Computing edge-colored ultrahomogeneous graphs *

Irene Heinrich, Eda Kaja, and Pascal Schweitzer

TU Darmstadt, Darmstadt, Germany

Abstract

We develop a practical algorithm to enumerate all ultrahomogeneous edge-colored graphs up to a specified order. As input, the algorithm can take either a list of all coherent configurations or all transitive permutation groups. Efficiency is achieved by pruning lexicographic products quickly. We provide numerical data on the number of objects up to isomorphism for orders up to 34.

1 Introduction

Ultrahomogeneous structures are classical objects in model theory with applications in algebra and combinatorics. A structure $\mathcal{R}$ is *ultrahomogeneous* if every isomorphism between two induced substructures of $\mathcal{R}$ extends to an automorphism of $\mathcal{R}$. We are interested in algorithms for generating and handling finite ultrahomogeneous structures. In this paper we develop algorithms for the base case of structures with irreflexive binary relations. These structures can always be translated into loopless edge-colored graphs. In our search for ultrahomogeneous structures in the base case, it suffices to consider vertex-monochromatic coherent configurations. These are binary relational structures that satisfy certain regularity conditions implied by ultrahomogeneity. There is a well-known Galois correspondence between coherent configurations and permutation groups. There are thus two starting points that we can take for our search: coherent configurations or transitive groups. Since we are only interested in ultrahomogeneous structures we can limit ourselves to so-called 2-closed permutation groups on the permutation group side.

For various subclasses of finite ultrahomogeneous structures, explicit classifications are known. This is the case for simple graphs [3, 12], directed [10], 3-edge-colored [11], vertex-colored [9], and vertex-colored oriented [8] graphs. None of these classifications allow arbitrarily many edge colors. Also the primitive permutation groups of finite binary ultrahomogeneous structures have recently been classified [4].

As the class of graphs considered becomes larger, the classification of ultrahomogeneous objects becomes ever more complicated. A computer assisted approach seems to be in order.

Results. We develop a practical algorithm to enumerate all ultrahomogeneous edge-colored graphs up to a specified order. The algorithm takes as input a list of all coherent configurations or all transitive permutation groups of at most the given order.

Techniques. First, we provide a fast practical algorithm that checks whether a given object is ultrahomogeneous. Our algorithm can take either coherent configurations or permutation groups as input and checks whether the graph induced by the input is ultrahomogeneous (Subsection 3.1).

The lexicographic product operation of graphs preserves ultrahomogeneity and it turns out that many ultrahomogeneous objects are in fact lexicographic products of smaller ultrahomogeneous objects.

*The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (EngageS: Grant Agreement No. 820148) and from the German Research Foundation DFG (SFB-TRR 195 "Symbolic Tools in Mathematics and their Application"). Emails: lastname@mathematik.tu-darmstadt.de.

Second, to avoid a combinatorial explosion, we develop a linear time algorithm to prune such products. The algorithm takes as input a coherent configuration and determines that it is “not a lexicographic product” or renders the input as “a product or not ultrahomogeneous” (Subsection 3.2). The crux here is that in the latter case, we do not need to process the input since both non-ultrahomogeneous objects and products can be discarded. This allows our algorithm to avoid checking regularity of the input and thus run in time $O(kn)$, where k is the rank of the configuration (number of colors/binary relations) and n the number of vertices. The computational results are aggregated in Section 4.

2 Preliminaries

For $k \in \mathbb{N}$ we set $[k] := \{1, 2, \ldots, k\}$ and for a k-tuple $t = (v_1, v_2, \ldots, v_k)$ we set $\pi_i(t) := v_i$. The *restriction* of a map ψ to a set U is $\psi|_U$. If φ is a restriction of ψ, then ψ is an *extension* of φ.

A *binary relational structure* is a tuple $\mathcal{R} = (V, R_1, R_2, \ldots, R_k)$ where V is a set of *vertices* and $R_i \subseteq V^2$ for each $i \in [k]$. We set $V(\mathcal{R}) := V$. Throughout this paper we assume, for our purposes w.l.o.g., that $\{R_i : i \in [k]\}$ is a partition of V^2, there exists a $d \in [k]$ such that $R_d = \{(v, v) : v \in V\}$ is the *diagonal* of $\mathcal{R}$, and for each $i \in [k]$ either R_i is symmetric or there exists $j \in [k]$ such that $R_j = \{(v, u) : (u, v) \in R_i\}$. All relational structures in this paper are binary and finite. Two relational structures $\mathcal{R} = (V, R_1, R_2, \ldots, R_k)$ and $\mathcal{S} = (W, S_1, S_2, \ldots, S_k)$ are *isomorphic* if there exists a bijection $\varphi : V \to W$ such that for all $i \in [k]$ it holds that $(u, v) \in R_i$ if and only if $(\varphi(u), \varphi(v)) \in S_i$. In this case φ is an *isomorphism* from $\mathcal{R}$ to $\mathcal{S}$. If, additionally, $\mathcal{R} = \mathcal{S}$, then φ is an *automorphism* of $\mathcal{R}$. The automorphisms of $\mathcal{R}$ form a group, denoted by $\mathrm{Aut}(\mathcal{R})$. For a subset U of V the *induced substructure* of $\mathcal{R}$ on U is $\mathcal{R}[U] := (U, R_1 \cap U^2, R_2 \cap U^2, \ldots, R_k \cap U^2)$. A *partial isomorphism* from $\mathcal{R}$ to itself is an isomorphism between two induced substructures of $\mathcal{R}$. If every partial isomorphism from $\mathcal{R}$ to itself extends to an automorphism of $\mathcal{R}$, then $\mathcal{R}$ is *ultrahomogeneous*. The structures $\mathcal{R}$ and $\mathcal{S}$ are *equivalent* if there exists a permutation σ of $[k]$ such that $\mathcal{R}$ is isomorphic to $(W, S_{\sigma(1)}, S_{\sigma(2)}, \ldots, S_{\sigma(k)})$.

A (vertex) *coloring* of $\mathcal{R}$ is a map $\chi : V(\mathcal{R}) \to C$ where C is some set of *colors*. Then $(\mathcal{R}, \chi)$ is a *colored relational structure*. An inclusion-wise maximal set $U \subseteq V(\mathcal{R})$ with $|\chi(U)| = 1$ is a color class of $(\mathcal{R}, \chi)$. The definitions of isomorphisms, automorphisms, and ultrahomogeneity directly transfer to the context of colored structures, where it is important that isomorphisms preserve vertex colors. Note that there is a 1:1-correspondence between colored relational structures we consider and complete edge-colored digraphs, where the edge colors of the digraph correspond to the indices of the relations.

Coherent configurations. A relational structure $\mathcal{R} = (V, R_1, R_2, \ldots, R_k)$ is a *coherent configuration*[1] if for every choice of three indices $a, b, c \in [k]$ and $(u, w) \in R_a$ the number of elements $v \in V$ such that $(u, v) \in R_b$ and $(v, w) \in R_c$ is a constant $\lambda^a_{b,c}$ which is independent of the choice of u and w. A coherent configuration $\mathcal{R}$ is *symmetric* if all the relations are symmetric.

Lexicographic products. Let $\mathcal{R} = (V, R_1, R_2, \ldots, R_k)$ and $\mathcal{S} = (W, S_1, S_2, \ldots, S_\ell)$ be two relational structures such that R_k is the diagonal of $\mathcal{R}$ and S_ℓ is the diagonal of S. The *lexicographic product* of $\mathcal{R}$ and $\mathcal{S}$ is the relational structure $\mathcal{R} \cdot \mathcal{S} = (V \times W, \dot{R}_1, \dot{R}_2, \ldots, \dot{R}_{k-1}, \dot{S}_1, \dot{S}_2, \ldots, \dot{S}_\ell)$ where $\dot{R}_i = \{((v, w), (v'w')) \in (V \times W)^2 : (v, v') \in R_i\}$ and $\dot{S}_j = \{((v, w), (v, w')) : v \in V, (w, w') \in S_j\}$. We emphasize that the index i of $\dot{R}_i$ is at most $k - 1$ (otherwise the diagonal $\dot{S}_\ell$ would have a non-empty intersection with $\dot{R}_k$). Observe that the relations of $\mathcal{R} \cdot \mathcal{S}$ indeed form a partition $V \times W$. We say that $\mathcal{R}$ or $\mathcal{S}$ is a *trivial factor* of $\mathcal{R} \cdot \mathcal{S}$ if $|V| = 1$ or $|W| = 1$, respectively.

Lemma 1. *If* $|W| \geq 2$ *and* $\min_{i \in [k]}\{|R_i|\} \geq |V|$, *then* $\max_{j \in [\ell]} |\dot{S}_j| < \min_{i \in [k-1]} |\dot{R}_i|$. *In particular, if* $\mathcal{R}$ *is a coherent configuration and* $\mathcal{S}$ *is a non-trivial factor of* $\mathcal{R} \cdot \mathcal{S}$, *then* $\max_{j \in [\ell]} |\dot{S}_j| < \min_{i \in [k-1]} |\dot{R}_i|$.

[1] Technically these are colored coherent configurations and usually the underlying uncolored object is considered, i.e., the ordering of the relations is ignored. Certain coherent configurations are sometimes called association schemes.

In general, we say that a structure *is a lexicographic product* whenever it is equivalent to a lexicographic product. We say that $\mathcal{R}$ is *prime* if $|V(\mathcal{R})| \geq 2$ and every structure $\mathcal{R}'$ equivalent to $\mathcal{R}$ satisfies: $\mathcal{R}' = \mathcal{S}_1 \cdot \mathcal{S}_2$ implies $\min\{|V(\mathcal{S}_1)|, |V(\mathcal{S}_2)|\} = 1$.

Groups. The *symmetric group* of a non-empty set V is $\mathrm{Sym}(V)$. A *permutation group* Γ on V is a subgroup of $\mathrm{Sym}(V)$. For $v \in V$ and $\gamma \in \Gamma$ we set $v^\gamma := \gamma(v)$. An *action* of Γ on V is a homomorphism ϕ from Γ to $\mathrm{Sym}(V)$. The image of Γ under ϕ is a subgroup of $\mathrm{Sym}(V)$ called the permutation group *induced* by Γ on V, denoted Γ^V. The *orbit* of $x \in V$ is $x^\Gamma := \{x^\gamma : \gamma \in \Gamma\}$. We say Γ is *transitive* on V if $x^\Gamma = V$ for all $x \in V$. The *stabilizer* of $x \in V$ is $\mathrm{Stab}_\Gamma(x) := \{\gamma \in \Gamma : x^\gamma = x\}$. The *pointwise stabilizer of* $X \subseteq V$ is $\mathrm{pwStab}_\Gamma(X) := \bigcap_{x \in X} \mathrm{Stab}_\Gamma(x)$. If Γ is a transitive permutation group on V, then the partiton of $V \times V$ into orbits of Γ is a coherent configuration $\mathcal{R}(\Gamma)$. Note that $\Gamma \leq \mathrm{Aut}(\mathcal{R}(\Gamma))$. If equality holds, then Γ is 2-*closed*, that is, Γ equals its 2-*closure*, which is the largest subgroup of $\mathrm{Sym}(V)$ which preserves the orbits of Γ on $V \times V$. A coherent configuration $\mathcal{R}$ such that $\mathcal{R} = \mathcal{R}(\Gamma)$ for some transitive permutation group Γ is called *Schurian*.

Block systems. Let $\Gamma \leq \mathrm{Sym}(V)$ be transitive. A *block* is a set $B \subseteq V$ such that $B^\gamma = B$ or $B^\gamma \cap B = \emptyset$ for all $\gamma \in \Gamma$. If $|B| \in \{1, |V|\}$, then B is *trivial*. If B is a block, then $\mathcal{B} := \{B^\gamma : \gamma \in \Gamma\}$ is a *block system* of V. Note that $\mathcal{B}$ is a partition of V which is invariant under the action of Γ. A Schurian coherent configuration $\mathcal{R}$ is *imprimitive* if $\mathrm{Aut}(\mathcal{R})$ is imprimitive, that is $\mathrm{Aut}(\mathcal{R})$ admits a non-trivial block system.

Lemma 2. *If $\mathcal{R}$ is a coherent configuration, then up to equivalence there is a unique factorization of $\mathcal{R}$ into prime factors with respect to the lexicographic product.*

Proof sketch. We first observe that, up to equivalence the lexicographic product is associative. Next, it can be shown that if $\mathcal{R} \cdot \mathcal{S} = \mathcal{R}' \cdot \mathcal{S}'$ then $\mathcal{S} \leq \mathcal{S}'$ or $\mathcal{S}' \geq \mathcal{S}'$, meaning $\mathcal{S}$ is an induced substructure of $\mathcal{S}'$ or vice versa. Finally we observe that if $\mathcal{S} < \mathcal{S}'$ then $\mathcal{R}' \cdot \mathcal{S}'$ is equivalent to $\mathcal{R}' \cdot \mathcal{T} \cdot \mathcal{S}$ for some structure $\mathcal{T}$. □

As for graphs [11], for coherent configurations lexicographic products also preserve ultrahomogeneity.

Lemma 3. *If $\mathcal{R}$ and $\mathcal{S}$ are relational structures, then $\mathcal{R} \cdot \mathcal{S}$ is ultrahomogeneous if and only if both structures $\mathcal{R}$ and $\mathcal{S}$ are ultrahomogeneous.*

3 Algorithms

Let $\mathcal{R} = (V, R_1, R_2, \ldots, R_k)$ be a coherent configuration. For our practical computations and their analysis we assume that $V = [|V|]$ and that we are given $\mathcal{R}$ as a $|V| \times |V|$-*adjacency matrix* $A(\mathcal{R})$ with $A_{i,j} = s$ precisely if $(i, j) \in R_s$.

Definition 4. *For a subset $W \subseteq V$, the* neighborhood partition *$\mathcal{P}_\mathcal{R}(W)$ is the partition of $V \setminus W$ where two elements $i, j \in V \setminus W$ are in the same part if and only if $A(\mathcal{R})_{w,i} = A(\mathcal{R})_{w,j}$ for every $w \in W$.*

3.1 Checking ultrahomogeneity

Lemma 5. *If $(\mathcal{R}, \chi)$ is a colored binary relational structure, then $(\mathcal{R}, \chi)$ is ultrahomogeneous if and only if the following conditions hold for every color class C of $(\mathcal{R}, \chi)$:*

1. *C is an orbit of $\mathrm{Aut}((\mathcal{R}, \chi))$.*
2. *For some (and thus by Part 1 every) $v_c \in C$ the structure $(\mathcal{R}[V(\mathcal{R}) \setminus \{v_c\}], \chi^{v_c})$ is ultrahomogeneous, where χ^{v_c} is a coloring whose color classes form the meet of the neighborhood partition $\mathcal{P}_\mathcal{R}(\{v_c\})$ and the color classes of χ (i.e., it is the coarsest partition which is finer than both of them).*

function `is_ultrahomogeneous`(A, W);
Input : an adjacency matrix A of a coherent configuration $\mathcal{R}$ and a subset $W \subseteq V(\mathcal{R})$
Output: **true** if $\mathcal{R}$ is ultrahomogeneous with respect to W and **false** otherwise

if *there is a part in* $\mathcal{P}_{\mathcal{R}}(W)$ *on which* pwStab(W) *does not act transitively* **then return false**;
$L :=$ a list containing precisely one vertex of every part in $\mathcal{P}_{\mathcal{R}}(W)$;
for $v \in L$ **do**
  if `is_ultrahomogeneous`$(A, W \cup \{v\}) ==$ **false then return false**;
end
return true;

Algorithm 1: Checking if a coherent configuration $\mathcal{R}$ is ultrahomogeneous. The input is the adjacency matrix A of $\mathcal{R}$ and a set of vertices W (default: empty).

Theorem 6. *A coherent configuration $\mathcal{R}$ is ultrahomogeneous if and only if Algorithm 1 returns "*true*" when called on the input (A, W) with $A = A(\mathcal{R})$ and $W = \emptyset$.*

Proof. For $\{w_1, w_2, \dots, w_s\} \subseteq V(\mathcal{R})$ observe that the neighborhood partition $\mathcal{P}_{\mathcal{R}}(\{w_1, w_2, \dots, w_i\})$ is precisely the partition of $\mathcal{R}[V(\mathcal{R}) \setminus \{w_1, w_2, \dots, w_i\}]$ into the color classes with respect to $(((\chi^{w_1})^{w_2})^{\cdots})^{w_i}$ (for the definition of this coloring, see Lemma 5). Recursively applying Lemma 5 yields the theorem. □

Ignoring the running time of basic group theoretic algorithms (i.e., using the Schreier-Sims algorithm to compute point-wise stabilizers), the running time of Algorithm 1 can be bounded using the number of irredundant bases up to equivalence under the group action. However, using some heuristics in particular to deal with permutations of the sequences of chosen points, one can significantly reduce this running time requirement.

3.2 Checking lexicographic products

Lemmas 2 and 3 imply that once we have, up to some order, the number of ultrahomogeneous relational structures that are not a lexicographic product, we can compute the number of all such structures, including the products. We therefore develop a fast algorithm that can discard lexicographic products.

Input : the adjacency matrix $A(\mathcal{R})$ of a coherent configuration $\mathcal{R} = (V, R_1, R_2, \dots, R_k)$ where R_1 is the diagonal of $\mathcal{R}$ and $|R_i| \leq |R_j|$ whenever $i \leq j$
Output: either "not a lexicographic product" or "lexicographic product or not ultrahomogeneous"

if $k \leq 2$ **then return** "not a lexicographic product";
min_$j := 2$;
choose $v_0 \in V(\mathcal{R})$;
for i **from** *1* **to** $k-1$ **do**
  choose $v \in V(\mathcal{R})$ with $(v_0, v) \in R_i$;
  for $w \in V(\mathcal{R}) \setminus \{v_0, v\}$ **do**
    if $A_{v_0 w} \neq A_{vw}$ **then** min_$j :=$ max(min_$j, A_{v_0,w} + 1, A_{v,w} + 1)$;
  end
  if $i + 1 ==$ min_j **then return** "lexicographic product or not ultrahomogeneous";
end
return "not a lexicographic product" ;

Algorithm 2: Check if a coherent configuration is a non-trivial lexicographic product.

Theorem 7. *The output of Algorithm 2 is correct.*

Assuming the relations are already ordered by size, the running time of Algorithm 2 is $O(kn)$ where k is the number of relations (rank) and n the number of vertices. Note that the fact that not even the entire input has to be checked is achieved by leveraging the assumed ultrahomogeneity.

4 Computations

Ultrahomogeneity implies coherence, and there is a complete database of coherent configurations of order at most 34 [6] (see also the paper series of Hanaki and Myamoto [5, 7]). Complete data for 38 is also available.

Our approach for the generation of ultrahomogeneous binary relational structures is to run our ultrahomogeneity test (Algorithm 1) on the configurations.

Thin coherent configurations are omitted in the data base of coherent configurations [6]. Since every transitive thin coherent configuration is ultrahomogeneous and the thin coherent configurations correspond exactly to the transitive permutation groups, they exactly account for the difference.

We used SageMath [13] for our computations. The coherent configurations are given via adjacency matrices. To filter out some of them we have the following observation.

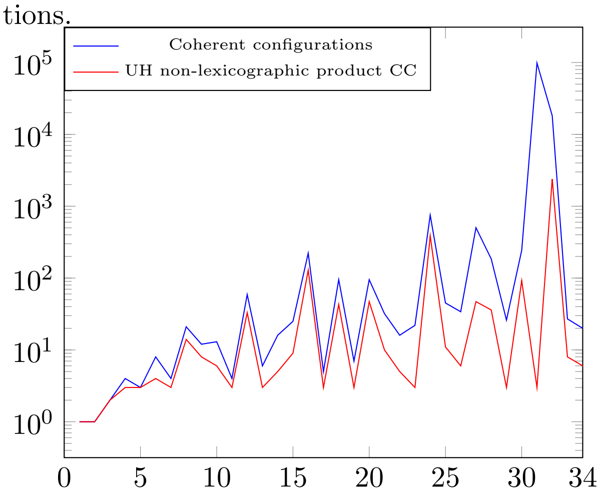
Lemma 8. *If a coherent configuration is ultrahomogeneous, then it is Schurian.*

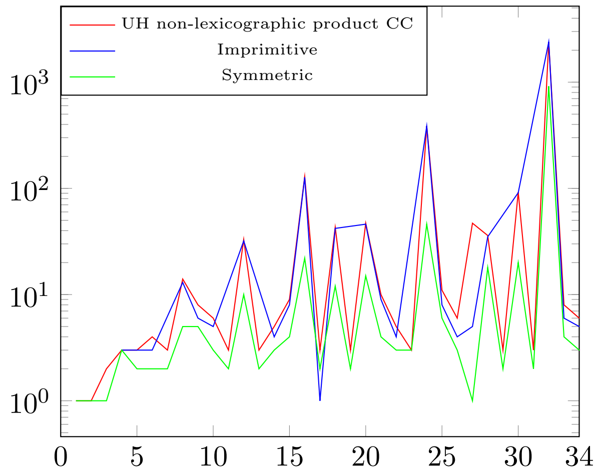
Proof. Suppose $\mathcal{R} = (V, R_1, R_2, \ldots, R_k)$ is an ultrahomogeneous coherent configuration. We argue that $\mathcal{R}' := \mathcal{R}(\mathrm{Aut}(\mathcal{R}))$ is equivalent to $\mathcal{R}$. It is clear that $\mathcal{R}'$ is at least as fine as $\mathcal{R}$, that is, if (v, w) and (v', w') are in the same relation of $\mathcal{R}'$ then they are in the same relation of $\mathcal{R}$. For the other direction, if (v, w) and (v', w') are in the same relation of $\mathcal{R}$ then by ultrahomogeneity there is an automorphism mapping (v, w) to (v', w'), and thus the two pairs are in the same relation of $\mathcal{R}'$. □

Thus we may restrict our attention to Schurian coherent configurations. Using Algorithm 2 we filter out the lexicographic products and then apply Algorithm 1 to the remaining configurations.

The fact that we can limit ourselves to Schurian coherent configurations is crucial since this gives us an alternative for order 31. Rather than considering the 98307 coherent configurations of order 31, we make use of GAP [2] and the *AssociationSchemes* [1] package. By the Galois correspondence, Schurian coherent configurations are in 1:1-relation with 2-closed groups. Hence we first compute the list consisting of the 2-closures of the 12 transitive groups of degree 31. There are 8 resulting Schurian coherent configurations coming from these groups (including one thin coherent configuration), whose adjacency matrices can be obtained using the *AssociationSchemes* package, and then we apply Algorithm 1 to check for ultrahomogeneity. This approach is equivalent to working with the matrices, and in this particular case it reduced the workload significantly. Indeed, it turns out that there are orders for which there are significantly fewer transitive groups, while there are other orders for which there are significantly fewer coherent configurations. The total computation was less than one day on a personal computer (Intel i7 at 2.8 GHz).

Figure 1: Numerical data surrounding ultrahomogeneity of vertex-monochromatic coherent configurations.





In Figure 1 we depict the number of ultrahomogeneous relational structures of order up to 34. On the left side, we present the total number of ultrahomogeneous coherent configurations which are not lexicographic products, compared to the total number of homogeneous coherent configurations. On the right side, we show the number of imprimitive coherent configurations and symmetric coherent configurations within the overall count of ultrahomogeneous coherent configurations.

5 Future work

We generated all ultrahomogeneous edge-colored graphs up to order 34. In particular by the pruning of lexicographic products, our algorithms are comfortably efficient enough to compute the number of ultrahomogeneous graphs in the order ranges in which the coherent configurations are available. However, there is ample room for speeding up the algorithms using additional pruning. Algorithm 1 can be sped up by considering only canonical sequences of points v chosen recursively. More pressing is an analysis of the ultrahomogeneous graphs that are not lexicographic products. Certain other ultrahomogeneity preserving general constructions are known, but the question of whether we can use product structures to provide a concise classification, preferably admitting efficient algorithms, remains.

References

[1] J. Bamberg, A. Hanaki, and J. Lansdown. AssociationSchemes: A GAP package for working with association schemes and homogeneous coherent configurations, Version 3.0.0. 2023.

[2] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.13.0*, 2024.

[3] A. Gardiner. Homogeneous graphs. *J. Combinatorial Theory Ser. B*, 20(1):94–102, 1976.

[4] N. Gill, M. W. Liebeck, and P. Spiga. *Cherlin's conjecture for finite primitive binary permutation groups*, volume 2302 of *Lecture Notes in Mathematics*. Springer, Cham, 2022.

[5] A. Hanaki, H. Kharaghani, A. Mohammadian, and B. Tayfeh-Rezaie. Classification of skew-Hadamard matrices of order 32 and association schemes of order 31. *J. Combin. Des.*, 28(6):421–427, 2020.

[6] A. Hanaki and I. Miyamoto. Classification of association schemes with small vertices. `http://math.shinshu-u.ac.jp/~hanaki/as/`.

[7] A. Hanaki and I. Miyamoto. Classification of association schemes of small order. *Discrete Math.*, 264(1-3):75–80, 2003.

[8] I. Heinrich, E. Kaja, and P. Schweitzer. Finite vertex-colored ultrahomogeneous oriented graphs. In *Graph-Theoretic Concepts in Computer Science - 50th International Workshop, WG 2024, Gozd Martuljek, Slovenia, June 19-21, 2024*, Lecture Notes in Computer Science, 2024. To appear.

[9] I. Heinrich, T. Schneider, and P. Schweitzer. Classification of finite highly regular vertex-coloured graphs. *arXiv preprint arXiv:2012.01058*, 2020.

[10] A. H. Lachlan. Finite homogeneous simple digraphs. In *Proceedings of the Herbrand Symposium Logic Colloquium '81*, pages 189–208. North-Holland Publishing Company, 1982.

[11] A. H. Lachlan. Binary homogeneous structures. I, II. *Proc. London Math. Soc. (3)*, 52(3):385–411, 412–426, 1986.

[12] J. Sheehan. Smoothly embeddable subgraphs. *J. London Math. Soc. (2)*, 9:212–218, 1974/75.

[13] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.3)*, 2024. `https://www.sagemath.org`.