

August 2025

In dieser Ausgabe

- Die neuen Sprecher:innen
- Rückblick 87. Theorietag
- Vorausschau 88. Theorietag
- Gastbeitrag Steffen van Bergerem
- Gastbeitrag Irene Heinrich

Konferenzen 2025/26

Konferenz	Frist '25	Event '25/26
FOCS	03.03.25	14.12.–17.12.25
MFCS	18.04.25	25.08.–29.08.25
Highlights	16.05.25	01.09.–05.09.25
IPEC	27.06.25	17.09.–19.09.25
SODA	14.07.25	11.01.–14.01.26
CSL	15.07.25	23.02.–28.02.26
ITCS	06.09.25	27.01.–30.01.26
STACS	25.09.25	09.03.–13.03.26
ICDT	08.10.25	24.03.–27.03.26
PODS	10.10.25	31.05.–05.06.26
FoSSaCS	16.10.25	11.04.–16.04.26
STOC	Nov 25	22.06.–26.06.26
LICS	Jan 26	20.07.–23.07.26
WG	01.02.26	02.06.–04.06.26
ICALP	Feb 26	07.07.–10.07.26
CCC	Feb 26	03.08.–06.08.26
SAT	TBA	20.07.–23.07.

Theorietage seit 2022

TT	Wo	Wann
88	Augsburg	09.10.–10.10.25
87	Jena	03.03.–04.03.25
86	Düsseldorf	18.04.–19.04.24
85	Bochum	19.09.–20.09.23
84	Hamburg	06.03.23
83	Bonn	10.11.–11.11.22
82	Frankfurt	08.06.–09.06.22

Fachgruppenleitung (seit 2024)

E-Mail an die Fachgruppenleitung

- Sebastian Siebertz (Sprecher)
- Sandra Kiefer (stv. Sprecherin)

Mitgliederzahl (GI): 277

Kostenlos Mitglied in FG-KP werden

Liebe Mitglieder der Fachgruppe Komplexität,

wir präsentieren Ihnen die sechzehnte Ausgabe des Newsletters der GI-Fachgruppe Komplexität, gleichzeitig die erste Newsletteredition unter unserer Leitung. In dieser Ausgabe möchten wir uns kurz vorstellen, Ihnen den Rückblick auf den letzten Theorietag der Fachgruppen Algorithmen, Komplexität und Logik zukommen lassen und zugleich den nächsten gemeinsamen Theorietag mit den Fachgruppen Logik und Algorithmen in Augsburg, organisiert von Pascal Lenzner, ankündigen. Zudem freuen wir uns über Gastbeiträge von Steffen van Bergerem und Irene Heinrich.

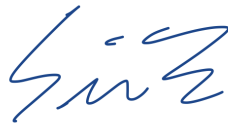
Wenn Sie Interesse an einem kurzen inhaltlichen Beitrag in einem zukünftigen Newsletter haben, melden Sie sich bitte einfach bei **uns**. Wir planen mit Textvorschlägen von 1–2 Seiten Länge.

Ebenfalls wie üblich gilt: Wenn Sie eine Konferenz in der linken Spalte vermissen, melden Sie sich bitte bei **uns**, damit wir die Konferenz für die Zukunft aufnehmen können.

Wenn Sie in die Fachgruppe eintreten möchten, dann ist dies **kostenlos** als assoziiertes Mitglied möglich – auch ohne eine GI-Mitgliedschaft.

Der Newsletter ist auch online von unserer **Webseite** zu beziehen (Publikationen → **Newsletter**).

Nun wünschen wir viel Spaß beim Lesen und hoffen, Sie alle beim nächsten Theorietag treffen zu dürfen!




Sebastian Siebertz und Sandra Kiefer, Sprecher:innen der Fachgruppe Komplexität

Die Fachgruppe Komplexität

Die Fachgruppe Komplexität ist ein Teil der Gesellschaft für Informatik. Sie beschäftigt sich mit komplexitätstheoretischen Fragestellungen. Manche der Themen sind eng gekoppelt an bzw. werden gemeinsam bearbeitet mit anderen Fachgruppen, insbesondere sind dies die **FG Algorithmen** (Thema: Obere Schranken), **FG Automaten und formale Sprachen** (Thema: spezielle Berechnungsmodelle, Abchlusseigenschaften von Klassen) sowie die **FG Logik in der Informatik** (Thema: Komplexität logischer Entscheidungsprobleme, Komplexität des logischen Programmierens, subrekursive Hierarchien).

Vorstellung der neuen Sprecher:innen

Prof. Dr. Sebastian Siebertz leitet seit September 2019 die Arbeitsgruppe *Theoretische Informatik* an der Universität Bremen. Nach seinem Studium der Informatik an der RWTH Aachen promovierte er 2015 an der TU Berlin und war als Postdoc an der TU Berlin, Universität Warschau und Humboldt-Universität zu Berlin. Seine Forschung beschäftigt sich vor allem mit der strukturellen und algorithmischen Graphentheorie, sowie logischen Methoden in der Informatik. Sein Hauptforschungsgebiet ist die Komplexität des Model-Checking-Problems der Prädikatenlogik, sowie deren Erweiterungen.

Prof. Dr. Sandra Kiefer ist Associate Professor of Computer Science an der University of Oxford sowie Tutorial Fellow in Computer Science am Exeter College. Nach ihrer Promotion an der RWTH Aachen, für die sie den Ackermann Award (EACSL Outstanding Dissertation Award) erhielt, war sie Postdoc an der Universität Warschau, Forschungsgruppenleiterin am Max-Planck-Institut für Softwaresysteme sowie Research Fellow an der University of Oxford und Jesus College Oxford. Ihre Forschungsschwerpunkte liegen in der deskriptiven Komplexitätstheorie, der algorithmischen Graphentheorie sowie der Anwendung von Methoden aus diesen Gebieten in Verifikation und maschinellem Lernen.

Impressum

GI Fachgruppe Komplexität

Fachgruppenleitung:

Sebastian Siebertz (Sprecher, ViSdPR),
Sandra Kiefer (stv. Sprecherin).

Web <https://fg-kp.gi.de>

Postalisch

GI-FG Komplexität
Prof. Dr. Sebastian Siebertz
AG Theoretische Informatik
Universität Bremen
Bibliothekstr. 5, MZH
D-28359 Bremen

Mail siebertz@uni-bremen.de

Rückblick: 87. Theorietag

Der 87. Theorietag fand am 3. und 4. März 2025 in Jena als gemeinsamer Workshop der Fachgruppen Algorithmen, Komplexität und Logik statt. Der Workshop fungierte als Auftakt der Konferenz **STACS** im historischen Ambiente der Rosensäle. Eingeladene Vorträge hielten **Heribert Vollmer** (Hannover) zu „The Complexity of Enumerating Satisfying Assignments“ und **Sebastian Wild** (Marburg) zu „Quicksort, Timsort, Powersort - Algorithmic ideas, engineering tricks, and trivia behind CPython’s new sorting algorithm“. Zusätzlich gab es 11 weitere Vorträge aus dem Spektrum von Algorithmen, Komplexität und Logik.

Der Theorietag war sehr gut besucht, es gab 67 Registrierungen und recht internationales Publikum, welches sich üblicherweise selten bei den Theorietagen einfindet. Die Koppelung des Theorietags mit der STACS, das erste Mal praktiziert in Hamburg zur STACS 2023, wurde allgemein begrüßt. Der Theorietag wurde organisiert auf der fachlichen Seite von **Olaf Beyersdorff** und **Christian Komusiewicz** (beide Jena) sowie organisatorisch von **Marlene Gründel** (jetzt TU Wien, Chair of the Local Organisation Committee) und weiteren Mitgliedern der Friedrich-Schiller-Universität Jena.

Mehr Details online auf der [Webseite des Theorietags](#).

Olaf Beyersdorff

Vorausschau: 88. Theorietag

Der 88. Theorietag der Fachgruppe Komplexität findet wie auch sein Vorgänger gemeinsam mit den Fachgruppen **Algorithmen** und **Logik in der Informatik** statt. Die Universität Augsburg heißt die Teilnehmenden am 09. und 10.10.2025 willkommen. Organisatoren und Ansprechpartner sind **Pascal Lenzen** und **Marcus Wunderlich**.

Weitere Informationen entnehmen Sie bitte der [Webseite des Theorietags](#).

Es wird um **Anmeldung** bis **26.09.2025** gebeten. Auch Vortragsvorschläge können eingereicht werden.

Gastbeitrag: Algorithmic Meta Theorems for Supervised Learning

Steffen van Berghem (Humboldt-Universität zu Berlin, steffen.van.berghem@hu-berlin.de)

Algorithmic metatheorems are results that apply to entire families of algorithmic problems. Typically, they state that all problems expressible in a certain logic can be solved efficiently on specific classes of inputs. The archetypical algorithmic metatheorem is due to Courcelle [4]. It states that every graph property expressible in monadic second-order logic (MSO) can be decided in linear time on graph classes of bounded tree-width. The theorem was generalised in [5] to classes of bounded clique-width.

In terms of parameterised complexity, the proof in [5] shows that checking MSO-definable properties is fixed-parameter tractable on classes of bounded clique-width. That is, for every such graph class $\mathcal{C}$, there are a computable function f , a constant c , and an algorithm that decides whether a given sentence $\varphi \in \text{MSO}$ is satisfied by a given graph $G \in \mathcal{C}$ in time $f(\varphi) \cdot n^c$, where n is the number of vertices in the graph. In fact, $c = 1$, so this problem (the *model-checking problem* for MSO) is fixed-parameter linear.

For first-order logic (FO), Grohe, Kreutzer, and Siebertz [6] provided an algorithm for nowhere dense classes $\mathcal{C}$ that checks whether a given sentence $\varphi \in \text{FO}$ is satisfied by a given graph $G \in \mathcal{C}$ in time $f(\varphi, \epsilon) \cdot n^{1+\epsilon}$ for every $\epsilon > 0$. For classes that are closed under taking subgraphs, this result is optimal. More precisely, under the assumption that the complexity classes FPT and AW[*] do not coincide, if a class $\mathcal{C}$ is closed under taking subgraphs and *not* nowhere dense, then checking FO-definable properties on $\mathcal{C}$ is not fixed-parameter tractable. The algorithmic result has been generalised to counting [7] and enumeration problems [10] via algorithms that efficiently count and enumerate the satisfying assignments for FO formulae on nowhere dense classes.

In this contribution, we study the computational complexity of problems that can be seen as converses of the enumeration problem: instead of enumerating satisfying assignments for a given logical formula, we want to reverse-engineer a formula based on a given (partial) list of assignments and corresponding truth values. In the following, we consider these problems in two different settings in the supervised-learning framework introduced by Grohe and Turán [8].

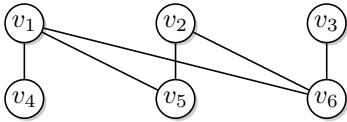


Figure 1. Input graph G in the consistent-learning example.

Consistent learning We start with the conceptionally simpler of the two settings, where we want to find a formula that is consistent with the given examples. Moreover, we start with the logic MSO. Let G be the graph in Figure 1, and consider the sequence $S = ((v_1, +), (v_3, +), (v_4, -), (v_5, -))$. This *training sequence* contains vertices labelled as positive or negative. Our goal is to return an MSO formula $\varphi(x)$ that is satisfied in G if we assign the positive instances v_1 and v_3 to x and not

satisfied if we assign the negative instances v_4 and v_5 to x . In order to prevent simply memorising the sequence, we are given a bound q on the maximum quantifier rank of the target formula and a bound ℓ on the number of vertices that can be used as constants in the formula. Assume, $q = 3$ and $\ell = 1$, so φ may have quantifier rank at most 3 and use at most one vertex as a constant.

Note that the graph G is bipartite, the positive examples are in the same part of the partition as v_1 , and the negative examples are in the other part. Hence, a suitable consistent formula would be $\varphi(x) = \exists Z (\psi_{\text{bipartite}}(Z) \wedge Z(v_1) \wedge Z(x))$, where $\psi_{\text{bipartite}}(Z) = \forall z_1 \forall z_2 (E(z_1, z_2) \rightarrow \neg(Z(z_1) \leftrightarrow Z(z_2)))$ says that the set Z is one part of the bipartition. The formula $\varphi(x)$ has quantifier rank 3 and only uses the vertex v_1 as a constant.

To summarise, the input for the MSO consistent-learning problem is a graph G , a sequence S of vertices from G that are each labelled with either $+$ or $-$, and two integers q and ℓ . The output should be an MSO formula $\varphi(x)$ of quantifier rank at most q using at most ℓ vertices from G as constants such that φ is satisfied in G if a positive example is assigned to x and not satisfied if a negative example is assigned to x . If no such formula exists, an algorithm solving the problem should reject the input. It turns out that, parameterised by q and ℓ , this problem is fixed-parameter linear on classes of bounded clique-width. This coincides with the computational complexity of the MSO model-checking problem on these classes.

Theorem 1 ([3]). *For every class $\mathcal{C}$ of bounded clique-width, there are a computable function f and an algorithm that solves the MSO consistent-learning problem on $\mathcal{C}$ in time $f(q, \ell) \cdot n$.*

We sketch the proof idea. When replacing the ℓ vertices used as constants in a formula by ℓ placeholder variables $y_1, \dots, y_\ell$, up to equivalence, there are only finitely many MSO formulae that we could return, and the number of them only depends on q and ℓ . Hence, to solve the consistent-learning problem, we can simply test all such formulae, and it remains to find the right vertices to set as constants. Testing all combinations would take time n^ℓ , so instead, we proceed as follows. First, we encode the training sequence S into the graph G by having one colour for positive examples and another colour for negative ones. This turns the existence of suitable constants for a candidate formula $\varphi(x)$ into an MSO-definable property, which we can check efficiently. In addition, by building upon the proof techniques from [5], we are also able to extract the right vertices to use as constants if they exist. This solves the MSO consistent-learning problem on graph classes of bounded clique-width.

On the class of all graphs, the MSO model-checking problem is para-NP-hard. As we show in [3], the same holds for the MSO consistent-learning problem.

Furthermore, in the aforementioned example and the description of the problem, we classify vertices of a graph, and we also call this problem the *1-dimensional MSO consistent-learning problem*. In the k -dimensional MSO consistent-learning problem,

lem, the training sequence S consists of k -tuples of vertices that are labelled as positive or negative, and a formula $\varphi(x_1, \dots, x_k)$ returned by an algorithm should have k free variables.

For $k \geq 2$, this problem turns out to be much harder than the 1-dimensional version: even on a class of graphs of clique-width at most 2, the problem is W[1]-hard [3].

However, we also give an algorithm that solves the problem in time $\mathcal{O}((m+1)^{g(c,q,k,\ell)}n^2)$ for a computable function g , where m is the number of examples given in the training sequence, c is the clique-width of the input graph, and k is the dimension of the problem [3]. Hence, for sufficiently short training sequences, the problem is still tractable.

Agnostic PAC learning We now turn to the agnostic probably approximately correct (PAC) setting [9], which is well-known in computational learning theory. Here, we assume an (unknown) probability distribution on $(V(G))^k \times \{+, -\}$. Intuitively, this models real-world scenarios where some instances will be observed with a higher probability than others. In addition, there might not be a definite label for every tuple, but there is rather a probability for each tuple to be classified as positive or negative.

In this setting, instead of having a training sequence as input, an algorithm may use an oracle to draw labelled examples from the distribution. The goal is to find a formula that *generalises well*, that is, with high probability, an algorithm should return a formula that has a small expected error on new examples drawn from the same distribution.

For MSO on graph classes of bounded clique-width, the results of [8] imply that it suffices to draw a constant number of examples from the distribution and then return a formula that minimises the number of errors on these examples. By encoding this constant number of examples into the graph, we can use an analogous approach to the one for the 1-dimensional consistent-learning case, and this even works for higher dimensions k of the agnostic-PAC-learning problem.

Theorem 2 ([3]). *For every graph class $\mathcal{C}$ of bounded clique-width and every $k \geq 1$, the k -dimensional MSO agnostic-PAC-learning problem is fixed-parameter linear on $\mathcal{C}$.*

(A variant of) the agnostic-PAC-learning problem has also been studied for FO instead of MSO. Building upon the model-checking results in [6], we show that the k -dimensional FO agnostic-PAC-learning problem is fixed-parameter tractable on nowhere dense classes [2]. Again, the problem boils down to finding the right vertices to set as constants. For solving this, we use a game characterisation of nowhere dense classes via the so-called Splitter game. In this game, the vertices chosen by one of the two players turn out to be well-suited constants in a formula.

Matching the hardness of FO model-checking in general, we also show that learning on the class of all graphs is hard for the parameterised complexity class AW[*] [2].

Concluding remarks and open questions Focussing on MSO, we have discussed recent results on the parameterised complexity of learning logical formulae. Thereby, we have presented algorithmic metatheorems that show tractability

(and hardness) for families of algorithmic learning problems.

For an overview of further algorithmic results within the learning framework from [8], we refer the reader to [1].

While the parameterised complexity of learning MSO has been studied for the consistent-learning and the agnostic-PAC-learning setting, results for FO so far mostly focus on agnostic PAC learning. On the tractability side, we are not aware of any results for FO consistent learning beyond the ones that carry over from MSO. In particular, for $k \geq 2$, it is open whether k -dimensional FO consistent learning is fixed-parameter tractable on nowhere dense classes. On the hardness side, however, the proof for FO PAC-learning can be adapted to the consistent-learning setting, so consistently learning FO formulae on the class of all graphs is also AW[*]-hard.

References

- [1] Steffen van Bergerem. *Descriptive Complexity of Learning*. PhD thesis, RWTH Aachen University, Germany, 2023.
- [2] Steffen van Bergerem, Martin Grohe, and Martin Ritzert. On the parameterized complexity of learning first-order logic. In Leonid Libkin and Pablo Barceló, editors, *PODS '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12–17, 2022*, pages 337–346. ACM, 2022.
- [3] Steffen van Bergerem, Martin Grohe, and Nina Runde. The parameterized complexity of learning monadic second-order logic. In Jörg Endrullis and Sylvain Schmitz, editors, *33rd EACSL Annual Conference on Computer Science Logic, CSL 2025, February 10–14, 2025, Amsterdam, Netherlands*, volume 326 of *LIPICs*, pages 8:1–8:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025.
- [4] Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990.
- [5] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.
- [6] Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017.
- [7] Martin Grohe and Nicole Schweikardt. First-order query evaluation with cardinality conditions. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2018, Houston, TX, USA, June 10–15, 2018*, pages 253–266. ACM, 2018.
- [8] Martin Grohe and György Turán. Learnability and definability in trees and similar structures. *Theory Comput. Syst.*, 37(1):193–220, 2004.
- [9] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.
- [10] Nicole Schweikardt, Luc Segoufin, and Alexandre Vigny. Enumeration for FO queries over nowhere dense graphs. *J. ACM*, 69(3):22:1–22:37, 2022.

Gastbeitrag: Hochsymmetrische und hochreguläre gefärbte Graphen: Wann erzwingt Regularität bereits die stärkste Form von Symmetrie?

Irene Heinrich (TU Darmstadt, heinrich@mathematik.tu-darmstadt.de)

Alle in diesem Beitrag besprochenen Graphen sind endlich. Sie können sowohl gerichtet als auch ungerichtet sein.

In den letzten Jahren habe ich in verschiedenen Zusammenarbeiten daran geforscht, hochreguläre Objekte vollständig zu klassifizieren und die Frage zu beantworten, wie stark regulär ein gefärbter Graph sein muss, um bereits hochsymmetrisch zu sein. Mit diesem Beitrag möchte ich einen (unvollständigen) Überblick über dieses Thema und die erzielten Ergebnisse geben.

Zunächst stellt sich die Frage: Was ist der Zusammenhang zur Komplexitätstheorie? Graphenisomorphie ist einer der bekanntesten, vielleicht der bekannteste Kandidat dafür, ein *NP-intermediate* Problem zu sein. In verschiedenen Algorithmen für Isomphietests ist es unvermeidbar, hochreguläre Strukturen extra zu behandeln (ein berühmtes Beispiel hierfür ist Babais Quasipolynomialzeitalgorithmus für das Graphenisomorphieproblem [Bab16]), denn auf hochregulären Graphen liefern viele Standardroutinen wenige oder gar keine Informationen. Da in diesem Kontext Knoten- und Kantenfärbungen eine wichtige Rolle spielen, beruht hierauf unser ursprüngliches Interesse, diese hochregulären gefärbten Graphen bestmöglich zu verstehen.

Hochsymmetrische oder -reguläre Strukturen spielen auch eine wichtige Rolle in der Modelltheorie (z.B. als Fraïssé-Limit) und weisen enge Verbindungen zu kombinatorischen Objekten wie projektiven Ebenen oder Blockplänen auf. Auch gibt es eine starke Verzahnung zur Gruppentheorie (insbesondere zur relationalen Komplexität von Gruppen).

Wann ist ein Graph (allgemeiner: eine relationale Struktur) *hochsymmetrisch* (HS)? Mindestens sollten wir verlangen, dass der Graph *transitiv* ist, dass also jeder Knoten in unserem Objekt durch eine Symmetrie auf jeden anderen Knoten abgebildet werden kann (im Vergleich: bei einer Spiegelsymmetrie können wir jeden Punkt auf den gegenüberliegenden Punkt abbilden). Doch wir wollen für unser strengstes Symmetriekonzept sogar noch mehr fordern: Auch je zwei Knoten, die zueinander in derselben Relation stehen wie zwei weitere Knoten sollen durch eine Symmetrie des Gesamtgraphen aufeinander abbildbar sein; wir fordern sogar noch mehr: jeder Isomorphismus zweier induzierter Subgraphen der Größe höchstens k muss sich auf einen Automorphismus des Gesamtgraphen erweitern. Ist das der Fall, so nennen wir den Graphen *k-ultrahomogen*. Falls der Graph sogar *k-ultrahomogen* für jede natürliche Zahl k ist, so nennen wir ihn *ultrahomogen*.

Wann ist ein Graph *hochregulär* (HR)? Ein Graph ist *k-tupelregulär*, falls je zwei isomorphe induzierte Subgraphen der Ordnung höchstens k dieselbe Anzahl an Nachbarknoten im Gesamtgraphen haben. Insbesondere sind 1-tupelreguläre Graphen genau die regulären Graphen und 2-tupelreguläre Graphen sind genau die stark regulären Graphen. Im Fall von gefärbten Graphen werden diese Definitionen so verallgemeinert, dass sie die Knoten- und Kantenfarben und gegebenenfalls auch Richtungen der Kanten zusätzlich respektieren. Man

beachte, dass für jedes feste k die k -Tupelregularität eines Graphen in Polynomialzeit überprüfbar ist.

Zugunsten der Lesbarkeit spreche ich in diesem Beitrag oft von HS und HR Graphen anstatt einen konkreten Parameter k für die Tupelregularität bzw. Ultrahomogenität anzugeben.

Eine kurze Literaturübersicht. Eine sehr empfehlenswerte Zusammenfassung zu ultrahomogenen Graphen und allgemeineren Strukturen ist [Mac11]. Ungerichtete (und ungefärbte) ultrahomogene Graphen wurden unabhängig von Gardiner [Gar76] und von Gol'fand und Klin [GK78] klassifiziert. Lachlan [Lac82] klassifizierte die gerichteten ultrahomogenen Graphen.

Satz 1. (Cameron [Cam80] und Gardiner [Gar76] und Gol'fand und Klin [GK78]) *Ein endlicher schlichter ungerichteter Graph ist genau dann ultrahomogen, wenn er 5-tupelregulär ist.*

Unsere Ergebnisse.

Satz 2. (Stark zusammengefasst und abstrahiert aus [HSS25] und [HKS24b]) *Die folgenden Graphenklassen sind klassifiziert:*

- 4-ultrahomogene knotengefärbte ungerichtete Graphen,
- 5-tupelreguläre knotengefärbte ungerichtete Graphen (siehe auch Abbildung 1),
- ultrahomogene knotengefärbte Orientierungen.

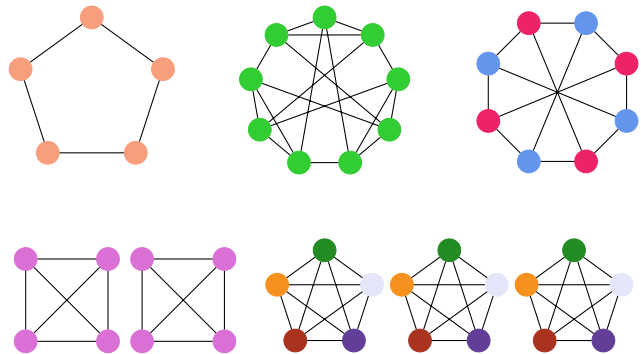


Abbildung 2. Die Bausteine eines 5-tupelregulären knotengefärbten ungerichteten Graphen. Orange: ein Kreis der Länge 5, Grün: der Paley-9-Graph, Rot und Blau: ein spezieller erweiterter Hadamard Graph in genau zwei Farben, Lila: monochromatische disjunkte Vereinigungen von Cliquen derselben Farbe (im Bild: die disjunkte Vereinigung von zwei Cliquen der Größe 4), Bunt: die disjunkte Vereinigung isomorpher Cliquen mit diskreter Färbung (im Bild: die disjunkte Vereinigung drei solcher Cliquen).

Wie auch im ungefärbten Fall erhalten wir als Konsequenz der Klassifikationen der 5-tupelregulären Graphen und der ultrahomogenen Graphen, dass diese übereinstimmen:

Satz 3 ([HSS25]). *Satz 1 gilt auch für knotengefärbte Graphen.*

Methoden. Die im Folgenden beschriebenen Methoden sind alle auch unter weitaus allgemeineren Voraussetzungen als jenen in Satz 3 anwendbar.

- *Hochsymmetrische /-reguläre Nachbarschaften.* Die Schnitte verallgemeinerter Nachbarschaften einer beliebigen Knotenteilmenge bilden selbst wieder hochsymmetrische /-reguläre Graphen. Das bedeutet, dass der Gesamtgraph sich aus vielen miteinander verschränkten hochregulären Graphen zusammensetzt.
- *Hochsymmetrische /-reguläre Farbklassen.* In gefärbten Graphen bildet jede Farbklassse bereits einen hochsymmetrischen oder -regulären Graphen. Wir können also auf Klassifizierungen der monochromatischen Graphen zurückgreifen, um die “Puzzleteile” unseres Graphen zu finden. Es verbleibt, deren Verbindungen miteinander zu verstehen.
- *Blow-ups.* Reduzierte HS/HR Graphen lassen sich zu größeren HS/HR Graphen aufblasen. Es genügt in diesem Sinne, die reduzierten Graphen vollständig zu beschreiben und eine klare und leicht anwendbare Definition der Blow-ups anzugeben.
- *Der allgemeine Erweiterbarkeitssatz.* Wir konnten zeigen (Theorem 2 in [HKS24b]), dass die Struktur eines HS-Graphen bereits vollständig beschreibt, wie weitere Farben an diese eine Farbe verknüpft werden können, ohne die hohe Symmetrie zu beeinträchtigen.
- *Zusammenhänge zu anderen kombinatorischen oder algebraischen Objekten.* Die in unseren Klassifikationen auftretende Ausnahmen stehen oft in enger Verbindung zu (beispielsweise) endlichen Geometrien, Hadamard-Matrizen, symmetrischen Blockplänen oder sporadischen Gruppen.
- *Computergestützte Berechnungen.* Wir nutzen den Rechner, um alle reduzierten HS Objekte (genauer: beliebig knoten- und kantengefärbte gerichtete wie auch ungerichtete Graphen) bis zur Ordnung 34 aufzuzählen und so deren Verhalten zu untersuchen [HKS24a]. Die dabei genutzten Techniken sind auf HR Graphen wie auch auf relationale Strukturen höherer Stelligkeit verallgemeinerbar.

Fazit. Es ist zu erwarten, dass die oben beschriebenen Methoden auch für weitere allgemeinere Graphenklassen dazu führen, deren hochsymmetrische und hochreguläre Graphen zu klassifizieren. Für mich ist die wichtigste Frage, ob wir allgemein für beliebige endliche relationale Strukturen zeigen können, dass ausreichend hohe Tupelregularität Ultrahomogenität erzwingt.

Ich bedanke mich bei allen, die mit mir gemeinsam an diesem Thema geforscht haben: Sofia Brenner, Eda Kaja, Thomas Schneider und Pascal Schweitzer.

Literatur

- [Bab16] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 684–697. ACM, 2016.
- [Cam80] Peter J. Cameron. 6-transitive graphs. *J. Comb. Theory, Ser. B*, 28(2):168–179, 1980.
- [Gar76] A. Gardiner. Homogeneous graphs. *J. Comb. Theory, Ser. B*, 20(1):94–102, 1976.
- [GK78] Ya Gol’fand and Mikhail Klin. On k-homogeneous graphs. *Algorithmic studies in combinatorics (Russian)*, 186:76–85, 1978.
- [HKS24a] Irene Heinrich, Eda Kaja, and Pascal Schweitzer. Computing edge-colored ultrahomogeneous graphs. In Delia Garijo Royo, David Orden Martín, and Francisco Santos Leal, editors, *Proceedings of the Discrete Mathematics Days 2024*, Alcalá de Henares, 2024.
- [HKS24b] Irene Heinrich, Eda Kaja, and Pascal Schweitzer. Finite vertex-colored ultrahomogeneous oriented graphs. In Daniel Král and Martin Milanic, editors, *Graph-Theoretic Concepts in Computer Science - 50th International Workshop, WG 2024, Gozd Martuljek, Slovenia, June 19-21, 2024, Revised Selected Papers*, volume 14760 of *Lecture Notes in Computer Science*, pages 251–265. Springer, 2024.
- [HSS25] Irene Heinrich, Thomas Schneider, and Pascal Schweitzer. Classification of finite highly regular vertex-colored graphs. *SIAM J. Discret. Math.*, 39(1):413–448, 2025.
- [Lac82] Alistair H. Lachlan. Finite homogeneous simple digraphs. In J. Stern, editor, *Proceedings of the Herbrand Symposium Logic Colloquium ’81*, pages 189–208. North-Holland Publishing Company, 1982.
- [Mac11] Dugald Macpherson. A survey of homogeneous structures. *Discret. Math.*, 311(15):1599–1634, 2011.