $\mathbb{F}_p$-registers, reading one advice letter per generator, multiplies in every finite extraspecial group at once (Proposition 3). The short streaming argument uses one register to buffer the digit just read and another to carry a running deficit. It prompts the question this paper answers: *what do the registers measure?*

A first hint that the answer is not "the number of active generators" comes from growing the center. Place one central generator at every leaf of a bushy binary shape tree, a generator pair at every inner node, and let each pair's commutator hit exactly the central generators below it. The center of this *tree-indexed* group grows linearly along such a family. Its commutation data is laminar rather than a matching, yet a single $\mathbb{F}_p$-register still suffices for a bottom-up automaton, provided sibling subtrees are forced to agree on the deficit they owe to the leaves beneath them (Proposition 4).

The answer, and the subject of this paper, is that in both warm-ups the commutation data crossing any cut of the layout has *rank one*. The registers never store digits; they store the image of the digits read so far under a rank factorisation of the *crossing block* at the current cut. What a streaming evaluator of the group law must carry across a cut is exactly this block, and if it factors through rank $r$, then $r$ residues suffice, whatever the number of active generators. Bounding the cut-rank over the cuts of a linear or tree layout is bounding the *rank-width of the commutation form*—the same step that graph theory takes from treewidth to rank-width and clique-width [15, 7], taken here inside group theory.

**Main result.** We prove one general theorem and specialise it. Fix a prime $p$, a depth $d \geq 1$ and a width $r$. A *site tree* is a binary tree whose nodes are generators ($x$-sites) or central generators ($z$-sites); a *cocycle tensor* over the chain ring $R = \mathbb{Z}/p^d$ assigns to each generator pair and each central target a coefficient in $R$, presenting a class-2 central extension with a distinguished central kernel $R^Z$. This kernel has exponent $p^d$ when $Z \neq \varnothing$, but it need not be the full center. For every subtree cut, the tensor entries crossing the cut sort into *six flattenings*— upward digit functionals on each side, resolved pair-sums, inward claims,

and two mixed kinds—and the *width* of the layout is the maximal *module rank* of these flattenings: the minimal number of generators of the spanned module, which over a field is the ordinary rank.

Our master theorem (Theorem 19) states that for fixed $p, d, r$ the pairs (site tree, tensor of width $\leq r$) form a uniformly tree-automatic class, with a multiplication automaton of $p^{O(rd)}$ states whose state *is* the cut interface. Chain-shaped layouts yield word-automatic presentations (Corollary 21). Placing the central generators on a chain above the layout gives fixed-central-kernel specialisations; we first prove their linear case as a fully worked model (Theorem 8). The node letters certify the finite module types and maps carried by these interfaces, and a finite tree automaton checks their local coherence. Thus the advice describes exactly the width-bounded tensors (Lemma 18).

Two algebraic obstacles separate the master theorem from its fixed-central-kernel analogue over a field. Their resolutions form the technical heart of the paper. First, the *ring*: over a field, the interface carried across a cut can be normalised to a basis, and the bilinear pairing of two sibling interfaces at a merge is again a matrix over the field. Over $\mathbb{Z}/p^d$ both steps fail, already at width one and depth two (Example 5). Normalised interfaces (bases of pure closures) are non-unique and need not restrict compatibly from one cut to the next. The natural interface—a generating set of the crossing *row module*—may carry valuation, in which case the merge value need not be bilinear in the two interface values. Over $\mathbb{Z}/4$, for example, the pairing $2ab$ of $w = 2a$ and $v = 2b$ is the well-defined function $2 \cdot (w/2)(v/2)$, but no $q$ satisfies $2ab = q\,wv$. The construction therefore carries row-module generators, whose restriction calculus is exact over any ring (Lemma 14), and lets merge letters carry bounded *pairing tables* rather than matrices (Lemma 15). Second, the *distributed central kernel*: commutator targets inside a subtree can be owed contributions by pairs resolved *outside* it, an inward flow that no bottom-up pass receives. The automaton meets it with a deterministic *claim-and-verify* protocol: the residuals of the central checks inside a subtree must lie in a bounded *claim module*, the automaton carries them as an element of that module, and determinism comes from a unique-extension property (Lemma 16)—over a field this is the familiar "commit to coordinates in a full-column-rank basis", but over the ring coordinates need not exist and the module formulation is the correct one. At $d = 1$ the claim protocol remains necessary, but coordinates suffice and the obstruction to matrix merges disappears. Both module-theoretic refinements are already forced over $\mathbb{Z}/4$.

**Contributions.**

- *The master theorem* (Theorem 19): for all fixed $p, d, r$, the class-2 central extensions with distinguished kernel over $\mathbb{Z}/p^d$ and a site-tree layout of module width $\leq r$ are uniformly tree-automatic; chain layouts give word-automaticity (Corollary 21). The fixed-kernel specialisations recover the exact word class of bounded linear rank-width over $R$; every bounded tree-cut-rank presentation has advice in the master class (Theorem 8, Corollary 22).
- *The ring calculus of interfaces*: module cut-rank as the correct width; the exact restriction calculus of row-module interfaces (Lemma 14); the failure of matrix merges over $\mathbb{Z}/p^d$ and their replacement by bounded pairing tables (Lemma 15); and claim modules with unique-extension determinism for distributed central targets (Lemma 16).
- *Exact advice*: finite algebraic types certify the six modules and their transition maps locally. Their regular language contains every width-$\leq r$ tensor and only such tensors (Lemma 18).
- *The wealth of the class* (Section 6): closure under direct products (width is the maximum,

not the sum, over blocks); low-width presentations of extraspecial groups, Heisenberg groups, tree-indexed laminar and point-target families, shared-target graph groups of bounded graph rank-width, and commutation graphs of bounded pathwidth, treewidth or vertex cover, all within one decidable family with modular advice. On the wide side, generic forms, ultraspecial groups and robust-complement Mekler families have width $\Omega(n)$; the relatively free group supplies a separate non-automatic boundary example.

- *Meta-theorems* (Section 7): compiled first-order model checking in time $O(\log |G|)$ per member, and decidability of the uniform theory of each exact regular presentation, including extensions by fixed modulo-counting quantifiers.
- *Implementation and validation* (Section 8): the fixed-kernel constructions and the general protocol over the ring are implemented in the open-source AutStr library [26]. Compiler assertions check the required factorisations on each generated instance, and retained regression tests compare the transition evaluators with the reference group law.

**Scope of the parameter.** The width bound imposes genuine structural restrictions. At $k = d = 1$ and odd $p$ the symplectic normal form collapses the fixed-kernel class, up to isomorphism, to central products of Heisenberg groups with abelian factors; bounded vertex cover collapses to a bounded core times an elementary abelian factor (Proposition 27). The isomorphism-theoretic content therefore begins at $k \geq 2$, where normal forms are unavailable, and already at width one for distributed central targets. The laminar and point-target classes of the master theorem contain bushy families with commutator subgroups of rank $\Theta(n)$. Conversely, a generic scalar presentation has cut-rank $\Omega(n)$ under *every* layout (Proposition 28); this is a statement about the sampled coordinates, not the isomorphism type of the presented group. More intrinsically, only $2^{O(m)}$ isomorphism types of order $p^m$ occur for fixed $p, d, r$ (Proposition 25). Thus the class is a structured, asymptotically negligible slice of the wild stratum.

**Organisation.** Section 2 fixes notation and develops the chain-ring algebra we need (valuation, minimal generating sets, module cut-rank). Section 3 proves the two warm-ups and isolates, informally, exactly what must generalise. Section 4 proves the fixed-kernel linear theorem over $R$ as a fully worked simple case. Section 5 defines site trees, tensors and the six flattenings, states the master theorem, and proves it; the string corollary and the fixed-kernel corollaries close the section. Section 6 maps the class: corners, closure properties, classical width measures, collapses, generic wideness, and the dividing line. Section 7 records the meta-theorems, Section 8 the implementation, validation scope and feasibility discussion, and Section 9 the open problems.

## 2 Preliminaries

**Words, trees, convolution.** $\Sigma$ denotes a finite alphabet and $\square \notin \Sigma$ a padding symbol. The *convolution* $w_1 \otimes \ldots \otimes w_m$ of words is the word over $(\Sigma \cup \{\square\})^m$ obtained by aligning the $w_i$ position-wise and padding the shorter ones. A *tree* is a finite, binary, rooted, labelled tree; the convolution of trees overlays them and pads missing positions. A (bottom-up deterministic) tree automaton processes a tree from the leaves to the root; a run assigns a state to every node as a function of the node's label and the states of its children.

**Uniformly automatic classes.** We follow [27]. A *uniformly automatic presentation* of a class $\mathcal{C}$ of $\tau$-structures consists of a regular advice set $A \subseteq \Sigma^*$ (or a regular set of advice

trees) and automata $\mathfrak{U}, (\mathfrak{A}_R)_{R\in\tau}$ such that for every $\alpha \in A$ the structure

$$\mathfrak{S}_\alpha \;=\; \big(\{u : \mathfrak{U} \text{ accepts } \alpha \otimes u\},\ \big(\{\bar{u} : \mathfrak{A}_R \text{ accepts } \alpha \otimes \bar{u}\}\big)_{R\in\tau}\big)$$

is well-defined, and $\mathcal{C} = \{\mathfrak{S}_\alpha : \alpha \in A\}$ up to isomorphism. The same structure may be presented by several advices. The fundamental theorem of automatic structures relativises: for every first-order formula $\varphi(\bar{x})$ there is an automaton recognising $\{\alpha \otimes \bar{u} : \mathfrak{S}_\alpha \models \varphi(\bar{u})\}$, effectively [27]. We present groups in the signature $\{M, \mathrm{eq}\}$ with $M$ the graph of multiplication.

**The chain ring and module rank.** Throughout, $R = \mathbb{Z}/p^d$ for a prime $p$ and a fixed depth $d \geq 1$; at $d = 1$, $R = \mathbb{F}_p$. $R$ is a finite chain ring: a local principal ideal ring whose ideals form the chain $R \supsetneq pR \supsetneq \cdots \supsetneq p^dR = 0$. Every element is $u\,p^s$ with $u$ a unit and $0 \leq s \leq d$ its *valuation* $v(\cdot)$; the units are exactly the valuation-0 elements. We write $\mathrm{rowsp}(V)$ for the row module of a matrix $V$ over $R$ (the $R$-span of its rows) and $\mathrm{colsp}(V)$ for its column module. For matrices $V, W$ with equally many columns, $W = TV$ is solvable for a matrix $T$ iff every row of $W$ lies in $\mathrm{rowsp}(V)$; we use this silently throughout.

▶ **Lemma 1** (module rank). *Every submodule $M \subseteq R^m$ has a generating set $p^{e_1}u_1, \ldots, p^{e_s}u_s$, where $u_1, \ldots, u_s$ extend to a basis of $R^m$, $0 \leq e_i < d$, and*

$$s = \mathrm{rk}_R(M) := \dim_{\mathbb{F}_p}\big(M/pM\big).$$

*No generating set of $M$ has fewer than $\mathrm{rk}_R(M)$ elements. Moreover:*

1. *if $N \subseteq M$, then $\mathrm{rk}_R(N) \leq \mathrm{rk}_R(M)$;*
2. *if $M \twoheadrightarrow Q$, then $\mathrm{rk}_R(Q) \leq \mathrm{rk}_R(M)$;*
3. *the row and column modules of a matrix over $R$ have the same module rank.*

**Proof.** Smith normal form over the principal ideal ring $R$: $M = \mathrm{rowsp}\big(\mathrm{diag}(p^{e_1}, \ldots)\,U\big)$ for a unimodular $U$, with the rows whose exponent satisfies $e_i \geq d$ vanishing and dropped; the surviving rows of $U$ are the $u_i$. Their number is $\dim_{\mathbb{F}_p} M/pM$, and Nakayama's lemma gives the lower bound for arbitrary generating sets. The invariant-factor decomposition identifies a finite $R$-module of module rank $s$ with a direct sum of $s$ nonzero cyclic modules. Submodules and quotients have at most $s$ cyclic factors, proving (1) and (2). Applying Smith normal form to a matrix shows that its nonzero invariant factors count both the minimal number of row generators and the minimal number of column generators, proving (3). ◀

We call $\mathrm{rk}_R(M)$ the *module rank* of $M$. At $d = 1$ it is the dimension of a subspace; the point of the definition for $d > 1$ is that a valuation-carrying generator still contributes a full unit of rank: over $\mathbb{Z}/4$ the module generated by $(2, 0)$ has module rank 1, although its reduction mod 2 is zero. Section 3 shows that this is exactly the right accounting for a streaming evaluator.

**Forms and groups over $R$.** Fix $k \geq 1$. A *form* of dimension $n$ over $R$ is a family $B = (B_{ji})_{1\leq i<j\leq n}$ with $B_{ji} \in R^k$; we write $\widehat{B}_{\{i,j\}} := B_{ji}$ for the unoriented label and treat absent labels as 0. Define the $R$-bilinear map $C\colon R^n \times R^n \to R^k$ by

$$C(a, a') \;=\; \sum_{1\leq i<j\leq n} B_{ji}\, a_j\, a'_i. \tag{1}$$

▶ **Lemma 2** (normal forms). *The set $R^k \times R^n$ with the operation*

$$(b, a) \cdot (b', a') \;=\; \big(b + b' + C(a, a'),\; a + a'\big)$$

*is a group $G(B)$ of order $p^{d(n+k)}$ and nilpotency class $\le 2$; the elements $(e_l, 0)$ are central, $[\,(0, e_j), (0, e_i)\,] = (\widehat{B}_{\{i,j\}}, 0)$ for $i < j$, and every element has a unique normal form $x_1^{a_1} \cdots x_n^{a_n}\, y_1^{b_1} \cdots y_k^{b_k}$ over the generators $x_i = (0, e_i)$, $y_l = (e_l, 0)$.*

**Proof.** Bilinearity of $C$ gives the cocycle identity

$$C(a, a') + C(a + a', a'') = C(a', a'') + C(a, a' + a''),$$

hence associativity; the inverse of $(b, a)$ is $(-b + C(a, a), -a)$. Since $B_{ii}$ is not among the labels, $C(e_i, e_i) = 0$, so $x_i$ has order $p^d$ exactly (also for $p = 2$); commutators are central, so the class is at most 2. The normal form is immediate from the displayed law. ◀

Write $K := R^k \times \{0\}$. This is the distinguished central kernel, or target subgroup, of the presentation; it has exponent $p^d$ when $k > 0$, but need not be the full center, since vectors in the radical of the commutator map also give central elements. Both coordinates range over $R$, and this is forced if commutators are to have order up to $p^d$: in a group of class $\le 2$, $[x, y]^p = [x^p, y]$, so if every generator had order $p$ the commutator subgroup would have exponent $p$ as well. For odd $p$ and $d = 1$, $G(B)$ has exponent $p$; for $p = 2$ the full center may contain elements of higher order even at $d = 1$: with $n = 3$, $k = 1$ and all three labels 1, the element $z = x_1 x_2 x_3$ is central with $z^2 = y$, and $G(B)$ is the 1-qubit Pauli group. Thus the exponent-$p^d$ assertion throughout concerns $K$, not necessarily $Z(G(B))$. For odd $p$ and $d = 1$, the Baer correspondence [1] identifies the isoclinism data of $G(B)$ with the pseudo-isometry class of the induced commutator map on the quotient by its radical and with target its image; after choosing target coordinates, this is the associated alternating-matrix-space problem [12]. The width parameters introduced next belong to the chosen coordinate presentation and layout; we do not claim that they are group-isomorphism or isoclinism invariants.

**Cut-rank and layouts.** For $S \subseteq [n]$ the *crossing block* of $B$ at $S$ is the matrix

$$M_S \in R^{\left(([n]\setminus S)\times[k]\right)\times S}, \qquad (M_S)_{(j,l),\,i} \;=\; \big(\widehat{B}_{\{i,j\}}\big)_l,$$

and the *module cut-rank* of $S$ is $\rho_B(S) := \mathrm{rk}_R\big(\mathrm{rowsp}(M_S)\big)$. A *linear layout* of $B$ is an ordering of $[n]$; renaming generators, we always take it to be the natural order, and define its width as $\max_{1 \le t < n} \rho_B([t])$. A *tree layout* is a binary rooted tree together with an injection of $[n]$ into its nodes. Generator-carrying nodes are enumerated in post-order; the remaining nodes are *structural* and carry a forced zero digit. We require the layout to be reduced: every structural node has two children, and each child subtree contains a generator. Thus there are fewer than $n$ structural nodes, and every subtree hosts an interval of the generator post-order. The width is $\max_v \rho_B(S_v)$ over the subtree generator sets $S_v$. We write $\mathrm{lcr}(B)$ and $\mathrm{tcr}(B)$ for the minimal width over all linear (tree) layouts; since a chain is a tree, $\mathrm{tcr} \le \mathrm{lcr}$. At $d = 1$ and $k = 1$ with 0/1 labels, $M_S$ is the bipartite adjacency matrix between $S$ and its complement in the commutation graph. Layouts in which every generator is a leaf coincide with $\mathbb{F}_p$ rank-decompositions; allowing generators at internal nodes can save at most one by Proposition 23. For fixed $p, d, k, r$ let $\mathcal{C}^{\mathrm{lin}}_{p,k,r,d}$ denote the class of pairs (form over $R$, linear layout of width $\le r$), each presenting $G(B)$, and $\mathcal{C}^{\mathrm{tree}}_{p,k,r,d}$ the tree analogue.

## 3 Warm-ups: two guiding automata

We give complete proofs of two elementary examples because they motivate the general construction and later reappear as corners of the master theorem.

**A: the uniform extraspecial class.** Fix a prime $p$. For $m \geq 1$ let $X_m$ be the *matching form*: $n = 2m$ generators, $k = 1$, and $B_{2t,2t-1} = 1$ for $t \in [m]$, all other labels zero. The presented group $G(X_m)$ of order $p^{1+2m}$ is, for odd $p$, the extraspecial group of exponent $p$; for $p = 2$, the central product of $m$ copies of $D_8$. For odd $p$, the infinite extraspecial group $E_p$ of Nies–Stephan is the direct limit of this family.

▶ **Proposition 3** (the finite extraspecial family, uniformly)**.** *The class* $\{G(X_m) : m \geq 1\}$ *is uniformly word-automatic, with advice* $\mathsf{c}\,\sigma^{2m}$ *for the* $m$*-th member and a multiplication automaton with* $2p^2 + 2$ *states.*

**Proof.** Encode $(b, a) \in \mathbb{F}_p \times \mathbb{F}_p^{2m}$ as the word $b\,a_1 \cdots a_{2m}$. By the cocycle normal form (Section 2), $z = x{\cdot}y$ holds iff $a_{z,i} = a_{x,i}{+}a_{y,i}$ position-wise and $b_z = b_x{+}b_y{+}\sum_{t=1}^{m} a_{x,2t}\,a_{y,2t-1}$. The automaton keeps a deficit $\delta \in \mathbb{F}_p$ and a buffer $w \in \mathbb{F}_p$. On the marker $\mathsf{c}$ it sets $\delta := b_z - b_x - b_y$. On each $\sigma$ it checks the digit sum; at odd positions $(2t{-}1)$ it stores $w := a_{y,2t-1}$, at even positions $(2t)$ it updates $\delta := \delta - a_{x,2t} \cdot w$. It accepts at the end iff $\delta = 0$. Parity is one bit of state; the state count is $2 \cdot p^2$ live states plus an initial and a sink state. Universe and equality automata check length parity and letter ranges only. ◀

The construction extends immediately to any *matching* form; what matters below is its shape: the buffer $w$ is a *one-dimensional interface*—the only information about the digits read so far that the future will consume.

**B: the laminar tree class.** Now let the distinguished central kernel grow. For a binary shape tree $T$, let $G_T$ be generated by a pair $x_w, y_w$ for every *inner* node $w$ of $T$ and a central $z_v$ for every *leaf* $v$, all of order $p$, subject to

$$[\,x_w,\, y_w\,] \;=\; \prod_{\text{leaves } v \text{ below } w} z_v,$$

all other pairs of generators commuting. The full center is the displayed $\mathbb{F}_p^{\text{leaves}}$: commuting with $y_w$ forces the coefficient of $x_w$ to vanish, and commuting with $x_w$ forces the coefficient of $y_w$ to vanish. The derived subgroup, however, is only the span of the descendant-leaf incidence vectors $\sum_{v \text{ below } w} z_v$ and may be proper (already a root with two leaves gives derived rank 1 and center rank 2). The commutation targets form the laminar family of the tree. Along bushy families with $|\,\text{leaves}(T)| = \Theta(|T|)$ the center grows linearly; a unary spine instead has a single central generator and recovers a $G(X_m)$-like fixed-kernel family. Thus the bushy families cannot lie in any family with a uniformly bounded central kernel.

▶ **Proposition 4** (the laminar tree family, uniformly)**.** *The class* $\{G_T : T$ *a binary shape tree*$\}$ *is uniformly tree-automatic, with the shape tree itself as advice and a multiplication automaton with* $p + 1$ *states.*

**Proof.** Encode an element by labelling inner nodes $w$ with the digit pair $(a_w, b_w)$ of $x_w, y_w$ and leaves $v$ with the central digit $c_v$; choose the local normal-form order $y_w, x_w$, so that $[x_w, y_w]$ has the displayed sign. The multiplication instance convolves three such labellings.

Writing the group law in normal form, $z = x \cdot y$ holds iff all digit sums match position-wise and, for every leaf $v$,

$$c_{z,v} - c_{x,v} - c_{y,v} \;=\; \sum_{w \text{ above } v} a_{x,w}\, b_{y,w}.$$

The right-hand side depends only on the root path of $v$, and—this is laminarity—it is the *same* sum for every leaf below a given node. So a bottom-up automaton can carry a single scalar: the deficit still owed to *all* leaves of the current subtree. At a leaf it starts as $c_{z,v} - c_{x,v} - c_{y,v}$; at an inner node $w$ it must equal the deficit of each child subtree—the automaton *rejects if the two children disagree*—and is decreased by $a_{x,w}b_{y,w}$, the contribution of the pair at $w$ to everything below. The automaton accepts at the root iff the owed deficit is 0. States: the $p$ values of the deficit plus a sink. Universe and equality automata check shapes and letter ranges. ◀

**What must generalise.** The two warm-ups look different—one streams left to right with a fixed central kernel, the other climbs a tree whose central kernel grows—but both automata carry rank-one data across every cut, and each isolates one of the two mechanisms of the master theorem.

In warm-up A the buffer is a *functional export*: the cut after position $2t-1$ is crossed by exactly one commutation pair, and $w$ is the value of the (rank-one) crossing block applied to the digits inside the cut. Replacing the matching by an arbitrary form, the crossing block $M_S$ at a cut $S$ becomes an arbitrary matrix; an automaton with $r$ registers can carry $V_S \cdot$ (digits in $S$) for an $r$-row matrix $V_S$ generating the row module of $M_S$, and the advice can spell out how these interfaces transform from one cut to the next. That programme, over the ring, is Section 4; on trees, where the two interfaces of sibling subtrees must be paired against each other at their merge, it is part of Section 5.

In warm-up B the deficit is a *claim*: leaves inside the subtree are owed contributions by pairs that live *above*—outside—the subtree, an inward flow that a bottom-up automaton cannot yet see. The automaton copes by committing to what the outside must eventually deliver: laminarity makes all leaves below a node interchangeable, so one scalar describes the debt, and the sibling-agreement check is the consistency of two commitments. In general the debts of the central positions inside a cut form a vector constrained to a bounded *claim module*—the column module of a claim flattening—and the automaton will carry that vector as an element of the module, deterministically, rejecting when a residual falls outside it; every merge will reconcile a child's claims against its sibling's exports. That is the claim-and-verify protocol of Section 5.

**Why the ring requires care.** Both mechanisms survive the passage from a distinguished kernel of exponent $p$ to one of exponent $p^d$—from $\mathbb{F}_p$ to $R = \mathbb{Z}/p^d$—but not by simply replacing $\mathbb{F}_p$ with $R$. Two phenomena, both already visible at width one over $\mathbb{Z}/4$, have no field analogue.

▶ **Example 5** (merges are not matrices over $\mathbb{Z}/4$)**.** Consider a single commutation pair split between two sibling subtrees, with label $2 \in \mathbb{Z}/4$, and suppose each subtree's interface is the functional $w = 2a$ (respectively $v = 2b$) of the digit $a$ (respectively $b$) of its endpoint. This generates the full crossing module $2R$, and the merge must add the contribution $2ab$. This value is completely determined by the two interface values, since $2ab = 2\,(w/2)(v/2)$, where $w/2$ denotes the well-defined residue $a \bmod 2$; but it is *not* a bilinear form in them: $q\,wv = 4qab = 0 \neq 2ab$ for every $q \in R$. Pairing two valuation-1 interfaces through a matrix

only ever produces valuation $\geq 2$; the valuation budget is spent. A merge over the ring must therefore be allowed to apply an arbitrary function of the two register values—a bounded *pairing table*—rather than a matrix (Lemma 15).

▶ **Example 6** (normalised interfaces do not restrict). One might hope to restore matrix merges by normalising the interface: by Lemma 1 every crossing module is contained in a free direct summand of the same module rank (a *pure closure*), and pairing free interfaces is unproblematic. But pure closures are not unique—in $(\mathbb{Z}/4)^3$, both $R(1,0,1)$ and $R(1,2,1)$ are minimal pure overmodules of $R(2,0,2)$—and they do not restrict into one another. For the form $B_{31} = 2$, $B_{32} = 1$, $B_{42} = 2$ over $\mathbb{Z}/4$, of width one, the crossing module at the cut $\{1,2,3\}$ is $R(0,2,0)$, with pure closure $R(0,1,0)$; restricting its generator to the first two coordinates gives $(0,1)$, which lies in no pure closure of the crossing module $R(2,1)$ at the cut $\{1,2\}$—indeed $R(2,1)$ is already pure, and $(0,1) \notin R(2,1)$. An automaton carrying the normalised functional $a_2$ mod 4 at the larger cut would need it delivered by an interface that can only ever supply $2a_1 + a_2$; no transition matrix exists. The row-module generator $(0,2,0)$, by contrast, restricts to $(0,2) = 2 \cdot (2,1)$, and the transition is the multiplication by 2.

The two examples fix the design: the interfaces carried across cuts are *generating sets of the crossing row modules*, for which restriction is exact over any ring (Lemma 14), and the merges consume them through tables (Lemma 15). The claims of warm-up B meet the same issue in a different guise—over the ring the residual vector inside a subtree has no canonical coordinates—and are carried as elements of a bounded *claim module*, with determinism supplied by a unique-extension property (Lemma 16). With these adjustments, both warm-up mechanisms extend to the general setting.

## 4 Linear layouts over the chain ring

Throughout this section fix $p, d, k, r$ and a form $B$ over $R$ with a linear layout of width $\leq r$; write $M_t := M_{[t]}$.

The multiplication automaton must verify $(b_z, a_z) = (b_x, a_x) \cdot (b_y, a_y)$ while reading the three encodings synchronously, digit by digit. The digit sums $a_{z,t} = a_{x,t} + a_{y,t}$ are position-wise; the work is the correction $C(a_x, a_y) = \sum_t a_{x,t}\big(\sum_{i<t} B_{ti}\, a_{y,i}\big)$: at position $t$ the automaton must know the vector $\sum_{i<t} B_{ti}\, a_{y,i} \in R^k$, a linear functional of all digits read so far. It cannot store the digits; it stores their image under the interface of the current cut. By Example 6, the interface must be a generating set of the crossing row module itself:

▶ **Lemma 7** (streaming interfaces). *For $0 \leq t \leq n$ fix $V_t \in R^{r\times t}$ whose rows generate* $\mathrm{rowsp}(M_t)$ *(possible by Lemma 1, padding with zero rows; $V_0$ is empty). Then for every $t \in [n]$ there are $P_t \in R^{r\times r}$, $v_t \in R^r$, $R_t \in R^{k\times r}$ with*

1. $V_t = \left[\, P_t V_{t-1} \;\middle|\; v_t \,\right]$, *i.e. the first $t-1$ columns of $V_t$ equal $P_t V_{t-1}$ and the last column is $v_t$;*
2. $R_t V_{t-1} = B^{(t)}$, *where $B^{(t)} \in R^{k\times(t-1)}$ has columns $B_{ti}$ for $i < t$.*

**Proof.** (1) Each row of $M_t$ is indexed by some $(j,l)$ with $j > t$; its restriction to the first $t-1$ columns is the corresponding row of $M_{t-1}$, because the same generator $j$ also lies outside $[t-1]$. Restriction is $R$-linear, so every row of $V_t$—an $R$-combination of rows of $M_t$—restricts to an $R$-combination of rows of $M_{t-1}$, i.e. into $\mathrm{rowsp}(M_{t-1}) = \mathrm{rowsp}(V_{t-1})$; $P_t$ collects the coefficients, and $v_t$ is the last column. (2) The $l$-th row of $B^{(t)}$ is the row $(t,l)$ of $M_{t-1}$, because $t \notin [t-1]$; hence it lies in $\mathrm{rowsp}(V_{t-1})$. ◀

The proof is verbatim the field argument, and this is the point: the restriction calculus of row modules is exact over any commutative ring. What fails over the ring is only the normalisation of the interface (Example 6), which the lemma never performs. Merges do not occur on a chain, so the obstruction of Example 5 first arises in Section 5.

▶ **Theorem 8** (linear layouts). *For all fixed $p, d, k, r$, the class $\mathcal{C}^{\mathrm{lin}}_{p,k,r,d}$ is uniformly word-automatic. The advice for $(B, \mathit{layout})$ has length $n + k$ over $p^{d(r^2+r+kr)} + 1$ advice letters (the presentation's base alphabet adds the $p^d$ element digits and the padding symbol); the multiplication automaton has $p^{d(k+r)} + \sum_{j<k} p^{dj} + 1$ states; and every well-formed advice presents a member of the class.*

**Proof.** *Encoding.* An element $(b, a)$ is the word $b_1 \cdots b_k a_1 \cdots a_n$ over the digit alphabet $\{0, \ldots, p^d - 1\}$ representing $R$. The advice is $\mathsf{c}^k \alpha_1 \cdots \alpha_n$ where $\alpha_t$ encodes the triple $(P_t, v_t, R_t)$ of Lemma 7 as $r^2 + r + kr$ ring entries.

*The automaton.* States are $(\mathsf{c}, j, \delta)$ with $j < k$, $\delta \in R^j \times \{0\}^{k-j}$, and $(\mathsf{x}, \delta, w)$ with $\delta \in R^k$, $w \in R^r$, plus a sink. On the $j$-th center position it records $\delta_j := b_{z,j} - b_{x,j} - b_{y,j}$. On position $t$ it checks $a_{x,t} + a_{y,t} = a_{z,t}$, updates

$$\delta \;\leftarrow\; \delta - a_{x,t} \cdot (R_t\, w), \qquad w \;\leftarrow\; P_t\, w + v_t \cdot a_{y,t},$$

and accepts at the end iff $\delta = 0$.

*Correctness.* By induction, after position $t$ the invariant $w = V_t\,(a_y)|_{[t]}$ holds: it is preserved by Lemma 7(1). Hence at position $t$, $R_t w = R_t V_{t-1}(a_y)|_{[t-1]} = \sum_{i<t} B_{ti} a_{y,i}$ by Lemma 7(2), so the accumulated subtraction is exactly $C(a_x, a_y)$ of (1), and acceptance means $b_z = b_x + b_y + C(a_x, a_y)$ as required by Lemma 2. The universe and equality automata only check the shape.

*Exactness.* Conversely, an arbitrary well-formed advice defines, through the same update equations, the correction $\tilde{C}(a, a') = \sum_{i<t} \big(R_t\, P_{t-1} \cdots P_{i+1}\, v_i\big)\, a_t\, a'_i$, which is $R$-bilinear and supported on pairs $i < t$; so it equals (1) for the form $B_{ti} := R_t P_{t-1} \cdots P_{i+1} v_i$, and the advice presents $G(B)$ for that form. Its layout has width $\le r$: the crossing block $M_t$ of the streamed form factors, by the same telescoping, as $A_t \cdot W_t$ where $W_t \in R^{r \times t}$ has columns $P_t \cdots P_{i+1} v_i$, so $\mathrm{rowsp}(M_t) \subseteq \mathrm{rowsp}(W_t)$ is generated by $r$ elements and $\mathrm{rk}_R\big(\mathrm{rowsp}(M_t)\big) \le r$ by Lemma 1. ◀

▶ Remark 9. The encoding has the expected size: an element of $G$ has length $n+k = \log_{p^d} |G|$, and for fixed $\varphi$ the model-checking run (Corollary 31) is linear in $\log |G|$—not in $|G|$.

▶ **Example 10** (the clique). Let $B_{ji} = 1 \in R^1$ for all $i < j$: no two generators commute, and the commutation graph is complete, with pathwidth $n-1$. Every crossing block is an all-ones matrix, of module rank 1; the layout data can be taken as $P_t = (1)$, $v_t = (1)$, $R_t = (1)$, and the automaton degenerates to two registers: $w = \sum_{i \le t} a_{y,i}$ and the deficit $\delta$. Under any vertex-based width measure this is the hardest instance; under cut-rank it is the easiest.

▶ **Example 11** (extraspecial groups, and valuation-carrying labels). Let $n = 2m$, $k = 1$, $d = 1$, and $B_{2t,2t-1} = 1$ for $t \in [m]$: a perfect matching, presenting the extraspecial-type group of order $p^{1+2m}$, and Theorem 8 specialises to the uniform presentation of Proposition 3. Over the ring the same matching with labels $p^s$, $0 \le s < d$, presents central extensions whose commutators have order $p^{d-s}$; the module cut-rank of every cut is still 1—the width charges a valuation-carrying label fully, as it must, since the automaton genuinely transports the functional $p^s a_{y,2t-1}$ across the cut.

## 5 The master theorem: distributed central targets over the ring

Theorem 8 fixes the central-kernel dimension and streams all central digits at one end of the layout, so every correction flows upward. This section removes both restrictions at once: the central generators live on the layout tree themselves, with commutator targets distributed arbitrarily, and everything happens over $R = \mathbb{Z}/p^d$.

**Site trees and cocycle tensors.** A *site tree* is a reduced binary tree whose nodes are $x$-sites (generators), $z$-sites (central generators), or structural nodes. Every structural node has two children, each containing an $x$- or $z$-site; $X$ and $Z$ denote the $x$- and $z$-sites in post-order. A *cocycle tensor* over $R$ is a family $T = (T_{jiv})$ with $T_{jiv} \in R$, indexed by $i < j$ in $X$ and $v \in Z$. It presents $G(T)$: the central extension of $R^X$ by the distinguished central kernel $R^Z$ with correction $C(a, a')_v = \sum_{i<j} T_{jiv}\, a_j a'_i$, to which Lemma 2 applies verbatim. Elements $(b, a)$ are digit labellings of the site tree, and a multiplication instance convolves three of them; the check at a $z$-site $v$ is $(b_z - b_x - b_y)_v = C(a_x, a_y)_v$, while $x$-sites carry position-wise digit sums.

▶ **Definition 12** (crossing flattenings, width). *Let $S$ be a subtree (a post-order interval). Organise the entries $T_{jiv}$ with at least one but not all sites in $S$ into the following six, partly overlapping, matrices:*

$$
\begin{aligned}
F_y &: \text{rows } i \in S,\ \text{cols } (j,v) \notin S && \text{(y-digit functionals, upward)}\\
F_x &: \text{rows } j \in S,\ \text{cols } (i,v) \notin S && \text{(x-digit functionals, upward)}\\
F_m &: \text{rows } (j,i) \subseteq S,\ \text{cols } v \notin S && \text{(resolved pair-sums, upward)}\\
F_g &: \text{rows } v \in S,\ \text{cols } (j,i) \not\subseteq S && \text{(claims: expected inward flow)}\\
F_{py} &: \text{rows } i \in S,\ \text{cols } (j \notin S, v \in S) && \text{(mixed: inside y-coefficients of future x-digits)}\\
F_{px} &: \text{rows } j \in S,\ \text{cols } (i \notin S, v \in S) && \text{(mixed, x-side).}
\end{aligned}
$$

*The* width *of the cut is the maximum of the six module ranks, and the width of the layout is the maximum over all subtree cuts. For fixed $p, d, r$ let $\mathcal{C}^{\mathrm{site}}_{p,d,r}$ be the class of pairs (site tree, tensor of width $\le r$), each presenting $G(T)$.*

By Lemma 1, row and column modules of a matrix have the same module rank, so the width bounds either side of each flattening. The six matrices are genuinely different: reshaping a flattening changes its rank. For the endpoint-indexed tensor $T_{jiv} = [\, v = \ell(i) \,]$ the flattening $[(i,v)] \times [j]$ has rank one while $[i] \times [(j,v)]$ has full rank, and conversely for other tensors; each of the six corresponds to a distinct kind of traffic a bottom-up evaluator must carry, so the width must bound each of them, and Example 13(iv) below shows the inward kind is genuinely charged.

▶ **Example 13** (the corners, quantitatively). (i) With all $z$-sites on a chain above the root, $F_g, F_{py}, F_{px}$ are empty at every layout cut and $\mathrm{rk}_R(F_m) \le k$: a form of tree cut-rank $r$ in the sense of Section 2 has width $\le \max(r, k)$. The split of $M_S$ into $F_y$ and $F_x$ by the orientation of the crossing pair requires care: zeroing a triangle of an all-ones matrix can increase its rank without bound. Here, however, the split partitions the *rows* of $M_S$: since $S$ is a post-order interval, an outside index above it is the larger endpoint of every pair it crosses into $S$, one below it the smaller, so each row belongs entirely to $F_y$ or to $F_x$. (ii) Encode the laminar tree-indexed groups of Proposition 4 as site trees: each inner node $w$ of the shape becomes a chain of two $x$-sites ($x_w$ directly above $y_w$), each leaf a $z$-site, and $T_{x_w y_w v} = 1$ for every

leaf $v$ below $w$. A subtree cut never separates a generator pair except along its own chain edge, and every target lies below its pair, so $F_y = F_x = F_m = 0$; every ancestor pair owes every central leaf below the cut equally, so each claim flattening is an all-ones matrix: width 1. (iii) *Point targets*: as in (ii) but each pair hits only the leftmost leaf below it. Width 1, growing central kernel, non-laminar law: as a site-tensor presentation, this family belongs to neither the fixed-kernel corner of Theorem 8 nor the laminar presentation above. (iv) *Scattered targets*: place $m$ private central generators on a chain below a chain of $m$ otherwise mutually commuting generator pairs, with pair $t$ targeting $z_t$. Every flattening has at most $m$ rows or columns, while the claim flattening at the cut containing the $z$-chain is the $m \times m$ identity. This displayed layout therefore has width exactly $m$.

**The protocol.** At site boundaries the macro-automaton carries six $R^r$-registers; throughout, "interface" means an advice-chosen generating tuple of the indicated module (Lemma 1), and elements of the two bounded modules below are stored as representatives with respect to such a tuple, canonicalised by the advice.

- $w^y = V_S^y(a_y)|_{S\cap X}$ and $w^x = V_S^x(a_x)|_{S\cap X}$, where the rows of $V_S^y$ generate the *column module* of $F_y(S)$—each column, indexed by an outside consumer, is a functional of the inside digits—and likewise $V_S^x$ for $F_x$;
- their mixed counterparts $\varphi^y = V_S^{py}(a_y)|_{S\cap X}$ and $\varphi^x = V_S^{px}(a_x)|_{S\cap X}$ for $F_{py}, F_{px}$: inside coefficients that a future outside digit will multiply, the product flowing back *into* the subtree's checks;
- $m$, a representative of the *exports*

$$\mu_S \;:=\; \Big(\textstyle\sum_{(j,i)\subseteq S} T_{jiv}\, a_{x,j}\, a_{y,i}\Big)_{v\notin S} \;\in\; E_S := \mathrm{rowsp}(F_m(S)),$$

  the contribution vector that pairs resolved inside $S$ owe to targets outside;
- $g$, a representative of the *residuals*

$$\delta_S \;:=\; \Big((b_z - b_x - b_y)_v - \textstyle\sum_{(j,i)\subseteq S} T_{jiv}\, a_{x,j}\, a_{y,i}\Big)_{v\in S\cap Z},$$

  which the run requires to lie in the *claim module* $\Gamma_S := \mathrm{colsp}(F_g(S))$, rejecting otherwise: the outside can only ever deliver elements of $\Gamma_S$, since each future pair contributes an $R$-multiple of its column of $F_g(S)$.

$E_S$ and $\Gamma_S$ have module rank $\le r$, so both have at most $p^{dr}$ elements and representatives in $R^r$; the committed state space is $R^{6r}$ plus a sink. A streamed implementation of one macro-transition uses additional temporary banks, counted below.

Three lemmas supply the transition data. The first is the restriction calculus, one membership argument instantiated per flattening pair.

▶ **Lemma 14** (restriction calculus)**.** *Let $u$ be a node with subtree $S$ and let $D$ index a child subtree (or the unique child at a unary node). Then:*

1. folds*: a column of $F_y(S)$, restricted to the rows in $D$, is a column of $F_y(D)$; a column of $F_{py}(S)$ restricted to $D$ is a column of $F_{py}(D)$ if its target lies in $D$, and a column of $F_y(D)$ otherwise; symmetrically on the $x$-side;*
2. read-offs*: for an $x$-site $u$ and $i$ ranging over $D$, the coefficient vector $(T_{uiv})_{i\in D}$ is, for each fixed target $v$, a column of $F_y(D)$ if $v \notin D$ and a column of $F_{py}(D)$ if $v \in D$; and for each fixed $i$, the target vector $(T_{uiv})_{v\in D\cap Z}$ is a column of $F_g(D)$, while $(T_{uiv})_{v\notin S}$ is a row of $F_m(S)$;*

3. exports*: restricting a row of $F_m(D)$ to the targets outside $S$ gives the corresponding row of $F_m(S)$; hence coordinate restriction maps $E_D$ into $E_S$. Restricting it instead to the targets in the sibling $D'$ gives a column of $F_g(D')$; hence coordinate restriction maps $E_D$ into $\Gamma_{D'}$;*
4. claims*: a column of $F_g(S)$, restricted to the $z$-rows in $D$, is a column of $F_g(D)$; hence coordinate restriction maps $\Gamma_S$ into $\Gamma_D$.*

*Consequently all fold and rebase systems below are solvable over $R$: whenever a carried tuple must be re-expressed over a target interface, the relevant vectors lie in the target module.*

**Proof.** Each assertion follows from the same membership argument. For (1), the consumer $(j, v)$ of a column of $F_y(S)$ satisfies $j, v \notin S \supseteq D$, so the pair indexes a column of $F_y(D)$; for $F_{py}(S)$ the consumer has $j \notin S$ and $v \in S$, and restricting to $D$ the target either stays inside $(F_{py}(D))$ or falls outside $D$ $(F_y(D))$. (2): $u \notin D$, so $(u, v)$ is an admissible column index of the child flattening in each case, and $(u, i) \subseteq S$ with $v \notin S$ indexes a row of $F_m(S)$. (3): a pair inside $D$ is inside $S$, and its targets outside $S$ (resp. in $D'$) index the same entries in $F_m(S)$ (resp. a column of $F_g(D')$, since the pair is not inside $D'$). (4): a pair not inside $S$ is not inside $D$. The final sentence is the definition of the modules as spans. ◀

The second lemma addresses Example 5: the merge of two sibling interfaces is a well-defined pairing, but not a matrix.

▶ **Lemma 15** (pairing tables)**.** *Let $u$ be a binary node with children $D_0, D_1$, and consider the split pairs $(i \in D_0 \cap X,\ j \in D_1 \cap X)$, sorted by target into blocks with $v \notin S_u$, $v \in D_0$, $v \in D_1$, and, when $u$ is a $z$-site, $v = u$. Then the total contribution of each block,*

$$\left(\sum_{i,j} T_{jiv}\, a_{x,j}\, a_{y,i}\right)_v,$$

*is a well-defined function of a pair of register values—of $(w^x_{D_1}, w^y_{D_0})$ for the first and fourth blocks, $(w^x_{D_1}, \varphi^y_{D_0})$ for the second, and $(\varphi^x_{D_1}, w^y_{D_0})$ for the third. As an element-valued map, the first three land in $E_{S_u}$, $\Gamma_{D_0}$ and $\Gamma_{D_1}$, respectively; the fourth is a scalar used in the check at $u$. Each map is $R$-bilinear on the images of the two interfaces. In general it does not extend to a bilinear form on $R^r \times R^r$ (Example 5); the advice therefore carries it as a table with $p^{2rd}$ entries. For $d = 1$ every such map extends to a bilinear form, and the table may be replaced by $r \times r$ matrices over $\mathbb{F}_p$.*

**Proof.** Fix a target $v$ of the first block and let $X_v[j, i] = T_{jiv}$. For fixed $j$, the vector $(T_{jiv})_{i \in D_0}$ is the column $(j, v)$ of $F_y(D_0)$—both $j$ and $v$ lie outside $D_0$—so every row of $X_v$ lies in $\mathrm{rowsp}(V^y_{D_0})$, i.e. $X_v = A_v V^y_{D_0}$ for some $A_v$. Hence $X_v a = A_v(V^y_{D_0} a)$ depends on $a := (a_y)|_{D_0}$ only through $w^y_{D_0}$, and symmetrically—the columns of $X_v$ being columns of $F_x(D_1)$—the value $b^{\mathsf{T}} X_v a$ depends on $b := (a_x)|_{D_1}$ only through $w^x_{D_1}$: the contribution is a well-defined function of $(w^x_{D_1}, w^y_{D_0})$, visibly $R$-bilinear on the images. The target vector of each split pair, $(T_{jiv})_{v \notin S_u}$, is a row of $F_m(S_u)$, so the element-valued map lands in $E_{S_u}$. For the second block replace $F_y(D_0)$ by $F_{py}(D_0)$ (the target lies in $D_0$) and note the target vectors $(T_{jiv})_{v \in D_0 \cap Z}$ are columns of $F_g(D_0)$; the third block is symmetric. For the fourth block, $u$ lies outside both children, so the first-block argument applies verbatim, with the single target coordinate retained as a scalar. Over a field, a bilinear map defined on subspaces extends to the ambient spaces by completing bases, which yields the matrices; Example 5 shows a map with no such extension over $\mathbb{Z}/4$. ◀

The third lemma packages the claim discipline. Elements of $\Gamma_S$ are actual vectors, so the bookkeeping that over a field is done with coordinates can be done with the elements themselves.

▶ **Lemma 16** (claim modules). $\mathrm{rk}_R(\Gamma_S) \le r$ *for every subtree* $S$*, and:*

1. partial extension and contraction*: suppose* $S'$ *is obtained from a child subtree* $S$ *by adding one parent site. If the parent is an* $x$*-site, then* $S' \cap Z = S \cap Z$ *and* $\Gamma_{S'} \subseteq \Gamma_S$*; hence the inverse of the inclusion is a partial map that is defined exactly when the updated residual belongs to* $\Gamma_{S'}$*. If the parent is a* $z$*-site* $v$*, then an element of* $\Gamma_{S'}$ *is determined by its restriction to* $S \cap Z$ *together with its coordinate at* $v$*. At a binary node, an element of* $\Gamma_{S_u}$ *is determined by its restrictions to the two children and, when* $u$ *is a* $z$*-site, its coordinate at* $u$*. Thus claim extension, contraction and sibling joins are partial functions defined exactly on consistent inputs;*
2. discharge*: subtracting from* $g$ *any* $R$*-combination of columns of* $F_g(S)$ *stays in* $\Gamma_S$*;*
3. restriction*: by Lemma 14(4), the restriction of a claim to a child is a claim of the child, so the extensions and joins of (1) are compatible with the invariants below.*

*At* $d = 1$*, choosing a full-column-rank submatrix of* $F_g(S)$ *turns* $\Gamma_S$ *into coordinates* $\mathbb{F}_p^{\mathrm{rk}_R}$ *and (1) into the familiar protocol: commit to the unique coordinates of the residual, reject if the system is inconsistent, and check that sibling subtrees agree.*

**Proof.** The rank bound is the width. If the added parent is an $x$-site, the rows of the two claim flattenings have the same index set, while every column of $F_g(S')$ is also a column of $F_g(S)$; hence $\Gamma_{S'} \subseteq \Gamma_S$. In the other cases the stated restrictions and the possible new coordinate list every coordinate of the vector, so the maps are injective. This proves (1). Parts (2) and (3) are closure of a module under subtraction and Lemma 14(4). ◀

The advice must certify these algebraic interfaces, rather than merely have the right syntactic shape. We first specify exactly what a node letter contains. For a proposition $P$, write $R^{[P]}$ for $R$ if $P$ holds and for 0 otherwise.

▶ **Definition 17** (certified local type). *At every node* $u$*, a certified local type declares six finite* $r$*-generated* $R$*-modules*

$$H_u^y,\ H_u^x,\ H_u^{py},\ H_u^{px},\ E_u,\ G_u,$$

*each with a fixed surjection from* $R^r$ *and a canonical representative for every element. It records the node kind and the following maps, with one copy for each child* $D_q$*.*

1. *For each* $\epsilon \in \{x, y\}$ *and* $h \in \{\mathrm{plain}, \mathrm{mix}\}$*, an* $R$*-linear functional-update map*

$$\bigoplus_q (H_{D_q}^{\epsilon} \oplus H_{D_q}^{p\epsilon}) \oplus R^{[u \in X]} \longrightarrow H_u^{h\epsilon}.$$

   *These are the folds and digit injections for the four functional registers; the last summand receives the digit* $\xi_u^{\epsilon}$ *from input* $\epsilon$*.*
2. *For each child, linear export maps*

$$q_q^{\mathrm{out}} : E_{D_q} \to E_u, \qquad q_q^{\mathrm{sib}} : E_{D_q} \to G_{D_{1-q}}, \qquad o_q : E_{D_q} \to R^{[u \in Z]},$$

   *omitting the sibling map at a unary node.*
3. *At a binary node, balanced biadditive split-pair maps*

$$P_{\mathrm{out}} : H_{D_1}^x \times H_{D_0}^y \to E_u, \qquad P_0 : H_{D_1}^x \times H_{D_0}^{py} \to G_{D_0},$$
$$P_1 : H_{D_1}^{px} \times H_{D_0}^y \to G_{D_1}, \qquad P_u : H_{D_1}^x \times H_{D_0}^y \to R^{[u \in Z]}.$$

4. *If $u \in X$, linear parent–descendant read-offs*

$$R_q^{\mathrm{out}} : H^y_{D_q} \to E_u, \qquad R_{q,t} : K_{q,t} \to G_{D_t}$$

*for every existing child $D_q$ and target child $D_t$. In the second display, $K_{q,t} := H^{py}_{D_q}$ when $q = t$ and $K_{q,t} := H^y_{D_q}$ otherwise.*

5. *An injective claim map*

$$j_u : G_u \longrightarrow \bigoplus_q G_{D_q} \oplus R^{[u \in Z]},$$

*together with its partial inverse on its image.*

*At a leaf, absent child modules and maps are omitted; moreover $E_u = 0$, and $G_u = 0$ unless $u$ is a $z$-site. At the root all six modules are zero. Child-module names in a parent letter must equal the module names exposed by the child.*

*Here* balanced biadditive *means additive in both arguments and $P(ra, b) = rP(a, b) = P(a, rb)$ for $r \in R$. The transition equations are part of the definition. Write $e_q, g_q$ for child exports and claims, $h_q^\bullet$ for their functional values, and $\xi_u^x, \xi_u^y$ for the two input digits at an $x$-site. Put $p_0 = P_0(h_1^x, h_0^{py})$ and $p_1 = P_1(h_1^{px}, h_0^y)$. Missing terms are zero. The new export and the temporary child claims are*

$$e_u = \sum_q q_q^{\mathrm{out}}(e_q) + P_{\mathrm{out}}(h_1^x, h_0^y) + [u \in X]\, \xi_u^x \sum_q R_q^{\mathrm{out}}(h_q^y), \tag{2}$$

$$\widetilde{g}_t = g_t - q_{1-t}^{\mathrm{sib}}(e_{1-t}) - p_t - [u \in X]\, \xi_u^x \sum_q R_{q,t}(k_{q,t}), \tag{3}$$

*where $k_{q,t}$ is the current value in $K_{q,t}$. If $u$ is a $z$-site, its new coordinate is*

$$d_u - \sum_q o_q(e_q) - P_u(h_1^x, h_0^y), \tag{4}$$

*where $d_u = b_{z,u} - b_{x,u} - b_{y,u}$. The new claim is the partial inverse of $j_u$ applied to the tuple of the $\widetilde{g}_t$ and, when present, the coordinate in (4); an undefined inverse means rejection. The four functional outputs are the values of the maps in (1). Equations (2)–(4), together with linearity, balanced biadditivity, injectivity of $j_u$, the boundary conditions and matching child types, are the complete local certificate conditions.*

▶ **Lemma 18** (regular certified macro-advice). *For fixed $p, d, r$ there are a finite alphabet $\Lambda_{p,d,r}$ and a regular language $A^{\mathrm{cert}}_{p,d,r}$ of $\Lambda_{p,d,r}$-labelled binary trees with the following properties.*

1. *Every site tensor $T$ of layout width at most $r$ effectively yields an advice $\alpha(T) \in A^{\mathrm{cert}}_{p,d,r}$.*
2. *Every $\alpha \in A^{\mathrm{cert}}_{p,d,r}$ determines a site tree and a unique triangular $R$-bilinear cocycle tensor $T_\alpha$ on it, of layout width at most $r$.*
3. *On $\alpha$, the six-register macro-protocol recognises exactly the multiplication graph of $G(T_\alpha)$.*

*The same statements hold after replacing each macro-letter by a fixed $O_{p,d,r}(1)$-node instruction gadget over the streamed alphabet.*

**Proof.** *Regularity.* A macro-letter is the finite structure in Definition 17. There are only finitely many such letters. An $r$-generated module is a quotient of $R^r$ and hence has at most $p^{dr}$ elements; for fixed $p, d, r$ there are only finitely many module structures, maps and tables on these carriers. Whether a node letter has the child types it declares, whether its maps are homomorphisms or balanced biadditive, and whether $j_u$ is injective with the stated partial inverse are finite checks. A bottom-up tree automaton stores the exposed six module types and performs these checks, so $A^{\mathrm{cert}}_{p,d,r}$ is regular.

*Completeness.* Given $T$, take at every cut the actual six modules of Definition 12, choose generating tuples padded to length $r$, and attach the maps and tables constructed in Lemma 14, Lemma 15 and Lemma 16. Partitioning contributions by their source and target location gives equations (2)–(4), so the resulting advice is certified.

*The reconstructed tensor.* Conversely, read the certified maps as an arithmetic circuit. Each input digit enters through a linear digit map, and every correction term is obtained either by multiplying a linear read-off by the other input digit or by a biadditive pairing, and is then routed linearly to one $z$-site. The circuit therefore computes an $R$-bilinear map $C_\alpha\colon R^X \times R^X \to R^Z$. To see that no pair is omitted or duplicated, take two distinct $x$-sites. If they are comparable in the tree order, their pair is introduced by the parent–descendant read-off at the upper site. Otherwise their least common ancestor has them in different child subtrees, and its split-pair table introduces the pair. These cases are disjoint and exhaustive. The upper site or least common ancestor follows both endpoints in post-order, so $C_\alpha$ is triangular with zero diagonal. Its basis values define a unique tensor $T_\alpha$.

*Width.* We give the factorisation argument at an arbitrary subtree $S$. First, the injective maps $j_u$ recursively embed every abstract claim module into its coordinate space:

$$\gamma_S : G_S \hookrightarrow R^{S\cap Z}.$$

An export element similarly determines a concrete vector outside $S$. Define $\varepsilon_S : E_S \to R^{Z\setminus S}$ downwards from the root. If $S = D_q$ is a child of $S_u$, the three disjoint coordinate blocks $Z \setminus S_u$, $D_{1-q} \cap Z$, and $\{u\} \cap Z$ receive, respectively,

$$\varepsilon_{S_u}(q_q^{\mathrm{out}}(e)), \qquad \gamma_{D_{1-q}}(q_q^{\mathrm{sib}}(e)), \qquad o_q(e),$$

with absent blocks omitted. These values uniquely define $\varepsilon_{D_q}(e)$; at the root the codomain is zero. Every pair resolved inside $S$ is routed through $E_S$, so every row of $F_m(S)$ lies in $\operatorname{im}\varepsilon_S$. Every pair resolved outside $S$ is routed into $S$ through $G_S$, so every column of $F_g(S)$ lies in $\operatorname{im}\gamma_S$.

For each functional type, the update maps recursively define a linear digit map $D_S^\bullet : R^{S\cap X} \to H_S^\bullet$. Any interaction whose one endpoint lies in $S$ is resolved above the cut using only the corresponding value $D_S^\bullet(a|_S)$. Hence every functional in the associated crossing flattening is a precomposition $\lambda \circ D_S^\bullet$ for some $\lambda \in \operatorname{Hom}_R(H_S^\bullet, R)$. If an $r$-generated module is written as a sum of at most $r$ cyclic modules, its dual has at most $r$ cyclic summands. Thus all six flattening modules are submodules or homomorphic images of $r$-generated modules and have module rank at most $r$ by Lemma 1.

*Protocol correctness.* For the true product defined by $C_\alpha$, the residual at a subtree is the sum of precisely those contributions resolved outside it. The circuit routes every such contribution through $G_S$. Hence the residual lies in $\operatorname{im}\gamma_S$, and at each node its child restrictions and possible new coordinate lie in the image of $j_u$. Every required partial inverse is therefore defined. Conversely, equations (2)–(4) show inductively that an accepting run has accounted for every pair and target exactly once. The macro-protocol therefore accepts precisely

$$(b, a) \cdot (b', a') = (b + b' + C_\alpha(a, a'), a + a'),$$

which is $G(T_\alpha)$ by Lemma 2.

*Streamed encoding.* Finally, encode each macro-letter by a fixed unary instruction gadget carrying one ring entry at a time. A finite tree automaton checks these finitely many codes and their attachment points, so this fixed expansion preserves regularity and all three properties. ◀

▶ **Theorem 19** (master theorem: distributed central targets over $\mathbb{Z}/p^d$)**.** *For all fixed $p$, $d$ and $r$, the class $\mathcal{C}^{\mathrm{site}}_{p,d,r}$ is uniformly tree-automatic, with exact regular advice set $A^{\mathrm{cert}}_{p,d,r}$. With one advice letter containing the complete bounded transition data of a site, the multiplication automaton has $p^{6rd}+1$ states. Alternatively, the advice may expand each site into $O_{p,d,r}(1)$ letters carrying one ring entry each, over an alphabet of $p^d + O(1)$ letters; the resulting streamed automaton has $p^{O(rd)}$ states.*

**Proof.** Use the certified advice language of Lemma 18. In the macro-letter presentation, the advice at each node carries the transition data of Lemmas 14–16 for its subtree: the fold matrices and injection columns for the four functional interfaces, the rebase data for export and claim representatives, the read-off data, and the three split-pair tables (plus the parent-target table at a binary $z$-site). In the streamed presentation these data become an instruction chain; element digits repeat along the site's stretch, and the universe automaton enforces constancy. We maintain, bottom-up at every subtree $S$, the corresponding abstract interfaces $H^{\bullet}_S, E^{\mathrm{cert}}_S, G^{\mathrm{cert}}_S$ declared by the certificate. The four functional registers have their certified values, $m$ represents the routed export in $E^{\mathrm{cert}}_S$, and $g$ represents the residual through the embedding $G^{\mathrm{cert}}_S \hookrightarrow R^{S\cap Z}$ constructed in Lemma 18. For advice compiled from a tensor, these may be chosen as the concrete modules $E_S$ and $\Gamma_S$ above; an arbitrary valid certificate need only factor those concrete modules. The run remains alive exactly while the certified partial maps are defined and every digit sum holds.

*Leaves.* At an $x$-site leaf the functional registers initialise from the certified digit maps, and $E^{\mathrm{cert}} = G^{\mathrm{cert}} = 0$. At a $z$-site leaf $v$, the residual $\delta_v = (b_z - b_x - b_y)_v$ is passed to the partial inverse of $j_v : G^{\mathrm{cert}}_v \hookrightarrow R$; it yields the claim representative or rejects.

*Unary $x$-site $u$ over $D$.* The new pairs are $\{u, i\}$, $i \in D$. By Lemma 14(2) their coefficient vectors over $i$ lie in the column modules of $F_y(D)$ or $F_{py}(D)$ according to the target's position, so the contribution is $a_{x,u}\cdot$ (advice-known linear function of $w^y_D$ resp. $\varphi^y_D$); the part with targets outside $S$ enters $m$ (its target vectors are rows of $F_m(S)$), while the part with targets inside discharges $g$. The updated child claim is then passed through the partial inverse of $j_u : G^{\mathrm{cert}}_S \hookrightarrow G^{\mathrm{cert}}_D$, rejecting when it is outside the image. The export and functional registers update by the declared maps and absorb the node's two input digits. On a certificate compiled from a tensor, solvability of these maps is precisely Lemma 14(1)–(3) and Lemma 16(1),(2).

*Unary $z$-site $v$ over $D$.* The coordinate $v$ leaves the outside: the letter reads the $v$-coordinate of the represented element $\mu_D$ (a fixed function of $m$), the residual $\delta_v = (b_z - b_x - b_y)_v - \mu_D(v)$ arrives, and the partial inverse of $j_u$ combines it with the child claim, uniquely or with rejection. The export and functional interfaces update by their declared linear maps.

*Binary node $u$ over $D_0, D_1$.* One macro-transition consumes both child states and performs the following operations. First initialise $m$ with the restrictions of $\mu_{D_0} + \mu_{D_1}$ to targets outside $S_u$, and initialise temporary claims $g_0 := \delta_{D_0}$ and $g_1 := \delta_{D_1}$. Partition the split pairs between $D_0$ and $D_1$ by target into

$$B_{\mathrm{out}}, \qquad B_0, \qquad B_1, \qquad B_u$$

(the last block exists only when $u$ is a $z$-site). The first three pairing tables of Lemma 15 add the contribution of $B_{\mathrm{out}}$ to $m$ and subtract those of $B_0, B_1$ from $g_0, g_1$. Next restrict each child's export to the sibling's central sites and subtract it from the sibling claim; these are the cross-flows $\mu_{D_0}|_{D_1\cap Z}$ and $\mu_{D_1}|_{D_0\cap Z}$ of Lemma 14(3).

If $u$ is an $x$-site, first check its position-wise digit sum, then process the pairs $\{u, i\}$ separately for $i \in D_0$ and $i \in D_1$. For each child, sort their targets into sites outside $S_u$, sites in that child, and sites in its sibling. The read-offs of Lemma 14(2), applied to $w^y$ or

$\varphi^y$ as appropriate, add the outside part to $m$, subtract the same-child part from that child's claim, and subtract the sibling part from the sibling's claim. This accounts for the target class omitted by treating the parent as a unary node. The two resulting claims then join through the partial inverse of $j_u$, uniquely or with rejection, to $g \in G^{\mathrm{cert}}_{S_u}$.

If $u$ is a $z$-site, no new generator pair involves $u$. Instead its new residual coordinate is

$$\delta_u = (b_z - b_x - b_y)_u - \mu_{D_0}(u) - \mu_{D_1}(u) - P_u(w^x_{D_1}, w^y_{D_0}),$$

where $P_u$ is the fourth pairing table, for the split pairs targeting $u$. The child claims together with this coordinate extend through the partial inverse of $j_u$, uniquely or with rejection, to $g \in G^{\mathrm{cert}}_{S_u}$.

If $u$ is structural, there is no site digit and no parent contribution. The two updated child claims join, uniquely or with rejection, to $g \in G^{\mathrm{cert}}_{S_u}$.

In all three cases the surviving child exports and the outside split-pair contribution are already represented by $m \in E^{\mathrm{cert}}_{S_u}$. Finally the four functional interfaces fold from the raw child registers to the $S_u$-interfaces by Lemma 14(1), and at an $x$-site they also absorb the node's own digit column. Thus every pair and every target is handled exactly once.

*Root.* Nothing is outside: the boundary conditions give $E^{\mathrm{cert}} = G^{\mathrm{cert}} = 0$, and the automaton accepts iff the run is alive and $g$ represents 0. Unwinding the invariants, this holds iff all digit sums hold and every residual was discharged exactly, i.e. $b_z = b_x + b_y + C(a_x, a_y)$ coordinate-wise: acceptance is $z = x \cdot y$ in $G(T)$.

*Counting.* At site boundaries, six registers with values in $R^r$ give $p^{6rd}$ live macro-states plus the sink. A macro advice symbol contains the complete collection of fold, injection, read-off, rebase and pairing data for that site; this alphabet is finite for fixed $p, d, r$, but large. To keep the alphabet small, stream those data one ring entry at a time. During the resulting unary instruction chain the automaton must retain both six-register child interfaces, the partially built six-register output, and a constant number of scratch registers and control values. Thus it uses $O(1)$ many $R^r$ banks—for example, the implementation uses nine output/scratch banks and two saved six-bank child interfaces—and hence $p^{O(rd)}$ states, not $p^{6rd}+1$. The data comprise $\Theta(r^2)$ entries per fold matrix, $\Theta(r)$ per injection, and $p^{2rd} \cdot r$ per vector-valued pairing table, plus the scalar parent-target table. Consequently a site expands into $O_{p,d,r}(1)$ letters over the $p^d$ digit letters and finitely many operation markers.

By Lemma 18(1), every width-$\leq r$ site tensor occurs. By parts (2)–(3), every accepted advice presents $G(T)$ for a width-$\leq r$ site tensor $T$. Hence the presented class is exactly $\mathcal{C}^{\mathrm{site}}_{p,d,r}$. ◀

▶ Remark 20 (why the certificates are necessary). The larger set of merely well-shaped instruction trees is regular, but is not a valid advice set: incomplete operations can make multiplication partial, and arbitrary completed tables can violate associativity. Nor is it enough to retain those programs that happen to satisfy the group axioms. Group axioms alone do not force coordinate-wise addition on the $x$- and $z$-digits, bilinearity of the correction, or the six width bounds; arbitrary tables may, for instance, implement carries between coordinate sites. The algebraic types in $A^{\mathrm{cert}}_{p,d,r}$ impose exactly these missing conditions.

For a certified advice, the distinguished coordinate kernel is $K_\alpha = R^Z$. Each coordinate copy has order $p^d$, every element has order dividing $p^d$, and hence $K_\alpha$ has order $p^{d|Z|}$ and, when $Z \neq \varnothing$, exponent exactly $p^d$. This says nothing stronger about the full center: it may be larger, and at $p = 2$ may contain elements of order greater than $p^d$.

▶ **Corollary 21** (string version). *For all fixed $p, d, r$, the subclass of $\mathcal{C}^{\mathrm{site}}_{p,d,r}$ given by chain-shaped site trees is uniformly word-automatic, with the same state bounds in the macro-letter and streamed encodings, respectively.*

**Proof.** On a chain every node has at most one child, so a bottom-up run visits the sites in a linear order and never branches: the tree automaton *is* a word automaton reading the advice and the element encodings leaf-to-root, and the convolution of chain-shaped trees is the convolution of the corresponding words. Merges—the only genuinely tree-shaped transitions—do not occur. ◀

▶ **Corollary 22** (fixed central kernels, and the warm-ups). *(i) For all fixed $p, d, k, r$, every member of $\mathcal{C}^{\mathrm{tree}}_{p,k,r,d}$ has advice in the uniformly tree-automatic master class of width $\max(r, k)$. Indeed, place the $k$ central generators on a chain above the layout root. At every layout cut $F_g, F_{py}, F_{px}$ are empty and $\mathrm{rk}_R(F_m) \le k$, so Example 13(i) and Theorem 19 apply. This is a common presentation by the master automata, not an exact uniform presentation of $\mathcal{C}^{\mathrm{tree}}_{p,k,r,d}$: the restricted master advice language may also contain forms whose combined crossing block has rank greater than $r$. On spine layouts the run simulates the exact word presentation of Theorem 8. (ii) The two genuinely uniform warm-up families are width-one corners: the matching forms of Proposition 3 under (i), and the laminar site trees of Example 13(ii), where $\Gamma_S$ is generated by the all-ones vector, the represented claim is the single deficit owed to every leaf below, the join of Lemma 16(1) is "sibling subtrees must owe equally", and the protocol collapses to the $(p+1)$-state automaton of Proposition 4. The point-target tensors of Example 13(iii) lie in neither syntactic corner.*

## 6 The wealth of the class

This section delineates the scope of bounded width. For scalar forms over $\mathbb{F}_p$, on leaf layouts the parameter is exactly labelled-graph rank-width, and on node layouts it differs by at most one (Proposition 23); the class is closed under direct products, with the grafted block layout having width the *maximum*, not the sum, of the widths of its factors (Proposition 24); the classical width measures of the commutation graph bound it (Proposition 26); and at the tame end it collapses, whereas generic forms have linear width (Proposition 27, Proposition 28). Table 1 places the named constructions.

▶ **Proposition 23** (rank-width interface). *For $k = 1$ and $d = 1$, let $A_B$ be the symmetric matrix with off-diagonal entry $(A_B)_{ij} = \widehat{B}_{\{i,j\}}$. Define $\mathrm{rw}_p(B)$ as the $\mathbb{F}_p$-rank-width of this labelled graph in the sense of Kanté and Rao [17]: the minimum, over subcubic leaf decompositions, of the maximum rank of $A_B[S, [n] \setminus S]$ across a decomposition edge. Let $\mathrm{tcr}_{\mathrm{node}}(B)$ be the minimum width when generators may occur at arbitrary nodes, and let $\mathrm{tcr}_{\mathrm{leaf}}(B)$ be the minimum when every generator is at a leaf. Then*

$$\mathrm{tcr}_{\mathrm{node}}(B) \;\le\; \mathrm{tcr}_{\mathrm{leaf}}(B) \;=\; \mathrm{rw}_p(B) \;\le\; \mathrm{tcr}_{\mathrm{node}}(B) + 1.$$

*In particular, a rank-decomposition of width $r$ gives a layout of width $r$, while an arbitrary node layout of width $r$ gives a rank-decomposition of width at most $r + 1$.*

The alternating commutator matrix differs from $A_B$ by signs. This does not affect the cuts used here: a subtree is a post-order interval, so each outside row lies wholly before or wholly after it and receives one uniform sign.

**Proof.** A rank-decomposition is a subcubic tree with the vertices at its leaves; subdividing an arbitrary edge and rooting there yields a binary leaf layout whose internal nodes are structural. The generator set below any edge is one side of the corresponding decomposition cut, and its crossing block is the bipartite adjacency block defining cut-rank. This proves $\mathrm{tcr}_{\mathrm{leaf}} = \mathrm{rw}_p$, and $\mathrm{tcr}_{\mathrm{node}} \le \mathrm{tcr}_{\mathrm{leaf}}$ is immediate.

Conversely, leafify an arbitrary node layout. Leave a generator-carrying leaf unchanged. At a unary generator node, detach the generator to a new leaf beside its old child. At a binary generator node, replace it by a structural parent with the new generator leaf as one child and a structural node carrying the two old children as the other. The old cut is represented at the new parent; the additional cuts delete that generator from an old cut or isolate it. The two crossing matrices then share a common block, with one obtained from it by adjoining a row and the other by adjoining a column, so their ranks differ by at most one. Hence the leafified layout has width at most $\mathrm{tcr}_{\mathrm{node}}+1$, proving the last inequality. ◀

▶ **Proposition 24** (closure under direct products). *Let $T_1, T_2$ be cocycle tensors over $R$ on disjoint site trees, whose given layouts have widths $r_1, r_2$. Grafting one tree below a leaf of the other and taking the block-diagonal tensor presents $G(T_1) \times G(T_2)$, and the resulting layout has width $\max(r_1, r_2)$. The same holds for any finite number of factors.*

**Proof.** The combined tensor is block-diagonal: no commutation pair and no target uses sites from different blocks. Attach the root of $T_1$ as a child of a leaf of $T_2$; the relative post-order within each factor is unchanged. A subtree cut lying in $T_1$ has exactly the corresponding flattenings of $T_1$, padded by zeros. A cut in $T_2$ either excludes $T_1$ or contains it in full; since a whole tensor block contributes no crossing entry, its flattenings are exactly those of the corresponding cut of $T_2$, again padded by zeros. Thus the width is at most $\max(r_1, r_2)$, and each factor's cuts show equality. Iterate the construction for finitely many factors. ◀

▶ **Proposition 25** (presentation-independent scarcity). *For fixed $p, d, r$, the groups presented by $\mathcal{C}^{\mathrm{site}}_{p,d,r}$ include at most $2^{O(m)}$ isomorphism types of order $p^m$. Consequently their proportion among the class-2 groups of order $p^m$ tends to zero as $m \to \infty$.*

**Proof.** A group of order $p^m$ in the class has $s = |X| + |Z| = m/d$ sites. A reduced site tree has fewer than $s$ structural nodes, hence fewer than $2s$ nodes in total. For fixed $p, d, r$, the certified alphabet is finite, and the number of ordered binary trees with $O(m)$ nodes over a fixed finite alphabet is $2^{O(m)}$. Passing from advices to isomorphism types can only decrease this number. In contrast, the Higman–Sims lower bound is already attained by class-2 groups: their number at order $p^m$ is $p^{\frac{2}{27}m^3+O(m^{8/3})}$ [13, 24, 23]. The ratio therefore tends to zero. ◀

The proposition gives a group-theoretic closure operation preserving the width bound: the advice for a direct product is obtained by grafting the factors' advices. In particular this includes adjoining homocyclic abelian direct factors over the same ring $R$. We make no corresponding claim for arbitrary central products: identifying target generators can mix rows of the claim flattenings, and the maximum-width bound does not follow from block diagonality.

▶ **Proposition 26** (classical width measures). *Let $\mathcal{G}_B$ be the commutation graph of a form $B$ over $R$ (edges = non-commuting pairs).*

1. *If $\mathcal{G}_B$ has a vertex cover $Q$ of size $c$, then every layout that places $Q$ first has width $\le c$; in particular $\mathrm{lcr}(B) \le c$. An arbitrary layout has width $\le kc$, and the factor $k$ is necessary.*
2. *If $\mathcal{G}_B$ has pathwidth $w$, then $\mathrm{lcr}(B) \le w + 1$.*
3. *If $\mathcal{G}_B$ has treewidth $w$, then $\mathrm{tcr}(B) \le k(w + 1)$.*

**Proof.** All three are counting arguments—the module rank of a matrix is at most its number of nonzero rows and at most its number of nonzero columns—and hold over $R$ verbatim. (1) Put the cover first. A cut $S \subseteq Q$ offers at most $|S| \le c$ columns. A cut $S \supseteq Q$ contains every

| Construction | Setting | Width statement |
|---|---|---|
| homocyclic abelian $(\mathbb{Z}/p^d)^m$ | zero tensor | 0 |
| extraspecial $p^{1+2m}$ | matching, $k = 1$ | 1 |
| binary clique cocycle | $p = 2$; triangle is the 1-qubit Pauli group | $\leq 1$ |
| shared-target graph group over $\mathbb{F}_p$ | $[x_i, x_j] = y$ on edges | graph rank-width (leaf layouts) |
| tree-indexed laminar | distributed central kernel, laminar | $\leq 1$ |
| point-target family | distributed, non-laminar | $\leq 1$ |
| rank-one families over $\mathbb{Z}/p^d$ | matching, clique, laminar, point-target | $\leq 1$ |
| direct products | grafted block layout | max of block widths |
| ultraspecial (Heisenberg over $\mathbb{F}_{p^m}$) | field-multiplication tensor | $\Theta(m)$ in fixed-kernel layouts |
| robust-complement Mekler families | private target per non-edge | $\Omega(n)$ |
| relatively free class-2, exponent $p$ ($p$ odd) | fully free | not automatic [20] |

**Table 1** Named tensor presentations and their displayed layout models. The rows below the line exhibit different obstructions to bounded width or automaticity; they are examples, not a characterisation of model-theoretic wildness. Over $R$, only the listed rank-one lifts are asserted to retain their field width; arbitrary lifts need not do so.

edge's cover endpoint, so the nonzero columns of the crossing block are indexed by $Q$: again module rank $\leq c$. In an arbitrary layout the nonzero entries lie in the $c_1 \leq |Q \cap S|$ columns of the inside cover vertices and the $\leq kc_2$ rows of the outside ones, whence module rank $\leq c_1 + kc_2 \leq kc$; a star with $c = 1$ whose edge to the $j$-th leaf is labelled $e_j$ attains $k$ at the cut separating the star center from its leaves. (2) Order the vertices by their introduction positions (first bags) in a path decomposition, breaking ties arbitrarily. At any cut, every inside endpoint of a crossing edge is still present in the boundary bag: it must survive until the first bag of its outside neighbour. Hence the nonzero columns number at most $w + 1$. (3) First replace the tree decomposition by a binary one of the same width. Root it and assign each vertex to its highest bag. Replace every bag by a chain carrying the vertices assigned to it, and connect these chains according to the decomposition tree. At every resulting subtree cut, an edge leaving the subtree has its outside endpoint in the current bag: an inside vertex assigned strictly below cannot share a bag above the cut, while the connectedness axiom for the outside endpoint carries it through the separator. The same statement holds at an intermediate cut of a bag-chain. Thus all nonzero rows of the crossing block are indexed by at most $w + 1$ outside vertices and the $k$ target coordinates, giving module rank at most $k(w + 1)$. ◀

▶ **Proposition 27** (collapse at the tame end). *Let $p$ be odd and $d = 1$.*

1. *If $\mathcal{G}_B$ has a vertex cover of size $c$, then $G(B) \cong H \times \mathbb{Z}_p^m$ with $|H| \leq p^{c(k+1)+k}$; for fixed $c$ and $k$ the class of such groups is, up to isomorphism, a finite list of cores times elementary abelian factors.*
2. *For $k = 1$ and $B \neq 0$, if $B$ has rank $2m$, then $G(B) \cong E \times \mathbb{Z}_p^{n-2m}$ for an extraspecial group $E$ of exponent $p$ and order $p^{1+2m}$. If $B = 0$, then $G(B)$ is elementary abelian.*

**Proof.** (1) Put $V = \mathbb{F}_p^n$, let $Q$ be the chosen vertex cover, let $U$ be the span of the vertices outside $Q$, and write $\mathrm{rad}(B) = \{v : B(v, V) = 0\}$. Since the vertices outside $Q$ form an

independent set, the kernel of

$$U \longrightarrow \mathrm{Hom}(\mathbb{F}_p^Q, \mathbb{F}_p^k), \qquad u \longmapsto B(u,-)|_{\mathbb{F}_p^Q},$$

is $U \cap \mathrm{rad}(B)$. Hence $\mathrm{codim}_V \mathrm{rad}(B) \le c + ck = c(k+1)$. For odd $p$, the triangular cocycle is cohomologous, by a quadratic change of section, to one half of its alternating commutator map. Choosing a complement $V = \mathrm{rad}(B) \oplus W$ therefore gives $G(B) \cong \mathbb{Z}_p^{\dim \mathrm{rad}(B)} \times H$, where $H$ has underlying space $\mathbb{F}_p^k \times W$. Thus $|H| \le p^{\,c(k+1)+k}$; the bound's shape is right: a star with $c = 1$, $k = n-1$ and $B_{ui} = e_i$ has trivial radical and $|H| = p^{2n-1}$. For (2), the classical normal form of a nonzero alternating form of rank $2m$ is a matching of $m$ hyperbolic pairs with the radical split off; the matching part presents $E$. The zero form gives the stated elementary abelian group. ◀

Thus, for odd $p$ at $k = d = 1$ (and for bounded vertex cover at any fixed $k$), the parameter provides presentation-theoretic rather than isomorphism-theoretic generality. The clique of Example 10, for example, is abstractly a Heisenberg group times an abelian group, but the automaton multiplies in the *given* presentation without first computing a normal form. In the odd-prime field setting, $k \ge 2$ already lies beyond the scalar symplectic normal form, and at $k = 3$ the general tensor-isomorphism difficulty appears [10]. Characteristic 2 introduces power-map or quadratic data already at $k = 1$, while $d \ge 2$ lets the distinguished central kernel carry exponent information absent over a field. Distributed central targets add further content already at width one: the laminar and point-target classes of Theorem 19 contain bushy families whose commutator subgroups have rank $\Theta(n)$, escaping the collapse of Proposition 27.

▶ **Proposition 28** (generic forms are wide). *Let $k = 1$, $d = 1$, and let $B$ be uniformly random. With high probability, every linear layout of $B$ has width $\ge n/2 - O(\sqrt{n})$ and every tree layout has width $\ge n/3 - O(\sqrt{n})$; in particular* $\mathrm{tcr}(B) = \Omega(n)$.

**Proof sketch.** For a fixed set $S$ with $|S| = m \le n/2$, the crossing block is a uniform random $(n-m) \times m$ matrix over $\mathbb{F}_p$, and $\Pr[\mathrm{rk} < m - s] \le p^{-s(n-2m+s)} \le p^{-s^2}$. A union bound over all $\le 2^n$ sets with $s = c\sqrt{n}$ leaves all sets of size $m \in [n/3, n/2]$ with rank $\ge m - c\sqrt{n}$. Every linear layout has a cut of size $\lfloor n/2 \rfloor$, and every binary tree has a subtree whose interval size lies in $[n/3 - O(1), 2n/3 + O(1)]$; complement if necessary. The additive constant is absorbed by the displayed $O(\sqrt{n})$ term. ◀

**The dividing line.** The following examples exhibit independent ways to leave the bounded-width regime; they do not characterise model-theoretic wildness.

▶ **Example 29** (ultraspecial groups). The Heisenberg group over the field $\mathbb{F}_{p^m}$—an *ultraspecial* group—has generators $a_1, \dots, a_m, b_1, \dots, b_m$ and commutation tensor the multiplication of $\mathbb{F}_{p^m}$, written over $\mathbb{F}_p$ with $k = m$: $[a_i, b_j]$ is the field product $e_i e_j$ in the center. Its fixed-kernel form has width $\Theta(m)$ under every generator layout. Indeed, the standard balanced-subtree argument gives, for $m$ larger than an absolute constant, a cut $S$ with $m/2 \le |S| \le 3m/2$. Put

$$a = |S \cap \{a_1, \dots, a_m\}|, \qquad b = |S \cap \{b_1, \dots, b_m\}|.$$

If $a \ge m/4$ and some $b_j$ lies outside $S$, restrict the crossing block to the rows $(b_j, l)$ and the columns $a_i \in S$. Its columns are the coordinate vectors of $e_i e_j$; multiplication by the nonzero field element $e_j$ is injective, so this submatrix has rank $a \ge m/4$. If no $b_j$ lies outside, then

$b = m$ and some $a_i$ does, and the symmetric submatrix has rank $m$. Finally, if $a < m/4$, then $b > m/4$ and some $a_i$ lies outside, so the same symmetric argument gives rank $b > m/4$. Thus every layout has width at least $m/4$, while the number of generators gives the matching $O(m)$ upper bound.

▶ **Example 30** (Mekler groups). For odd $p$, Mekler's construction [19] interprets a graph in a class-2 exponent-$p$ group whose commutator structure gives each non-adjacent generator pair a *private* central target. To state the width consequence at the right level of generality, suppose a family of underlying graphs has the following robust-complement property: across every vertex cut with both sides of size at least $n/4$, each side contains $\Omega(n)$ vertices having a non-neighbour on the other. Bounded-degree families have this property for all sufficiently large $n$.

Fix any site layout of the corresponding Mekler tensor. Give $x$-sites weight one and $z$-sites weight zero; the balanced-subtree argument gives a subtree $S$ containing between one quarter and three quarters of the $x$-sites. For each of $\Omega(n)$ inside vertices choose one non-neighbour outside. The chosen pairs have distinct inside endpoints and distinct private targets. If a target lies outside $S$, its entry gives a row of $F_y$ or $F_x$, according to the orientation of the pair; if it lies inside, it gives a row of $F_g$. Hence one of these three flattenings receives $\Omega(n)$ chosen rows. Within that flattening their supports are disjoint: in the first two cases the support column contains the private target, and in the last case both the target row and private-pair column are unique. The rows are therefore independent, and the layout has width $\Omega(n)$. Here the tensor has $|X| = n$ and $|Z| = \Theta(n^2)$, so for fixed $p$ this is a lower bound $\Omega(\sqrt{\log_p |G|})$ in the size of the group. The argument applies to robust-complement Mekler families, not to every graph to which Mekler's construction can be applied.

Together with Proposition 28 and the theorem of Nies and Stephan that, for odd $p$, the relatively free class-2 exponent-$p$ group is not word-automatic [20], these examples show that generic coordinate presentations, fixed-kernel field multiplication, robust private-target families and full freeness encounter different obstructions. The first two statements concern the specified presentation models; the Mekler bound ranges over all site layouts of that tensor. The presentation-independent conclusion is Proposition 25: bounded width captures an asymptotically negligible family of isomorphism types.

## 7 Algorithmic consequences

The generic machinery of uniformly automatic classes [27, 25] yields the following for the master presentation. Because every member of a width-bounded family in Section 6 has advice in the master class, the same compiled evaluator applies to each member. Statements about the common theory of a proper subfamily additionally require an effective restriction of the advice language.

▶ **Corollary 31** (compiled model checking). *For every first-order formula $\varphi(x_1, \dots, x_m)$ in the signature of groups there is (effectively) an automaton $\mathfrak{A}_\varphi$ such that for every advice $\alpha$ in the advice set and all elements $u_1, \dots, u_m$ of $G_\alpha$:*

$$G_\alpha \models \varphi(\bar{u}) \iff \mathfrak{A}_\varphi \text{ accepts } \alpha \otimes u_1 \otimes \dots \otimes u_m.$$

*The input is the valid advice together with the encodings of the free variables. Once $\varphi$ is compiled, the automaton runs in time $O(N) = O(\log |G|)$, where $N$ is the size of the macro- or streamed advice. In particular, for a sentence $\varphi$, deciding $G_\alpha \models \varphi$ from the advice alone takes $O(\log |G_\alpha|)$ time.*

**Proof.** The usual induction on $\varphi$ uses the effective closure of synchronous word and tree automata under Boolean operations and projection. Once the resulting automaton is fixed, its run is linear in the convolution size $N$. Here $|G_\alpha| = p^{d(|X|+|Z|)}$, while both advice encodings use $\Theta_{p,d,r}(|X| + |Z|)$ nodes, proving the final equality. ◀

For fixed $p, d, r$, certified advice can be compiled in polynomial time from a tensor and a supplied width-$r$ layout by Smith normal form and linear-system solving over $R$. Finding such a layout is a separate problem and is not included in the model-checking bound; even in the fixed-kernel setting it is known algorithmically only in special cases (see Section 9).

▶ **Corollary 32** (decidable uniform theory)**.** *The advice sets of Theorem 8 and (by Lemma 18) of Theorem 19 are regular. Hence their uniform first-order theories*

$$\{\varphi : G_\alpha \models \varphi \text{ for every valid advice } \alpha\}$$

*are decidable, as is the question whether some valid advice presents a group satisfying $\varphi$. The same conclusion holds after restriction to any regular sublanguage of valid advices, but does not follow for an arbitrary subclass merely from its inclusion in the master class. It also holds for extensions of first-order logic by regularity-preserving quantifiers, in particular by any fixed collection of modulo-counting quantifiers [27].*

**Proof.** For a sentence $\varphi$, Corollary 31 effectively constructs a word or tree automaton for the valid advices whose groups satisfy $\varphi$. The universal question is inclusion of the regular advice language in this language, and the existential question is non-emptiness of their intersection. Both are decidable. The same argument applies to a regular advice restriction and to every quantifier for which the corresponding automata relations remain effectively regular. Infinite-cardinality and Ramsey quantifiers add nothing here, since every $G_\alpha$ is finite. ◀

▶ **Example 33.** The center is definable, $\mathrm{Cen}(x) \equiv \forall u \forall v \forall w\, (M(x,u,v) \wedge M(u,x,w) \to v = w)$, so Corollary 31 decides centrality of a given element in time $O(\log |G|)$ on every member; likewise commutativity of the member, existence of central elements of order 4 at $p = 2$ (distinguishing the Pauli triangle from the dihedral matching), and the exponent of the full center for $d > 1$. The same applies to any fixed first-order property of the multiplication structure. One automaton per property serves the entire class and, by Proposition 24, every finite direct product of its members.

## 8 Implementation, validation, and feasibility

The automaton constructions of this paper have executable counterparts in the open-source library AutStr [26]: `CutRankGroups` (Theorem 8), `CutRankTreeGroups` (Corollary 22(i)), and `CocycleRankWidthGroups` (the protocol of Theorem 19 for configurable $r$ and $d$). Each is equipped with a layout compiler that realises the interface lemmas by linear algebra over $R$. The current constructors restrict $p$ to the single-digit primes, solely because ring digits are encoded by single characters. For the distributed-target class, the library exposes the compiled instruction stream, its exact transition simulator, and implicit first-order evaluation. It does not materialise the full explicit presentation automaton.

**Instance checks and regression tests.** Every algebraic constant in compiler-generated advice is obtained by solving a linear system whose solvability is an instance of Lemma 7, Lemma 14, Lemma 15 or Lemma 16. An assertion guards each solution, so a successful

compilation has checked the relevant membership and factorisation obligations for that instance. This is not a formal verification of the general theorem: in particular, the library does not implement the recogniser for arbitrary certified advice from Lemma 18.

The retained regression suites compare the reference law of Lemma 2 with the explicit multiplication automata where these can be built, and otherwise with the exact transition simulator. They cover structured layouts (spines and balanced trees; zero, complete, matching and star forms; laminar, point-target, scattered-target and mixed-claim site trees), seeded random width-1 and width-2 instances, and the rings $\mathbb{Z}/4$, $\mathbb{Z}/8$ and $\mathbb{Z}/9$, including the phenomena of Example 5 and Example 6.

The distributed-target implementation uses the six committed registers of Theorem 19 for any configured $r, d$; its one-entry stream uses saved child banks and three scratch registers to execute each merge. The fixed-kernel classes likewise offer flat and streamed advice, the latter permitting $r \geq 2$ over the ring without enumerating the full macro-alphabet.

**Feasibility.** Two evaluation paths are available. The *compiled* path builds the presentation automata explicitly and evaluates formulas by automata products and projections—the promise of Corollary 31, with the one-time compilation cost amortised over members. The *implicit* path never builds an automaton: it evaluates a formula directly over the transition functions, taking products of composite states on the fly and handling quantifiers by an on-the-fly powerset construction over the element digits. The fixed-kernel classes use a conservative cost guard: small presentations may take the compiled path, while instances whose transition enumeration exceeds the guard are directed to the implicit evaluator. For `CocycleRankWidthGroups` the explicit class constructor is disabled altogether; simulation and implicit evaluation are the supported paths. Regression tests compare the two paths wherever both are available and compare both with direct finite evaluation on small members.

These facts establish executable coverage, not a machine-independent performance boundary. The repository contains a benchmark driver for an explicit/implicit sweep, but no result of that sweep is presently archived with the library. We therefore make no quantitative scalability claim here. In particular, implicit evaluation is an implementation of first-order evaluation on a fixed compiler-generated member; it is not an implementation of the class-wide uniform-theory decision procedure of Corollary 32.

## 9 Open problems

1. *Converse / characterisation.* Suppose a class of class-2 groups in the present algebraic setting is uniformly word- or tree-automatic. Must every member admit *some* cocycle presentation and layout of uniformly bounded width? The existential quantifier over presentations is essential: Proposition 28 gives wide scalar presentations of groups that also have width-one presentations by Proposition 27. A positive answer in this presentation-aware form would be a Courcelle–Oum-style characterisation [7]. The scarcity bound (Proposition 25) and the lower-bound examples provide partial evidence, but no converse is known.
2. *Isometry in P.* Is isometry of alternating matrix spaces of bounded $\mathbb{F}_p$ rank-width decidable in polynomial time? Here the width of a matrix space $\mathcal{B} \leq \mathrm{Alt}(n, \mathbb{F}_p)$ means the minimum over layouts of the maximum, across a cut $S$, of the rank of the vertical concatenation of $A_{[n]\setminus S,S}$ over a basis $A$ of $\mathcal{B}$. This number is independent of the chosen basis. The question is the matrix-space analogue of isomorphism testing for graphs of bounded rank-width [11] and would identify a tame island inside a tensor-isomorphism-complete

problem [10]; with distributed central targets and $d > 1$ the same question extends to isomorphism of the width-$r$ tensor classes of Theorem 19. Computing a witnessing layout is itself nontrivial; for $k = 1$, $d = 1$ rank-decompositions are FPT-computable [14].

3. *Beyond exponent $p^d$ and class 2.* Arbitrary bounded abelian quotients add a carry layer to the extension data; class 3 makes the correction a degree-three stream. Does the cut-rank philosophy—bound the module rank of every flattening a one-pass evaluation must transport—extend along the Lazard correspondence [5] to higher class, and where exactly does it stop short of the non-automaticity of [20]?

## Declaration of Generative AI Usage

In accordance with the Schloss Dagstuhl Publishing policy, the author declares that `Claude Fable 5`, `Claude Opus`, and `OpenAI Codex` assisted with drafting, proof auditing, implementation comparison, and bibliographic checks. The human author supplied the mathematics and editorial decisions, reviewed and edited all outputs, and takes full responsibility for the manuscript.

## References


1. Reinhold Baer. Groups with abelian central quotient group. *Transactions of the American Mathematical Society*, 44(3):357–386, 1938. doi:10.1090/S0002-9947-1938-1501972-1.
2. Vince Bárány, Erich Grädel, and Sasha Rubin. Automata-based presentations of infinite structures. In Javier Esparza, Christian Michaux, and Charles Steinhorn, editors, *Finite and Algorithmic Model Theory*, pages 1–76. Cambridge University Press, 2011. doi:10.1017/CBO9780511974960.002.
3. Achim Blumensath and Erich Grädel. Automatic structures. In *15th Annual IEEE Symposium on Logic in Computer Science (LICS 2000)*, pages 51–62. IEEE Computer Society, 2000. doi:10.1109/LICS.2000.855755.
4. Peter A. Brooksbank, Joshua Maglione, and James B. Wilson. A fast isomorphism test for groups whose Lie algebra has genus 2. *Journal of Algebra*, 473:545–590, 2017. doi:10.1016/j.jalgebra.2016.12.007.
5. Serena Cicalò, Willem A. de Graaf, and Michael Vaughan-Lee. An effective version of the Lazard correspondence. *Journal of Algebra*, 352(1):430–450, 2012. doi:10.1016/j.jalgebra.2011.11.031.
6. Bruno Courcelle. The monadic second-order logic of graphs. I: Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.
7. Bruno Courcelle and Sang il Oum. Vertex-minors, monadic second-order logic, and a conjecture by Seese. *Journal of Combinatorial Theory, Series B*, 97(1):91–126, 2007. doi:10.1016/j.jctb.2006.04.003.
8. Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000. doi:10.1007/s002249910009.
9. Erich Grädel. Automatic structures: Twenty years later. In *35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2020)*, pages 21–34. ACM, 2020. doi:10.1145/3373718.3394734.
10. Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials I: Tensor isomorphism-completeness. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *LIPIcs*, pages 31:1–31:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.31.

**11** Martin Grohe and Pascal Schweitzer. Isomorphism testing for graphs of bounded rank width. In *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS 2015)*, pages 1010–1029. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.66.

**12** Philip Hall. The classification of prime-power groups. *Journal für die reine und angewandte Mathematik*, 182:130–141, 1940. doi:10.1515/crll.1940.182.130.

**13** Graham Higman. Enumerating p-groups. I: Inequalities. *Proceedings of the London Mathematical Society*, s3-10(1):24–30, 1960. doi:10.1112/plms/s3-10.1.24.

**14** Petr Hliněný and Sang il Oum. Finding branch-decompositions and rank-decompositions. *SIAM Journal on Computing*, 38(3):1012–1032, 2008. doi:10.1137/070685920.

**15** Sang il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006. doi:10.1016/j.jctb.2005.10.006.

**16** Gábor Ivanyos and Youming Qiao. Algorithms based on *-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing. *SIAM Journal on Computing*, 48(3):926–963, 2019. doi:10.1137/18M1165682.

**17** Mamadou Moustapha Kanté and Michael Rao. The rank-width of edge-colored graphs. *Theory of Computing Systems*, 52(4):599–644, 2013. doi:10.1007/s00224-012-9399-y.

**18** Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In *Logic and Computational Complexity (LCC 1994)*, volume 960 of *Lecture Notes in Computer Science*, pages 367–392. Springer, 1995. doi:10.1007/3-540-60178-3_93.

**19** Alan H. Mekler. Stability of nilpotent groups of class 2 and prime exponent. *The Journal of Symbolic Logic*, 46(4):781–788, 1981. doi:10.2307/2273227.

**20** André Nies and Frank Stephan. Word automatic groups of nilpotency class 2. *Information Processing Letters*, 183:106426, 2024. arXiv:2302.12446. doi:10.1016/j.ipl.2023.106426.

**21** André Nies and Richard M. Thomas. FA-presentable groups and rings. *Journal of Algebra*, 320(2):569–585, 2008. doi:10.1016/j.jalgebra.2007.04.015.

**22** Graham P. Oliver and Richard M. Thomas. Automatic presentations for finitely generated groups. In *22nd Annual Symposium on Theoretical Aspects of Computer Science (STACS 2005)*, volume 3404 of *Lecture Notes in Computer Science*, pages 693–704. Springer, 2005. doi:10.1007/978-3-540-31856-9_57.

**23** László Pyber. Enumerating finite groups of given order. *Annals of Mathematics*, 137(1):203–220, 1993. doi:10.2307/2946623.

**24** Charles C. Sims. Enumerating p-groups. *Proceedings of the London Mathematical Society*, s3-15(1):151–166, 1965. doi:10.1112/plms/s3-15.1.151.

**25** Faried Abu Zaid. Uniformly automatic classes of finite structures. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018)*, volume 122 of *LIPIcs*, pages 10:1–10:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.FSTTCS.2018.10.

**26** Faried Abu Zaid. AutStr. Software, version 4.0.0, 2026. URL: https://pypi.org/project/autstr/4.0.0/.

**27** Faried Abu Zaid, Erich Grädel, and Frederic Reinhardt. Advice automatic structures and uniformly automatic classes. In *26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*, volume 82 of *LIPIcs*, pages 35:1–35:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.CSL.2017.35.