Power and Limits of the Weisfeiler-Leman Algorithm

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Sandra Kiefer, Master of Science

aus Frankfurt am Main

Berichter: Universitätsprofessor Dr. Martin Grohe
Universitätsprofessor Dr. Pascal Schweitzer
Professor Neil Immerman, Ph.D.

Tag der mündlichen Prüfung: 16. März 2020

Abstract

The Weisfeiler-Leman (WL) algorithm is a fundamental combinatorial technique used to classify graphs and other relational structures. It dates back to the 1960s and has applications in numerous fields of theoretical and practical computer science, such as finite model theory, descriptive complexity theory, propositional proof complexity, and machine learning. Recently discovered links to graph kernels and neural networks demonstrate the persisting importance of the algorithm. It finds particularly prominent use in approaches to the graph isomorphism problem, the task to decide whether two graphs are structurally equivalent or not. Most notably, Babai's breakthrough result from 2016, a quasipolynomial-time graph isomorphism test, relies heavily on a high-dimensional version of the algorithm.

Roughly speaking, for every $k \in \mathbb{N}$, the k-dimensional version of the algorithm (k-WL) iteratively computes a colouring of the vertex k-tuples of the input graph. The larger k, the more powerful k-WL becomes. Still, there is no fixed dimension which decides graph isomorphism in general. On the other hand, it is usually highly non-trivial to tell if and when k-WL distinguishes two particular graphs. This thesis explores that frontier and aims at providing a better understanding of the dynamics, the power, and the limits of the WL algorithm.

Through its connections to many research areas, beautiful and surprising characterisations of k-WL have been discovered. For example, its expressive power is captured by the counting logic C^{k+1} and also by a certain type of Ehrenfeucht-Fraïssé game. This thesis combines the three characterisations of the algorithm to obtain powerful proof techniques.

We first study the number of iterations of the algorithm until stabilisation, which is closely connected to the quantifier depth of a distinguishing C^{k}-formula. Via the construction of infinite graph families, we show that the trivial upper bound $n-1$ on the number of iterations of 1-WL on n-vertex graphs is tight (up to an additive constant of 1). This improves upon the previous best lower bound of $n-\sqrt{n}$. In contrast to this, the situation for 2-WL is quite different: we prove that the trivial upper bound on the iteration number is not tight, not even asymptotically.

As simple examples show, even if the algorithm needs few iterations to compute its output, this does not imply that it distinguishes the given graph from all others. The parameter to study concerning the expressive power of the algorithm is its dimension. In this direction, we present a complete characterisation of the graphs and all relational structures for which 1-WL correctly decides isomorphism.

Proceeding to higher dimensions, we investigate the ability of the algorithm to decompose graphs. By applying the results, we show that 3-WL identifies every planar graph, which drastically improves upon all previously known bounds. Planar graphs constitute the base case in our further consideration of graph classes parameterised by their Euler genus. We show that the WL dimension of every graph is at most linear in its Euler genus. This result is the first explicit parametrisation of the WL dimension by the Euler genus of the input graph.

Zusammenfassung

Der Weisfeiler-Leman (WL) Algorithmus ist eine wichtige kombinatorische Technik, die in den 1960er Jahren entwickelt wurde und vorrangig zur Klassifizierung von Graphen und anderen relationalen Strukturen verwendet wird. Er findet in zahlreichen Bereichen der theoretischen und praktischen Informatik Anwendung, darunter Modelltheorie, deskriptive Komplexitätstheorie, Beweiskomplexität und maschinelles Lernen. Der Algorithmus ist vor allem für seine zentrale Rolle in Graphisomorphie-Algorithmen bekannt. Besonders erwähnenswert ist Babais bahnbrechender Quasipolynomialzeit-Isomorphietest aus dem Jahr 2016, der als ein Kernstück eine hochdimensionale Version des WL Algorithmus beinhaltet.

Für jedes $k \in \mathbb{N}$ berechnet die k-dimensionale Version (k-WL) iterativ eine Färbung der k-Tupel von Knoten des Eingabegraphen. Je größer k, desto ausdrucksstärker ist k-WL. Dennoch gibt es keine Dimension k, die Graphisomorphie im Allgemeinen korrekt entscheidet. Andererseits ist es oft schwierig zu entscheiden, ob und wann k-WL zwei konkrete Graphen unterscheidet. Diese Arbeit untersucht diese Wissensgrenze und zielt darauf, das Verständnis der Funktionsweise, der Ausdrucksstärke und der Grenzen des WL Algorithmus zu verbessern.

Durch seine Verbindungen zu vielen Forschungsbereichen sind elegante und überraschende Charakterisierungen von k-WL entdeckt worden. Beispielsweise entspricht die Ausdrucksstärke von k-WL jener der Zähllogik C^{k+1} und lässt sich mit Gewinnstrategien in einem Ehrenfeucht-Fraïssé-Spiel beschreiben. Diese Arbeit kombiniert die drei Charakterisierungen des Algorithmus zu effektiven Beweistechniken.

Zunächst untersuchen wir die Iterationenzahl des Algorithmus, welche einen direkten Bezug zur Quantorentiefe von C^k-Formeln hat. Mittels einer Konstruktion von unendlichen Graphfamilien zeigen wir, dass die triviale obere Schranke von $n-1$ an die Iterationszahl von 1-WL auf Graphen mit n Knoten (bis auf eine additive Konstante von 1) scharf ist. Dies verbessert die bisherige untere Schranke von $n - \sqrt{n}$. Für 2-WL sieht die Situation komplett anders aus: Wir zeigen, dass die triviale obere Schranke an die Iterationszahl nicht einmal asymptotisch scharf ist.

Selbst wenn k-WL wenige Iterationen zur Berechnung seiner Ausgabe benötigt, gilt keinesfalls, dass er den gegebenen Graphen von allen anderen unterscheidet. Bezüglich seiner Ausdrucksstärke ist viel eher die Dimension des Algorithmus zu betrachten. Hierzu präsentieren wir eine vollständige Charakterisierung der Graphen und aller relationalen Strukturen, für die 1-WL Isomorphie korrekt entscheidet.

Am Übergang zu höheren Dimensionen studieren wir die Fähigkeit des Algorithmus, Graphen zu zerlegen. Wir wenden die Ergebnisse an, um zu zeigen, dass 3-WL jeden planaren Graphen identifiziert, was eine drastische Verbesserung gegenüber allen bisherigen bekannten Schranken darstellt. Planare Graphen stellen den Basisfall in unserer weiteren Betrachtung von Graphklassen dar, die durch ihren Euler-Genus parametrisiert sind. Wir zeigen, dass die WL Dimension jedes Graphen höchstens linear in seinem Euler-Genus ist. Dieses Resultat ist die erste explizite Parametrisierung der WL Dimension durch den Euler-Genus des Eingabegraphen.

Acknowledgements

This thesis is the result of many people seeing and nurturing talents in me. I am thankful to everyone who has played a motivating role in this process.

My first and sincere thanks go to my supervisors and collaborators Martin Grohe and Pascal Schweitzer for giving me the opportunity to pursue research in directions that I enjoy and wish to learn more about.

Pascal Schweitzer introduced me to the post-graduate academic world, taught me a lot about graphs, and has oftentimes given me valuable personal advice on scientific as well as soft skill topics. I am particularly thankful to him for integrating me in his academic network, which led to some of the major results included in this thesis.

During my time at i7, Martin Grohe has given me a lot of freedom to organise my work the way I prefer. He has supported all the opportunities I have been given for my academic career, even if those meant proceeding in unconventional ways. I sincerely thank him for trusting in my abilities and for pushing my skills to formalise my work.

Furthermore, I am grateful to Nicole Schweikardt for being a sort of mentor in my undergraduate studies. She drew my attention to the graph isomorphism problem and convinced me to apply for the position at RWTH Aachen University.

In addition, I feel very lucky to have been invited by Brendan McKay and by Mikołaj Bojańczyk for research visits. It is a pleasure to work with Brendan and the time in Canberra led to some of my favourite results of this thesis. When in Warsaw, I particularly enjoy the inclusive atmosphere in our discussions, in which I have learned a lot about various concepts in theoretical computer science and about academic life.

There are a few people who had an immediate positive effect on the scientific quality of this thesis during the process of writing. I express my profound thanks to Christoph for sharing his amazing Tikz and LaTeX skills with me, for travelling long ways to meet, and for letting me keep his office headphones. I am also grateful to Daniel for a fruitful collaboration, for proof-reading parts of this thesis, and for chocolate afternoons at the office. Moreover, I would like to thank Steffen for patient discussions in the final phase of this thesis, which have improved the structure of quite a few arguments.

Finally, I thank my parents for always wanting the best for me and for supporting me in every possible way.

Contents

1 Introduction

The Weisfeiler-Leman algorithm is a simple combinatorial technique mainly used to classify graphs and other relational structures. Its original version was introduced in 1968 with the purpose of studying symmetries in highly regular graphs [162]. Ever since, the algorithm has played a crucial role in both theoretical and practical approaches to the graph isomorphism problem. Roughly speaking, the classical 2-dimensional algorithm consists in an iterative computation of a colouring of the vertex pairs of the input graph. During the computation, colours containing information about the graph structure are propagated through the graph. The colouring gradually becomes finer until it is stable. This final output can often ,but not always, be used to detect non-isomorphism of two graphs.

Over the decades, more and more connections of the algorithm to other fields of computer science have been discovered, underpinning its position as an elegant algebraic-combinatorial tool for understanding the structure of objects (see, for example, [5, 7, 8, 30, 39, 55, 64, 71, 89, 124, 134, 138], further discussed below).

This thesis has a twofold objective. Firstly, it is meant to provide in a didactic way an overview of what is known and what is beautiful about the Weisfeiler-Leman algorithm. Secondly, it shall formalise the notions of the *power* and the *limits* of the Weisfeiler-Leman algorithm and advance our intuition about these. In the following two sections, framed in a brief survey of what is known about the algorithm, we motivate its study in more detail. This introduction concludes with a discussion of the scientific contribution in the context of related work as well as an outline of this thesis.

1.1 Motivation

The beauty of the Weisfeiler-Leman algorithm lies in the combination of a simple concept with a vast and ever-growing range of applications in theoretical and practical computer science, some of which we discuss in this section.

The original, 2-dimensional algorithm is due to Weisfeiler and Leman and was designed to detect symmetries in graphs [162]. Iteration by iteration, the algorithm refines a colouring of vertex pairs of the input graph in an isomorphism-invariant fashion until the computed colouring is stable with respect to the refinement criterion. The very basic concept behind it is a simple procedure that propagates structural information through a graph by iteratively recomputing colours of vertices based on their neighbourhoods. This procedure is accordingly also known as Naïve Vertex Classification or the 1-dimensional Weisfeiler-Leman algorithm and has been rediscovered many times. One of the oldest published versions can be found in [132].

Independently, Babai and Mathon [9] as well as Immerman and Lander [90] developed a generalisation of the low-dimensional combinatorial procedures to higher dimensions, which resulted in the k-dimensional Weisfeiler-Leman algorithm for $k \in \mathbb{N}$. Instead of colouring pairs of vertices, the k-dimensional algorithm colours k-tuples based on "adjacent" k-tuples with respect to a certain notion of adjacency. See [91] for a short survey and a pseudocode of an implementation. Babai's purpose was to render the algorithm more powerful with respect to the structural properties it can detect in a graph, in order to use it in approaches to the graph isomorphism problem. Immerman and Lander defined their generalisation in the context of descriptive complexity theory, which has also fostered a lot of work in the field (see, for example, [64, 89, 138]).

In the following subsection, we recapitulate the graph isomorphism problem as one of the most prominent applications of the Weisfeiler-Leman algorithm, in particular since the publication of Babai's quasipolynomial-time graph isomorphism test [12]. With a focus on logics, we discuss the various connections of the algorithm to other fields in the second subsection.

1.1.1 The Graph Isomorphism Problem

One of the most prominent and natural open problems from theoretical computer science is the graph isomorphism problem (GI). It consists in the task to decide whether two given input graphs are structurally equivalent or not. The answer is YES if there is a bijection from the first to the second vertex set that preserves the edge relation, and NO otherwise. If the graphs have the same vertex sets, this means that the second is obtained from the first after a suitable permutation of the vertices. With graphs being a wide-spread model for relations between objects, GI naturally comes into play whenever relational structures are to be compared or classified. For example, applications range from the identification of chemical molecules to the analysis of social networks or links between websites.

The bijection serves as a witness for the isomorphism of the two graphs. Given a compact encoding of such a certificate mapping, it is straight-forward to verify efficiently and deterministically that it indeed constitutes a bijection which preserves the edge relation. Therefore, GI is contained in the complexity class NP, the class of problems which can be solved nondeterministically in polynomial time.

The quest for an efficient graph isomorphism test was started almost half a century ago [97]. In 1977, Read and Corneil published *The Graph Isomorphism Disease*, a paper whose title reflects the fascinating effect the problem has had on researchers that start tackling graph isomorphism [143]. As an evidence for its infectious nature, the computational complexity of the problem was intensively investigated, in particular between 1980 and the beginning of the 1990s. The first lower bound on the complexity of GI was published in 2003, proving its hardness for the complexity class NC^1 [93, 94]. Until today, the best known result implying a lower bound on the computational complexity of the problem remains that, under logarithmic space many-one reductions, GI is hard for the complexity class DET, therefore in particular for NL [155]. However, GI remains one of the two computational problems from Garey and Johnson's 1979 list of natural problems whose complexity is yet unknown [51]: it is still neither known whether there exists a polynomial-time algorithm that solves GI nor whether it is NP-hard. Note that both might be true at the same time, since PTIME could equal NP.

If the general graph isomorphism problem turns out not to be solvable in polynomial time, there are graph classes whose isomorphism problem is neither isomorphism-complete (i.e. as hard as GI) nor decidable in polynomial time (see [136]). It is also possible that GI itself is NP-intermediate, i.e. neither NP-complete nor in PTIME, assuming PTIME $\neq$ NP. Indeed, there are quite a few strong arguments against its NP-completeness: GI is known to be contained in co-AM [53, 54], which in combination with its NP-completeness would yield a collapse of the polynomial-time hierarchy to its second level [26]. The validity of the collapse is highly doubted. Also, counting the number of isomorphisms between two graphs is as difficult as deciding whether they are isomorphic, a property that distinguishes GI from many known NP-complete problems [125, 154].

The best known upper bound on the complexity of GI up to date is Babai's quasipolynomial-time isomorphism test, showing the problem to be solvable in time $2^{O((\log n)^c)}$ for a fixed constant $c \in \mathbb{N}$ [12]. Babai published his algorithm on the preprint server arXiv on the first day of a Dagstuhl seminar dedicated to bring together researchers studying theoretical and practical approaches to GI [13]. Consequently, at Dagstuhl, Babai gave a detailed presentation of the proof. It combines algorithmic strategies from group theory with graph-theoretic approaches to the isomorphism problem. Notably, Babai employs a high-dimensional version of the Weisfeiler-Leman algorithm as an important subroutine.

The recent quasipolynomial-time algorithm is actually the strongest indicator against the NP-completeness of GI. In fact, the NP-completeness would imply that every problem in NP is solvable in quasipolynomial time, since the problem would be polynomial-time reducible to GI. In particular, this would yield a subexponential-time algorithm for 3-SAT, which refutes the Exponential Time Hypothesis [92]. Still, we seem to be far from a polynomial-time algorithm that solves the general graph isomorphism problem.

The question to decide whether two graphs are isomorphic or not is also of practical relevance, for example for symmetry reduction when solving satisfiability and verification problems (see, for instance, [147]). In practice, despite the fact that GI is not known to be in PTIME, software applications can cope well with most of the relevant instances. Again, as a subroutine, all competitive isomorphism solvers, such as `Nauty` and `Traces` [127], `Bliss` [95, 96], `saucy` [37, 36], and `conauto` [122], exploit the 1-dimensional Weisfeiler-Leman algorithm, which is also called Colour Refinement.

Colour Refinement can be implemented to run in time $O((m+n)\log n)$, where n is the order of the input graph and m is its number of edges (cf. [32, 126, 139]). Under some very reasonable assumptions, this running time is asymptotically optimal [19]. The procedure is applied to speed up algorithms in other fields, too, for example in the context of graph kernels in machine learning [149] or static program analysis [118]. It is known that Colour Refinement distinguishes two graphs if and only if there is a fractional isomorphism between them [142, 152, 153]. Furthermore, it can be modified to allow for effective dimension reduction in linear programming [65].

Maybe somewhat surprisingly in view of its numerous applications, there are very simple examples of non-isomorphic graphs that show the limits of Colour Refinement, i.e. graphs that are not distinguished by the 1-dimensional Weisfeiler-Leman algorithm. As mentioned above, its generalisation to higher dimensions colours k-tuples of vertices, taking into account colours of "adjacent" k-tuples with respect to a certain notion of adjacency. Informally speaking, it thus propagates more information through the graph than when just considering colours of vertices and arcs. Therefore, the higher the dimension, the more powerful becomes the algorithm with respect to the set of pairs of non-isomorphic graphs it can distinguish from each other. The k-dimensional Weisfeiler-Leman algorithm can be implemented to run in time $O(n^{k+1}\log n)$ on n-vertex input graphs [90], but for practical purposes, due to large memory consumption, these implementations are often excessive already for $k=2$.

For some time, there was hope that a sufficiently high fixed dimension of this combinatorial procedure would actually solve the general graph isomorphism problem. This would be a proof that graph isomorphism is decidable in polynomial time. However, in their seminal result [30], Cai, Fürer, and Immerman showed that even the high-dimensional versions of the algorithm have their limits: by their construction, for every $k \in \mathbb{N}$, there are pairs of non-isomorphic graphs of order $O(k)$ that are not distinguished by the k-dimensional Weisfeiler-Leman algorithm. Thus, as an isomorphism test, the algorithm is incomplete. Nonetheless, the situation can change when restricting the input to come from a certain graph class. The idea behind this general approach in the context of the graph isomorphism problem is that structural properties which the members of the considered class have in common might be exploited in order to find an efficient algorithm that correctly decides isomorphism graphs from the class. With a strong restriction of the input graphs, we may lose isomorphism completeness, i.e. it will not be possible to draw direct conclusions about the computational complexity of the general graph isomorphism problem. Still, solving special classes of instances of the isomorphism problem may provide valuable insights for the general challenge.

For many natural graph classes, polynomial-time isomorphism tests have been discovered, for instance, for the class of planar graphs [84], all classes of graphs with excluded minors [140], and, subsuming both classes, all classes of graphs with excluded topological subgraphs [68]. There is also an extensive body of work on the isomorphism problem for graph classes characterised by certain parameters. Here, for instance, polynomial-time isomorphism algorithms are known for the classes of graphs of bounded eigenvalue multiplicity [15, 47], graphs of bounded Euler genus [45, 129], graphs of bounded degree [70, 123], graphs of bounded colour class size [10, 50], graphs of bounded tree width [23], and graphs of bounded rank width [69, 72]. For random graphs, isomorphism can be decided in polynomial expected time [14].

Similarly, even though no fixed dimension of the Weisfeiler-Leman algorithm identifies every graph (i.e. distinguishes it from every other, non-isomorphic graph), we can study the dimension needed to identify graphs from a particular class. Following Grohe [64, Definition 18.4.3], we say that a graph class $\mathcal{C}$ has *Weisfeiler-Leman dimension k* if k is the smallest integer such that all graphs in $\mathcal{C}$ are identified by the k-dimensional Weisfeiler Leman algorithm. If there is no such $k \in \mathbb{N}$, we define the Weisfeiler-Leman dimension of $\mathcal{C}$ to be ∞. The Weisfeiler-Leman dimension is a measure for the inherent structural (and descriptive) complexity of a graph or graph class. Its boundedness yields a polynomial-time algorithm that identifies the considered graphs. Since every graph G is identified by the $(|G|-1)$-dimensional Weisfeiler-Leman algorithm, it is more interesting to consider infinite graph classes instead of classes with only finitely many isomorphism types in this context.

For some of the aforementioned graph classes, in addition to the polynomial-time isomorphism tests, their Weisfeiler-Leman dimensions have been determined. For example, it is not very difficult to see that already the 1-dimensional Weisfeiler-Leman algorithm serves as a complete isomorphism test on the class of trees [41, 90]. Moreover, a graph is asymptotically almost surely identified by Colour Refinement, which means that the fraction of graphs of size n which are not identified by Colour Refinement tends to 0 when n tends to infinity [14].

The 2-dimensional algorithm solves the isomorphism problem for cographs and interval graphs ([42], see also [108, 116]) and asymptotically almost surely for random regular graphs [25, 114]. Also for some of the parameterised graph classes, bounds on their Weisfeiler-Leman dimensions have been proved, for instance, for graphs of bounded tree width [67], graphs of bounded Euler genus [61], and graphs of bounded rank width [69]. Most notably, as was shown in Grohe's seminal work [62], for every graph class that excludes some graph as a minor, a sufficiently high-dimensional Weisfeiler-Leman algorithm correctly decides graph isomorphism (see also [64]). However, most of the aforementioned results do not state explicit bounds and the ones that would follow from the proofs cannot be assumed to be anywhere near tight.

1.1.2 Further Connections

In 2018, the 50-year-anniversary of the Weisfeiler-Leman algorithm was celebrated at the *Symmetry vs Regularity* conference on algebraic graph theory in Pilsen [151], which can be considered an indication for persisting importance of the algorithm in theoretical and practical computer science. Even apart from the use as a pure isomorphism test, the algorithm has proven itself to be worth profound study. For example, its practical applications reach beyond isomorphism solvers into the area of machine learning (see, for instance, [1, 75, 106, 149]). As shown in [134], the 1-dimensional Weisfeiler-Leman algorithm essentially corresponds to graph neural networks. Its iteration number equals the depth of a graph neural network that outputs the stable vertex colouring of the underlying graph with respect to Colour Refinement. The authors of [134] propose an extension of graph neural networks based on the k-dimensional Weisfeiler-Leman algorithm (see also [133]).

On the theoretical side, the 2-dimensional algorithm is closely related to the algebraic theory of coherent configurations [11, 35]. For example, by analysing coherent configurations systematically, Fuhlbrück, Köbler, and Verbitsky provide an efficient test to check whether the 2-dimensional Weisfeiler-Leman algorithm is able to distinguish a given input graph of colour class size at most 4 from all others [46]. There is also a close connection between the 2-dimensional algorithm and matrix multiplication. Namely, it is possible to execute t iterations of the algorithm by performing t matrix multiplications over a certain ring (see [17, Section 5]). However, it is also well-known that, using randomisation, these multiplications can be performed over the integers (see, for example, [148], Sections 2.9.2 and 2.9.3), yielding a running time of $O(n^{\omega})$, where $\omega < 3$ is the coefficient for matrix multiplication (see [22] for details on fast multiplication of matrices).

A very fruitful connection of the Weisfeiler-Leman algorithm to logics was established by Immerman and Lander [90] and Cai, Fürer, and Immerman [30]. Independently of Babai and Mathon, Immerman and Lander defined the k-dimensional Weisfeiler-Leman algorithm and proved that two graphs are distinguished from each other by the k-dimensional Weisfeiler-Leman algorithm if and only if there is a formula in the logic C^{k+1} that is satisfied by exactly one of the two graphs. Here, C^{k+1} denotes the $(k+1)$-variable fragment of the extension of first-order logic by counting quantifiers of the form $\exists^{\geq p}x$. Such bounded-variable fragments of first-order logic have played an important role in finite model theory and descriptive complexity theory since the 1980s (see [58] for a survey). They have been studied extensively, since their model-checking problem can be decided in polynomial time [86, 158], and for C^k and its non-counting version L^k, the equivalence problem is solvable in time $n^{O(k)}$, i.e. in polynomial time for constant k (see [90]). It is also known that equivalence for

C^k and for L^k is PTIME-complete [60]. The fragment C^2 is particularly interesting because its satisfiability problem is known to be decidable [57] and the complexity of the decision problem is well-studied [141]. For more results about this logic, also see the survey by Grädel and Otto [56].

As an illustration of the powerful link of the Weisfeiler-Leman algorithm to descriptive complexity theory and logics, consider the famous open problem which asks for a logic that captures PTIME (on the class of all structures) [33, 74]. The question is closely related to the canonisation problem and therefore also to the graph isomorphism problem (see [138] for details). By Fagin's Theorem, a negative answer to the question would imply PTIME $\neq$ NP [44]. A positive answer would mean that there is a logic L such that the model-checking problem for all L-sentences can be decided in polynomial time and furthermore, all PTIME-decidable properties are definable in L (for a precise definition, see [64, Definition 3.1.9]). As an example, the Immerman-Vardi Theorem says that least fixed-point logic LFP captures PTIME on the class of ordered relational structures [87, 157]. But the general quest for a logic for PTIME has not yet succeeded.

Immerman proposed fixed-point logic with counting FP+C as a candidate logic to capture PTIME [88] and in fact, Hella, Kolaitis, and Luosto proved that FP+C captures PTIME on almost all structures [78]. However, the CFI-construction from [30] yields that in FP+C, $\Omega(n)$ variables are needed to identify every single n-vertex graph. Therefore, in particular, none of the finite-variable FO-fragments C^k identifies every graph. Since all of the graph isomorphism instances obtained via the construction in [30] can be solved in polynomial time, this shows that none of the logics C^k captures PTIME on graphs. By the correspondence between C^k and the Weisfeiler-Leman algorithm, we obtain a similar negative conclusion about the power of the k-dimensional Weisfeiler-Leman algorithm to detect polynomial-time decidable properties of graphs.

Still, as already discussed above, the situation can change when restricting ourselves to certain graph classes. For example, Grohe showed that FP+C captures PTIME on the class of planar graphs [59] and, more generally, on graphs of bounded Euler genus [61]. For graphs parameterised by their treewidth, this was shown in [67].

Subsequent work yields that for 3-connected planar graphs and for graphs of bounded treewidth, it is possible to restrict the quantifier depth (or equivalently, the number of iterations that the Weisfeiler-Leman algorithm performs until it terminates) to be polylogarithmic, which translates to parallel isomorphism tests [73, 159]. For graphs in general, recent results give improved lower bounds [21, 113]. Extending the results for planar graphs in the direction of dynamic complexity, Mehta shows that isomorphism of 3-connected planar graphs is contained in DynFO+ [128].

Various other aspects of the Weisfeiler-Leman algorithm and its relation to logic have been studied in detail in recent years, see, for instance, [5, 6, 49].

Another correspondence exists to a certain type of Ehrenfeucht-Fraïssé game, namely the bijective pebble game, which was first introduced by Hella [77]. The k-dimensional Weisfeiler-Leman algorithm distinguishes two graphs if and only if the player Spoiler has a winning strategy in the bijective pebble game played with $k + 1$ pairs of pebbles on the two graphs [30]. The connection is even more precise: the number of iterations of the Weisfeiler-Leman algorithm needed to distinguish the two graphs from each other corresponds to the quantifier depth of a distinguishing C^{k+1}-formula and also to the number of rounds that Spoiler requires to win the pebble game with $k + 1$ pebble pairs. The trifold equivalence between the Weisfeiler-Leman algorithm, logics, and games has proven to be very useful when investigating the power of the algorithm and has spawned very interesting developments (see, for instance, [64, 89, 138]).

Over the years, probably to some extent due to the persisting interest in the graph isomorphism problem and the associated tools, further characterisations of the Weisfeiler-Leman algorithm in seemingly unrelated areas have been discovered. For example, Atserias and Maneva [7] showed that the dimension of the algorithm needed to distinguish two graphs corresponds to the level of the Sherali-Adams relaxation of the natural integer linear program for graph isomorphism (also see [71, 124]). Their result was the starting signal for work relating the Weisfeiler-Leman algorithm to semidefinite programming [8, 135] and Gröbner basis approaches to graph isomorphism testing [20, 55]. A recent element in the Weisfeiler-Leman puzzle characterises the expressive power of the k-dimensional algorithm in terms of homomorphism counts of graphs of treewidth k [39].

In a related direction of research, it has been studied which graph properties the Weisfeiler-Leman algorithm can detect, which may become particularly relevant in the graph-learning framework. In this context, Fürer [49] as well as Arvind, Fuhlbrück, Köbler, and Verbitsky [3] obtained results concerning the ability of the algorithm to detect and count certain subgraphs.

Via its surprising connections to numerous other fields, the study of the algorithm has an individual strong justification. By obtaining more insight into the structural information that the algorithm is able to detect, we can expect its already quite dense web of links to be woven even further.

1.2 Contribution

The purpose of this thesis is to advance our understanding of the *power* as well as the *limits* of the Weisfeiler-Leman algorithm. The two key parameters which we shall investigate to this end are the *iteration number until stabilisation* and the *dimension* of the algorithm. They have precise correspondences in other fields of theoretical computer science. With respect to the dimension, there are two possible perspectives to be taken: we can either fix a dimension k and investigate the expressive power and limits of the k-dimensional Weisfeiler-Leman algorithm or we can fix a graph or a graph class and study the dimension of the algorithm that is needed to decide isomorphism for such a graph and an arbitrary second one.

In the first part of the thesis, we are concerned with the computational complexity of the algorithm. Although we know that its complexity can be bounded by $O(n^{k+1} \log n)$ [90], with respect to bounds on the iteration number, for a long time, only the trivial upper bound of $n^k - 1$ and a linear lower bound of $O(n)$ were known [48]. Note that the trivial upper bound holds for every repeated partitioning of a set of size n^k. By the correspondence established by Cai, Fürer, and Immerman, the number of iterations needed for the k-dimensional algorithm to compute distinct colourings on two non-isomorphic input graphs esssentially equals the quantifier depth of a distinguishing formula in C^{k+1}. Furthermore, the iteration number is crucial for the parallelisation of the algorithm [73, 109].

For the 1-dimensional Weisfeiler-Leman algorithm, the best published lower bound on its iteration number on n-vertex graphs is $n - O(\sqrt{n})$ [113]. We improve this bound to $n-2$, which by the trivial upper bound is tight up to an additive constant of 1. We obtain our bounds via an empirical approach. More precisely, we have designed a procedure that enables a systematic generatation of graphs G of order at most 64 that obey certain constraints (to render the procedure tractable) and on which Colour Refinement takes $|G| - 1$ iterations to stabilise. Our analysis of the results yields compact string representations of graphs with low vertex degrees that take $n-1$ iterations to stabilise. Using these encodings, we are able to provide infinite families with $n - 1$ Colour Refinement iterations until stabilisation. We modify those graphs in order to provide for all $n \in \mathbb{N}_{\geq 10}$ graphs with $n - 2$ Colour Refinement iterations.

For all constant $k \geq 2$, the lower bound $\Omega(n)$ on the iteration number of the k-dimensional Weisfeiler-Leman algorithm is an implicit result in [60]. An explicit linear lower bound is due to Fürer [48]. Up to date, no improvements over this result have been found. Still, with our result for $k = 1$ in mind, one could expect the trivial upper bound of $n^k - 1$ is tight also for $k \geq 2$. Via the link to logics, Berkholz and Nordström [21] show a lower bound of $n^{\Omega(k/\log k)}$ for the number of iterations of the k-dimensional Weisfeiler-Leman algorithm to distinguish finite relational structures over a universe of size n, whenever $k < n^{0.01}$. The fact that this bound is very close to the trivial upper bound $n^k - 1$ supports the hypothesis that the latter is actually tight.[1]

However, we show that this is not the case for $k = 2$: we provide the first proof that the trivial upper bound is not even asymptotically tight. To obtain our bound of $O(n^2/\log n)$, we define a 2-player game on the input graph, which allows us to decouple the causal dependencies that happen when executing the 2-dimensional

[1] Additionally, Berkholz's result in [18] yields the tightness of the lower bound of $n^{\Omega(k)}$ for a related logic fragment, namely the k-variable existential-positive fragment of FO.

Weisfeiler-Leman algorithm. While the first player assumes a role of an adversary even stronger than the Weisfeiler-Leman algorithm, the second player repeatedly rectifies the graph to maintain a certain consistency for the colouring of the graph. Defining vertex colour classes with sizes beyond a certain threshold to be large, we can first bound the number of iterations in which colour classes are concerned that interact with large vertex colour classes. We then show that the total number of iterations dealing exclusively with small vertex colour classes is linear. Our result led to further study of the iteration number of the 2-dimensional algorithm, which resulted in an improved upper bound of $O(n \log n)$ [120]. A benefit of our work is that it also yields a linear upper bound on the iteration number for graphs of bounded colour class size. Since the bound Fürer proved in [48] also uses graphs of bounded colour class size, for such graphs, lower and upper bounds match asymptotically. Another positive side-effect of our result is that it translates to bounds on the quantifier depth in the non-counting version L^3 of C^3.

In the second part of the thesis, we focus more on the expressive power of the algorithm. We provide a complete characterisation of the graphs and, more generally, all finite relational structures for which the 1-dimensional Weisfeiler-Leman algorithm constitutes a complete isomorphism test. Recall that a graph is identified by the 1-dimensional algorithm if and only if it is definable in the logic C^2. To prove the correctness of our classification, we employ explicit constructions that solve the inversion problem and the canonisation problem for the logic C^2 over graphs. A celebrated result by Otto shows that the inversion problem and the canonisation problem for C^2 can be solved in polynomial time [137]. As a by-product, our direct constructions for the characterisation show that the inversion problem for C^2 can actually be solved in linear time. To this end, we make use of circulant and doubly-circulant graphs. With these, we observe that every C^2-equivalence class contains a graph whose C^2-partition classes are the vertex orbits.

In the graphs that are identified by the 1-dimensional Weisfeiler-Leman algorithm (or, equivalently, definable in the logic C^2), there is only one special case that may appear within a colour class in the colouring computed by the 1-dimensional Weisfeiler-Leman algorithm, namely a cycle of size 5. However, for finite relational structures, there are 7 different special cases, which are of sizes 3, 4, 5, and 6. Our classification theorem describes how these may be combined to form structures that are identified by C^2. We show that one can decide in almost linear time whether a relational structure is identified by the 1-dimensional Weisfeiler-Leman algorithm.

Due to the nature of our classification, we obtain further insights about the structure of identified graphs. For instance, we show that in time $O((n+m)\log n)$, it is possible to decide whether an undirected graph is identified by the 1-dimensional Weifeiler-Leman algorithm. Also, if an undirected graph is identified by C^2, then the C^2-partition classes of the vertices are the vertex orbits of the automorphism group of the graph. This corollary is neither true when considering general relational structures nor when considering higher dimensions of the Weisfeiler-Leman algorithm, which correspond to the logics C^k with $k > 2$. For C^2, we also conclude that every vertex-coloured version of an identified graph is identified by C^2 as well. This statement holds for finite relational structures, too, but is again not true for C^2 with $k > 2$.

As these insights indicate, the situation for $k = 1$ is particular among the dimensions of the Weisfeiler-Leman algorithm. For higher dimensions, we obtain several negative results. We first argue why a complete characterisation of the graphs identified by the k-dimensional algorithm for $k \geq 2$ is out of sight. The triangular graphs form an infinite non-trivial class of strongly regular graphs which are identified by the 2-dimensional algorithm, implying that any classification result would have to include non-trivial infinite families [34, 79].

Then we provide examples of graphs identified by the 3-dimensional Weisfeiler-Leman algorithm for which conclusions analogous to the ones mentioned above do not hold. More specifically, on our examples, even the logic C^k with k linear in the size of the graph does not determine the vertex orbit partition. This yields graphs which the k-dimensional Weisfeiler-Leman algorithm identifies, but for which the algorithm fails on some vertex-coloured versions. The existence of these graphs highlights an important fact. Even if a graph G is identified by the k-dimensional algorithm, it is not clear that it is possible to take advantage of that in order to canonise G. In fact, the crucial property required for the approach hinted at there to be successful is that all vertex-coloured versions of G need to be identified as well. Indeed, if this property holds, then a standard recursive individualisation approach canonises the graph G. It would suffice to show that for all vertex-coloured versions of G, the vertex orbits are determined by the logic. However, with a slight alteration of our construction, we obtain a graph G that is identified and whose vertex orbits are determined by the k-dimensional Weisfeiler-Leman algorithm, but for some vertex-coloured versions of G, the algorithm fails to determine the vertex orbits.

Instead of attempting to answer the question which graphs are identified by a fixed dimension of the Weisfeiler-Leman algorithm, we can fix a class of graphs and investigate which dimension is necessary to identify them. To this end, we first provide a powerful reduction from the identification problem for graphs from a minor-closed graph class to the same problem for 3-connected graphs from the class. It is not difficult to see that the $(k + 2)$-dimensional Weisfeiler-Leman algorithm distinguishes vertex sets of size k that split a graph from vertex sets that are not separators. For our reduction, we show that, actually, the k-dimensional Weisfeiler-Leman algorithm is able to detect k-separators in graphs. We use this insight to prove that the 2-dimensional Weisfeiler-Leman algorithm implicitly computes the decomposition of a graph into its 3-connected components. In general, decompositions are popular and powerful tools to work with graphs whose structure is complex (see e.g. [81, 82, 83, 84]).

Equipped with our reduction, we consider the structural complexity of minor-closed graph classes with respect to the Weisfeiler-Leman algorithm. We investigate the dimension needed to identify graphs stemming from such a class. The first graph class that we treat are planar graphs. Isomorphism for planar graphs is known to be decidable in linear time [84] and also in logarithmic space [38]. For this class, Grohe had shown that the necessary dimension of the Weisfeiler-Leman algorithm is bounded, however not specifying the precise bound. He generalised this result to all graph classes with excluded minors [62]. By a careful analysis of Grohe's proof, Redies was able to show that the dimension k can in fact be chosen to be 14 ([144], see also [64, Subsection 18.4.4]). Feeling that this is far from optimal, Grohe asked in his book [64, Subsection 18.4.4] and also at the 2015 Dagstuhl meeting on the Graph Isomorphism Problem [13] for a tight k. We show that, in fact, the Weisfeiler-Leman dimension of the class of planar graphs is at most 3. For many graphs, we can apply a Spring Embedding Theorem due to Tutte to show that they are identified by the 3-dimensional Weisfeiler-Leman algorithm. With a precise analysis of the remaining exceptions, we then bring the upper bound for all graphs down to 3, which implies that the Weisfeiler-Leman dimension of the class of planar graphs is either 2 or 3. On the algorithmic side, we obtain a very easy routine to check isomorphism of 3-connected planar graphs.

Having determined the Weisfeiler-Leman dimension of planar graphs up to an additive constant of 1, the natural follow-up project is a generalisation to graphs that can be embedded into arbitrary surfaces. As mentioned above, it is known that the Weisfeiler-Leman dimension of graphs of bounded Euler genus is finite ([61], see also the journal version [62]). However, from these works, no explicit good linear upper bounds on the precise dimension can be derived. The proof of [62] yields a bound that is quadratic in the Euler genus of the input graph. The proof of [61] might yield a linear bound, but in that case only with a very large constant factor. In a quite involved approach that is completely different to the ones in [61] and [62], we obtain a much better bound and, in particular, the first explicit parametrisation of the Weisfeiler-Leman dimension by the Euler genus of the input graph.

Apart from the references to previously conducted research, this work contains only research results to which I contributed significantly. It is based on some new and up to date unpublished work (see Chapter 4) as well as the following publications.

- Sandra Kiefer, Pascal Schweitzer, and Erkal Selman. Graphs identified by logics with counting. In *Proceedings of MFCS 2015*, pages 319–330, 2015 [105]
- Sandra Kiefer and Pascal Schweitzer. Upper bounds on the quantifier depth for graph differentiation in first order logic. In *Proceedings of LICS 2016*, pages 287–296, 2016 [102]
- Sandra Kiefer and Pascal Schweitzer. Upper bounds on the quantifier depth for graph differentiation in first-order logic. *Logical Methods in Computer Science*, 15(2), 2019 [103]

- Sandra Kiefer and Daniel Neuen. The power of the Weisfeiler-Leman algorithm to decompose graphs. In *Proceedings of MFCS 2019*, 45:1–45:15, 2019 [99]
- Sandra Kiefer, Ilia N. Ponomarenko, and Pascal Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. In *Proceedings of LICS 2017*, pages 1–12, 2017 [100]
- Sandra Kiefer, Ilia N. Ponomarenko, and Pascal Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. *Journal of the ACM*, 66(6):44:1–44:31, 2019 [101]
- Martin Grohe and Sandra Kiefer. A linear upper bound on the Weisfeiler-Leman dimension of graphs of bounded genus. In *Proceedings of ICALP 2019*, 117:1–117:15, 2019 [66]

In Chapter 3, the presented statements are mainly known results, which I, however, formalised and proved myself. Chapter 4 is based on joint computational work and discussions with Brendan McKay. I developed a great part of the presented theoretical framework. The formalised statements and proofs, in particular for Theorem 4.13, were written solely by me.

The results in Chapter 5 were obtained together with Pascal Schweitzer [102, 103]. My major contribution lies in important insights and proofs about the strategies of the players of the game as well as in a significant portion of the concepts and their formalisation regarding the treatment of small vertex colour classes.

Regarding Chapter 6, the presented results are due to a collaboration with Pascal Schweitzer and Erkal Selman [105]. One of my ideas led to the discovery of the 1-factorisations that we use. Moreover, I particularly contributed to the generalisation of our findings to finite relational structures. As a result, the classification from Theorem 6.26 was obtained in a research marathon with Pascal Schweitzer after the Summer Festival of Computer Science at RWTH Aachen University in 2015. My observation that the determination of orbits does not necessarily imply an efficient canonisation algorithm led to the contents of Section 3.2. All illustrations that are also contained in the joint publication [105] or its arXiv version [104] have been designed and implemented by Erkal Selman.

Chapter 7 is based on publications with Daniel Neuen and with Ilia Ponomarenko and Pascal Schweitzer [99, 100, 101]. The structure of this chapter as well as a very large portion of the presented concepts, statements, and proof techniques have been developed by me. In particular, being the only author involved in both publications, I composed the contents of Section 7.4, which contain improved and stronger versions of statements from [101]. Chapter 8 is the second part of the joint work with Ilia Ponomarenko and Pascal Schweitzer [100, 101]. The lengthy case distinctions in Section 8.2 are again the results of numerous whiteboard discussions with Pascal Schweitzer.

Chapter 9 builds on concepts and results from [64] and was developed in joint discussions with Martin Grohe. The contents were published in [66]. I contributed many of the technical ingredients and formalisations as well as the general outline and a major part of the proofs that are necessary for the statement of the main lemma, which is Lemma 9.28.

1.3 Outline

The thesis is structured as follows. In Chapter 2, we provide the relevant concepts from graph theory and logics. In Chapter 3, we explain the various dimensions of the Weisfeiler-Leman algorithm in detail. We give some basic insights about their functionality and limits. We finish the chapter with a brief overview of the connection between the Weisfeiler-Leman algorithm, logic, and games, which we use in various places in the course of the thesis.

The first parameter of the algorithm we consider is the iteration number. Here, Chapter 4 recapitulates the known lower bounds with respect to the 1-dimensional Weisfeiler-Leman algorithm. Then we prove our new lower bound on the iteration number of the 1-dimensional Weisfeiler-Leman algorithm via explicit constructions of infinite graph families.

We investigate the iteration number of the 2-dimensional algorithm in Chapter 5. We first introduce the game that allows us to consider the algorithm from a sequential point of view. Then we deduce our improved upper bound on the iteration number, which also holds when we exclude counting quantifiers.

Next, we study the Weisfeiler-Leman dimension as the second important parameter of the algorithm. In Chapter 6, we characterise the graphs and relational structures that the 1-dimensional algorithm identifies. We provide an efficient procedure to decide whether a graph is identified. We close the study of Colour Refinement with a discussion of properties which are particular about the 1-dimensional algorithm.

Chapter 7 is the most technical one in this thesis. It serves as a preparation for the results in the subsequent chapters: we reduce the problem of determining the Weisfeiler-Leman dimension of a minor-closed graph class to the analysis of the 3-connected graphs in the class. The surprising fact is that this reduction is actually performed by the 2-dimensional Weisfeiler-Leman algorithm. To this end, we first show in a schematic study that the 2-dimensional algorithm detects 2-separators in a graph. As an intermediate step, we reduce the general problem of the determination of the Weisfeiler-Leman dimension to 2-connected graphs. Finally, we perform the reduction to 3-connected graphs.

In Chapter 8, we apply the obtained reduction to show that the 3-dimensional Weisfeiler-Leman algorithm identifies every planar graph. We show that for many graphs, we can apply Tutte's Spring Embedding Theorem to identify them. All graphs for which this is not possible are so-called “exceptions”. We analyse them individually to conclude the proof.

Finally, Chapter 9 generalises the results from planar graphs to arbitrary graphs parameterised by their Euler genus. After a detailed introduction to general surface topology and the specific tools we need for the proof, we proceed via an induction depending on whether the given graph has a so-called simplifying patch or not. Interestingly, the case of an absence of simplifying patches is simpler.

We conclude the thesis in Chapter 10 with an outlook that places the provided scientific contribution in the context of possible future projects.

2 Preliminaries

We introduce the notation and general concepts used in this thesis, as well as some basic facts upon which the presented results build.

2.1 General Notation

We let $\mathbb{N}$ denote the set of positive integers, i.e. the set $\{1, 2, \dots\}$. For $m \in \mathbb{N}$, we set

$$\mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\} \qquad \text{and} \qquad \mathbb{N}_{\geq m} \coloneqq \{n \in \mathbb{N} \mid n \geq m\}.$$

Following the notation from [64], for $m, n \in \mathbb{N}_0$, we further let

$$[m, n] \coloneqq \{\ell \in \mathbb{N}_0 \mid m \leq \ell \leq n\} \qquad \text{and} \qquad [n] \coloneqq [1, n].$$

The *cardinality* or *size* of a set S is the number of elements it contains and we write $|S|$ for it. Furthermore, for every $k \in \mathbb{N}_0$, we let $\binom{S}{k} \coloneqq \bigl\{S' \subseteq S \bigm| |S'| = k\bigr\}$.

For two arbitrary sets S, T, we write $S \mathbin{\dot\cup} T$ to denote the *disjoint union* of S and T. More precisely, even if $S \cap T \neq \emptyset$, every element in $S \mathbin{\dot\cup} T$ stems from either S or T (but not from both) and thus, it holds that $|S \mathbin{\dot\cup} T| = |S| + |T|$. We formally define $S \mathbin{\dot\cup} T \coloneqq \{(x, 0) \mid x \in S\} \cup \{(x, 1) \mid x \in T\}$, but we usually just write x for $(x, 0)$ or $(x, 1)$ if the precise origin of the element is irrelevant for the purpose.

A *partition* of S is a set Π of non-empty sets such that $\bigcup_{M \in \Pi} M = S$ and for all $M, M' \in \Pi$ with $M \neq M'$, it holds that $M \cap M' = \emptyset$. For two partitions Π and Π' of the same set S, we say that Π' is *finer* than Π (or Π' *refines* Π) if every element of Π' is a (not necessarily proper) subset of an element of Π. We write $\Pi \succeq \Pi'$ (and equivalently $\Pi' \preceq \Pi$) to express that Π' is finer than Π. Concurrently, we say that Π is *coarser* than Π'. If it holds that $\Pi \succeq \Pi'$ and $\Pi' \succeq \Pi$, we denote this by $\Pi \equiv \Pi'$ and we call Π and Π' *equivalent*. The partition $\bigl\{\{s\} \mid s \in S\bigr\}$ is called the *discrete partition* of S. The partition $\{S\}$ is the *unit* partition of S.

A *multiset* is a collection of objects which may occur more than once in the collection. We can think of a multiset as a set that allows repetitions of elements. This way, sets are special cases of multisets. We use the notation $\{\!\{\dots\}\!\}$ for multisets. The *multiplicity* of an element x in a multiset S is the number of times x appears in S. The *cardinality* of a multiset is the sum of all multiplicities of its elements.

Example 2.1. *In the multiset* $\{\{1,1,4,8,9,9,9\}\}$*, the number* 1 *has multiplicity 2 and the number 9 has multiplicity 3, whereas 4 and 8 have multiplicity 1, and 3 has multiplicity 0. The cardinality of the multiset is 7.*

We use $(\ldots)$ to denote *tuples*, i.e. ordered multisets. The *length* of a tuple is the cardinality of the underlying multiset. For $k \in \mathbb{N}_0$, a k-tuple is a tuple of length k. An *arc* is a 2-tuple. In an arc (u,v), we call u the *head* and v the *tail* of the arc.

2.2 Graphs

We provide a brief introduction to the basic graph-theoretic notions and facts we need in the context of this work. For more details on graphs, we refer the reader to [40] and [130].

2.2.1 Basic Terminology

A *directed graph* is a pair $G = \big(V(G), E(G)\big)$, where $V(G)$ is a finite set (called *vertex set* of G) and $E(G)$ is a binary relation on $V(G)$ (called *edge set* of G). A *vertex* of G is an element from $V(G)$ and an *edge* is an element from $E(G)$. A *loop* in G is an edge of the form (v,v) for some $v \in V(G)$.

An *undirected graph* is a directed graph G for which $E(G)$ is irreflexive and symmetric. For edges of undirected graphs, we also write $\{v,w\}$ or vw instead of (v,w) and (w,v), i.e. we regard them as elements from $\binom{V(G)}{2}$.

The *order* of a (directed or undirected) graph G is $|G| \coloneqq |V(G)|$. If G is undirected, we let $\|G\| \coloneqq |E(G)|$ (where $E(G)$ is considered as a subset of $\binom{V(G)}{2}$).

When speaking about *graphs*, if not explicitly stated otherwise, we mean undirected graphs. In particular, graphs contain no loops. For example, for every $n \in \mathbb{N}$, we define the *complete graph* K_n to be the graph with vertex set $[n]$ and edge set $\binom{[n]}{2}$.

A *matching* in G is a set $E \subseteq E(G)$ such that $e \cap e' = \emptyset$ holds for all $e, e' \in E$. A *perfect matching* is a matching E such that for every $v \in V(G)$, there is an $e \in E$ with $v \in e$.

Two (directed or undirected) graphs G and H are *isomorphic*, denoted by $G \cong H$, if there is an edge-respecting bijection between the vertex sets $V(G)$ and $V(H)$, i.e. there is a bijective mapping $\sigma\colon V(G) \to V(H)$ such that

$$(v,w) \in E(G) \iff \big(\sigma(v), \sigma(w)\big) \in E(H)$$

holds for all $v, w \in V(G)$. Such a mapping σ is an *isomorphism* from G to H. An isomorphism from G to G is an *automorphism* of G. The set of automorphisms of

G equipped with concatenation forms a group $\mathrm{Aut}(G)$, which acts on $V(G)$ and on $E(G)$. For a vertex $v \in V(G)$, the *(vertex) orbit* of v is the set $\{\sigma(v) \mid \sigma \in \mathrm{Aut}(G)\}$. The *(vertex) orbit partition* of $V(G)$ is the partition of $V(G)$ whose elements are the orbits of $\mathrm{Aut}(G)$. A set $V \subseteq V(G)$ is called a *block* of $\mathrm{Aut}(G)$ if for every $\sigma \in \mathrm{Aut}(G)$, it holds that $V \cap \sigma(V) \in \{\emptyset, V\}$, i.e. if every automorphism of G maps V onto itself or onto a set that is disjoint from V.

If, for two vertices $v \neq w$ in an undirected graph G, it holds that $\{v, w\} \in E(G)$, we say that v and w are *adjacent* and that v and w are *incident* to the edge $\{v, w\}$. For a set $V \subseteq V(G)$, we let the *neighbourhood* of V in G be the set $N^G(V) := \{w \mid w \in V(G) \setminus V, \exists v \in V\colon \{v, w\} \in E(G)\}$. The elements in $N^G(V)$ are the *neighbours* of V. Here, and in similar notations, we omit the superscript ^{G} if G is clear from the context. Furthermore, for every $v \in V(G)$, we let $N(v) := N(\{v\})$ and set $N[v] := N(v) \cup \{v\}$. The *degree* of v is $\deg(v) := |N(v)|$.

For two (directed or undirected) graphs G and H, the graph H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. In this case, we write $H \subseteq G$ and let $N^G(H) := N^G\big(V(H)\big)$. If, additionally, $E(H) = \big(V(H)\big)^2 \cap E(G)$, then H is an *induced subgraph* of G. For sets $V, V' \subseteq V(G)$, we write $G[V]$ for the unique induced subgraph of G with vertex set V and $G[V, V']$ for the graph with vertex set $V \cup V'$ and edge set $E(G) \cap \big\{\{v, v'\} \,\big|\, v \in V, v' \in V'\big\}$.

A *clique* in G is a set $V \subseteq V(G)$ such that $G[V]$ is a complete graph. A *k-clique* is a clique of order k. An *independent set* in G is a set $V \subseteq V(G)$ such that $E(G[V]) = \emptyset$.

If there is a $k \in \mathbb{N}_0$ such that $\deg(v) = k$ holds for every $v \in V(G)$, then the graph G is *k-regular.* A graph G is *bipartite (on bipartition (P, Q))* if $V(G) = P \cup Q$ with $P \cap Q = \emptyset$ and P and Q are independent sets in G. If, additionally, $\big\{\{v, w\} \mid v \in P, w \in Q\big\} = E(G)$, the graph G is *complete bipartite.* We also use the notation $G = (P, Q, E(G))$ for a bipartite graph G on bipartition (P, Q).

For $m, n \in \mathbb{N}$, we let $K_{m,n}$ be the *complete bipartite graph* with vertex set $[m + n]$ and edge set $\big\{\{i, j\} \,\big|\, i \in [m], j \in [m + 1, m + n]\big\}$.

For $k, \ell \in \mathbb{N}_0$, a *(k, ℓ)-biregular graph (on bipartition (P, Q))* is a bipartite graph on bipartition (P, Q) such that for every $v \in P$, it holds that $|N(v)| = k$, and for every $w \in Q$, it holds that $|N(w)| = \ell$. A *biregular* graph is a graph G for which there are $P, Q \subseteq V(G)$ and $k, \ell \in \mathbb{N}_0$ such that G is (k, ℓ)-biregular on bipartition (P, Q).

Let G be a graph and let Π be a partition of $V(G)$. We say that Π is *equitable* if, for all $P, Q \in \Pi$ and all $v, v' \in P$, it holds that $|N(v) \cap Q| = |N(v') \cap Q|$. This is equivalent to all graphs $G[P]$ being regular and all graphs $G[P, Q]$ being biregular.

For a set X (not necessarily a subset of $V(G)$), we let $G - X \coloneqq G[V(G) \setminus X]$. For a graph H, we let $G - H \coloneqq G - V(H)$, and we denote by $G \cup H$ the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$. Similarly, $G \mathbin{\dot\cup} H$ denotes the graph with vertex set $V(G) \mathbin{\dot\cup} V(H)$ and edge set $E(G) \mathbin{\dot\cup} E(H)$.

A *graph class* is a set of graphs that is closed under isomorphism. We denote graph classes by calligraphic letters, e.g. $\mathcal{G}$, $\mathcal{H}$. A graph class $\mathcal{G}$ is *hereditary* if it is closed under taking induced subgraphs, i.e. for every $G \in \mathcal{G}$, every induced subgraph of G is also contained in $\mathcal{G}$.

2.2.2 Connectivity

For $n \in \mathbb{N}_0$, a *path* of *length* n is a graph that is isomorphic to the graph P_n defined via $V(P_n) \coloneqq \{v_0, \dots, v_n\}$ and $E(P_n) \coloneqq \big\{\{v_{i-1}, v_i\} \bigm| i \in [n]\big\}$. In this case, the graph is *a path from* (the vertex corresponding to) v_0 to (the vertex corresponding to) v_n. In the path, the vertices $v_1, \dots, v_{n-1}$ are *internal* and the vertices v_0 and v_n are the *endvertices* of the path.

For a graph G, a *path in* G is a subgraph of G that is a path of some length. Two paths P, P' in G are *internally disjoint* if no internal vertex of P is a vertex of P' and vice versa.

For $n \in \mathbb{N}_{\geq 3}$, a *cycle* of *length* n is a graph that is isomorphic to the graph C_n defined via $V(C_n) \coloneqq \{v_1, \dots, v_n\}$ and $E(C_n) \coloneqq \big\{\{v_{i-1}, v_i\} \bigm| i \in [2, n]\big\} \cup \big\{\{v_n, v_1\}\big\}$. A *cycle in* G is a subgraph of G that is a cycle. A cycle in G is *chordless* if it is an induced subgraph of G. The graph G is *acyclic* if there is no cycle in G.

A *segment* of a path or a cycle Q is a path P such that $P \subseteq Q$. For vertices v, w of a path P, we denote the unique segment of P with endvertices v and w by vPw.

The *distance* between vertices $u, v \in V(G)$ is denoted by $\mathrm{dist}^G(u, v)$ and defined as the minimal length of a path from u to v in G. Again, we drop the superscript ^{G} if it is clear from the context.

For $n \in \mathbb{N}_0$, a *walk* of *length* n in G is a sequence $v_0, \dots, v_n$ with $v_i \in V(G)$ for every $i \in [0, n]$ and $(v_i, v_{i+1}) \in E(G)$ for every $i \in [0, n-1]$. That is, a walk is a generalised path in which vertices can occur more than once.

If G is non-empty and contains for every $\{u, v\} \in \binom{V(G)}{2}$ a path from u to v, then G is *connected.* Otherwise, G is *disconnected.* A *connected component* of G is an inclusion-maximal connected subgraph of G.

A *forest* is an undirected acyclic graph. A *tree* is a connected forest.

For $k \in \mathbb{N}$, the graph G is *k-connected* if it holds that $|G| > k$ and for every $V \subseteq V(G)$ with $|V| < k$, the graph $G - V$ is connected. A *separator* of G is a set $S \subseteq V(G)$ for which there are vertices $u, v \in V(G) \setminus S$ such that there is a path from u to v in G, but none from u to v in $G - S$. A *k-separator* is a separator S of order k. A *cut vertex* is a 1-separator, and a *separating pair* is a 2-separator.

A 2-*connected component* of G is a set $V \subseteq V(G)$ such that the graph $G[V]$ is connected, has no cut vertex, and V is maximal with respect to inclusion. The concept of a *3-connected component* of a graph G is more complex, since such a component are not necessarily a subgraph of G (in fact, it is just a minor of G). We do not rely heavily on this concept and provide all the necessary intuition about it in Chapter 8. For a precise definition, we refer the reader to [64, Chapter 8].

2.2.3 Minors and Embeddability

For a set $W \subseteq V(G)$, let G/W be the graph obtained from G by identifying all vertices in W with each other and eliminating loops and parallel edges. We usually denote the vertex of G/W representing the set W by w. Formally, G/W is the graph with vertex set $V(G/W) := V(G - W) \cup \{w\}$ and edge set $E(G/W) := E(G - W) \cup \big\{\{v, w\} \bigm| v \in N^G(W)\big\}$. In accordance with [64], for a subgraph $B \subseteq G$ whose connected components are $A_1, \ldots, A_m$, we define

$$G/B := G/A_1/\cdots/A_m$$

and the resulting graph is well-defined, since it does not depend on the order of the connected components $A_1, \ldots, A_m$. The graph H is a *contraction* of the graph G if there is a subgraph $G' \subseteq G$ such that $H = G/G'$. We say that H is a *minor* of G if H is a contraction of some subgraph of G. Intuitively, this means that H can be obtained by contracting edges in some subgraph of G.

We briefly recapitulate some basic notions from surface topology. Further aspects are introduced in Chapter 9. For more background on embeddings of graphs into topological spaces, we refer the reader to [130] and [40, Appendix B].

Let $\boldsymbol{S}$ be a topological space. A *simple curve* in $\boldsymbol{S}$ is a homeomorphic image of the real interval $[0, 1]$, equipped with the usual topology. An *embedded graph* in $\boldsymbol{S}$ is a pair $G = (V, E)$, where V is a finite subset of $\boldsymbol{S}$ (the *vertex set*), and E is a set of simple curves in $\boldsymbol{S}$ the *edge set*) such that for all $\boldsymbol{e} \in E$, both endpoints and no internal point of $\boldsymbol{e}$ are in $V(G)$, and any two distinct $\boldsymbol{e}, \boldsymbol{e}' \in E$ have at most one endpoint and no internal points in common. That is, the edges of G do not cross in $\boldsymbol{S}$.

If $G = (V, E)$ is a graph embedded in $\boldsymbol{S}$, we denote by $\boldsymbol{G}$ the subset of $\boldsymbol{S}$ consisting of all points that are either vertices of G or contained in an edge. The *underlying graph* of G is the graph with vertex set $V(G)$ and edge set $\{\boldsymbol{e} \cap V \mid \boldsymbol{e} \in E\}$. We often do not distinguish explicitly between an embedded graph G and its underlying graph. The *faces* of G are the arcwise connected components of the space $\boldsymbol{S} \setminus \boldsymbol{G}$. For every face $\boldsymbol{f}$ of G, there is a subgraph $B \subseteq G$ such that the (topological) boundary $\boldsymbol{bd}(\boldsymbol{f})$ of $\boldsymbol{f}$ in $\boldsymbol{S}$ is precisely $\boldsymbol{B}$. We call B a *facial subgraph* of G.

A graph G is *planar* if there is an isomorphism from G to the underlying graph of a graph embedded in $\boldsymbol{R}^2$, the real plane $\mathbb{R}^2$ considered as a topological space. Informally, this is equivalent to the statement that G can be drawn in the plane without edge crossings. The following result due to Wagner provides a useful tool to prove that a graph is planar. It is a version of Kuratowski's theorem (see [115]) with minors instead of topological minors.

Theorem 2.2 ([160]). *Let G be a graph. Then G is planar if and only if neither $K_{3,3}$ nor K_5 is a minor of G.*

A *plane* graph is a graph embedded in $\boldsymbol{R}^2$. By Whitney's Theorem [163], if the underlying planar graph G of a plane graph is 3-connected, its facial cycles are exactly the chordless non-separating cycles of G. Also, every 3-connected planar graph has, up to homeomorphism, a unique embedding into $\boldsymbol{R}^2$.

2.3 Coloured Graphs

An *edge-coloured graph* (G, λ) is a graph G with a function $\lambda\colon \big\{(u,u) \mid u \in V(G)\big\} \cup \big\{(u,v) \mid \{u,v\} \in E(G)\big\} \to \mathcal{C}$, where $\mathcal{C}$ is some set of colours. For the colouring function λ, we assume that the set of colours of loops and the set of colours of other arcs are disjoint, that is, we have

$$\big\{\lambda(u,u) \mid u \in V\big\} \cap \big\{\lambda(u,v) \mid u \neq v, \{u,v\} \in E(G)\big\} = \emptyset.$$

We interpret $\lambda(u,u)$ as the vertex colour of u and sometimes just write $\lambda(u)$ for $\lambda(u,u)$. For $\{u,v\} \in E(G)$, we interpret $\lambda(u,v)$ as the colour of the arc from u to v. In particular, it may be the case that $\lambda(u,v) \neq \lambda(v,u)$, that is, the two orientations of an (undirected) edge $\{u,v\}$ may receive different colours. Nevertheless, throughout this thesis, we require for all considered colourings λ that $\lambda(v_1,u_1) = \lambda(v_2,u_2)$ if and only if $\lambda(u_1,v_1) = \lambda(u_2,v_2)$ for all vertices u_1,u_2,v_1,v_2 (where we set $\lambda(u,v) = \bot$ if (u,v) is not an edge). We say the colouring respects *converse equivalence*.[1] In Section 3.2, we argue why this is actually not a restriction.

Recall that graphs are undirected if not explicitly stated otherwise, even though we allow distinct colours for the two orientations of an undirected edge. Furthermore, we treat every uncoloured graph as a monochromatic coloured graph.

A *vertex-coloured graph* is the special case of an edge-coloured graph where all edges receive the same colour, say, 1, that is, $\lambda(u,v) = 1$ for all $u \neq v$ with $\{u,v\} \in E(G)$.

[1] Also in the context of coherent configurations, which are closely related to the 2-dimensional Weisfeiler-Leman algorithm, this is always required [31].

When we refer to *coloured graphs* in this thesis, we usually mean edge-coloured graphs. To simplify the notation, we often do not mention the colouring explicitly and just denote an edge-coloured graph by G.

Two coloured graphs (G, λ) and (G', λ') are *isomorphic*, denoted by $(G, \lambda) \cong (G', \lambda')$, if there is a colour-preserving isomorphism σ from G to G', i.e. $\lambda(v, w) = \lambda'\big(\sigma(v), \sigma(w)\big)$ for all $v, w \in V(G)$ with $\{v, w\} \in E(G)$.

For a (possibly coloured) graph G and a sequence of vertices $v_1, \dots, v_\ell$, we write $G_{(v_1,\dots,v_\ell)}$ to denote the graph resulting from individualising every v_i, i.e. by assigning every v_i for $i \in [\ell]$ a unique fresh colour. We set $G_v := G_{(v)}$. When comparing two graphs with individualised vertices $G_{(v_1,\dots,v_\ell)}$ and $G'_{(v'_1,\dots,v'_\ell)}$, we assume that for $i \in [\ell]$, the two vertices v_i and v'_i have the same colours.

For a set $W \subseteq V(G)$, recall the definition of G/W from Section 2.2.3: for a fresh vertex w, the graph G/W has the vertex set $V(G/W) = V(G - W) \cup \{w\}$ and the edge set $E(G/W) = E(G - W) \cup \big\{\{v, w\} \,\big|\, v \in N^G(W)\big\}$. Additionally, if G has the colouring λ, then we assign to G/W the colouring function λ', where $\lambda'(w, w) = \emptyset$ and $\lambda'(u, v) = \lambda(u, v)$ and $\lambda'(u, w) := \{\!\{\lambda(u, w') \mid w' \in W\}\!\}$ and $\lambda'(w, v) := \{\!\{\lambda'(w', v) \mid w' \in W\}\!\}$ for all $u, v \in V(G - W)$.

Let G be a graph and let $M := \big\{(u, u) \,\big|\, u \in V(G)\big\} \cup \big\{(u, v) \,\big|\, \{u, v\} \in E(G)\big\}$. Every edge colouring $\lambda\colon M \to \mathcal{C}$ induces a partition $\pi(\lambda)$ of M: for $c \in \lambda(M)$, we call $\lambda^{-1}(c) \subseteq M$ a *colour class* of G. Since we are only interested in the colour classes induced by the colourings we consider, and not in the actual colours, we often do not distinguish in our notation between a colour c and the corresponding colour class of 2-tuples.

A *vertex colour class* is a colour class only consisting of tuples of the form (v, v) with $v \in V(G)$. Similarly, an *edge colour class* is a colour class that consists only of tuples of the form (v, w) with $\{v, w\} \in E(G)$. As the term "vertex colour class" already implies, we implicitly identify every vertex tuple (u, u) with the corresponding vertex u.

Let (G, λ) be a coloured directed graph. The *in-neighbourhood* of a vertex $v \in V(G)$ with respect to a set of colours $\mathcal{C}' \subseteq \mathcal{C}$ is the set

$$N^-_{\mathcal{C}'}(v) := \big\{u \,\big|\, u \in V(G), (u, v) \in E(G), \lambda(u, v) \in \mathcal{C}'\big\}.$$

Likewise, the *out-neighbourhood* of v with respect to $\mathcal{C}'$ is the set

$$N^+_{\mathcal{C}'}(v) := \big\{u \,\big|\, u \in V(G), (v, u) \in E(G), \lambda(v, u) \in \mathcal{C}'\big\}.$$

The *colour in-degree* of v with respect to the set $\mathcal{C}'$ is the number $d^-_{\mathcal{C}'}(v) := |N^-_{\mathcal{C}'}(v)|$. The *colour out-degree* is defined analogously. When talking about *colour degrees*, we mean both colour in-degrees and colour out-degrees.

All notions introduced in this subsection extend naturally to directed graphs.

2.4 Relational Structures and Mathematical Logic

We interpret the logics that we treat over coloured (directed or undirected) graphs and, more generally, over finite relational structures. We introduce the necessary machinery here.

All graphs in this subsection may be directed or undirected. As a notational convention, we shall use x, y, z for variables in first-order logic, whereas u, v, w denote graph vertices.

A *vocabulary* is a finite set of *relation symbols*, each of which is attributed with an *arity* $k \in \mathbb{N}$. Let τ be a vocabulary. A *(relational) structure* $\mathcal{A}$ over τ is a tuple consisting of a *universe* $V(\mathcal{A})$ and a relation $R(\mathcal{A}) \subseteq (V(\mathcal{A}))^k$ for every relation symbol $R \in \tau$, where k is the arity of R. We say R is *interpreted* by $R(\mathcal{A})$.

For two relational structures $\mathcal{A}$ and $\mathcal{B}$ over τ, a *homomorphism* from $\mathcal{A}$ to $\mathcal{B}$ is a mapping $\sigma\colon V(\mathcal{A}) \to V(\mathcal{B})$ such that for every relation symbol $R \in \tau$ and every tuple $(a_1, \dots, a_k) \in R(\mathcal{A})$, the tuple $(\sigma(a_1), \dots, \sigma(a_k))$ is in $R(\mathcal{B})$. If σ is bijective and its inverse is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$, we say $\mathcal{A}$ and $\mathcal{B}$ are *isomorphic* and denote this as $\mathcal{A} \cong \mathcal{B}$.

We can view uncoloured graphs G as relational structures whose universe is the vertex set $V(G)$ and whose vocabulary consists of the single binary relation $E(G)$. For coloured graphs (G, λ), the vocabulary additionally contains a relation symbol R_c for every colour c in the range of λ. Usually, G is edge-coloured and thus, all relations R_c are binary. More precisely, R_c is interpreted by the set of all pairs (u, v) which satisfy $\lambda(u, v) = c$.

We now give an introduction into the logic C, the extension of first-order logic (FO) by *counting quantifiers* and its finite-variable fragments. We mainly follow [64].

We briefly recapitulate FO. Atomic formulae in FO are equalities $x = y$, where x and y are individual variables (which will be ranging over the universe of the considered structures), and $R\bar{x}$, where $R \in \tau$ and $\bar{x}$ is a tuple of variables whose length equals the arity of R. All other FO-formulae are constructed recursively from the atomic formulae using negation $\neg\varphi$, disjunction $(\varphi \vee \psi)$, and existential quantifiers $\exists x \varphi$, where x is a variable, and φ, ψ are FO-formulae. As abbreviations, we also use conjunctions $(\varphi \wedge \psi)$ for $\neg(\neg\varphi \vee \neg\psi)$, implications $(\varphi \to \psi)$ for $(\neg\varphi \vee \psi)$, and universal quantifiers $\forall x \varphi$ for $\neg\exists x \neg\varphi$. We also use $\textsc{true}$ for $\forall x (x = x)$, $\textsc{false}$ for $\neg\textsc{true}$, $\varphi \leftrightarrow \psi$ for $(\varphi \to \psi) \wedge (\psi \to \varphi)$, and $x \neq y$ for $\neg x = y$.

The logic C extends FO by counting quantifiers. C-formulae are constructed from the same atomic formulae as for FO. However, since we now allow counting, the logical connectives used to build the formulae are negation $\neg\varphi$, disjunction $(\varphi \vee \psi)$, and counting quantifiers $\exists^{\geq p} x \varphi$, where $p \in \mathbb{N}$, x is a variable, and φ, ψ are formulae. Again, we allow analogous abbreviations as for FO-formulae, letting $\exists x \varphi := \exists^{\geq 1} x \varphi$

and $\forall x\varphi := \neg\exists^{\geq 1}x\neg\varphi$. We also use variants of the counting quantifiers such as $\exists^{<p}x\varphi$ and $\exists^{=p}x\varphi$.

The semantics of C is defined in the usual way by inductively defining a satisfaction relation $\models$ between pairs $(\mathcal{A}, \nu)$ and formulae φ, where $\mathcal{A}$ is a finite relational structure (typically, a directed or undirected graph) and ν is an assignment of elements in $V(\mathcal{A})$ to the variables. The only step going beyond standard FO is that of counting quantifiers: $(\mathcal{A}, \nu) \models \exists^{\geq p}x\varphi$ if and only if there are distinct vertices $v_1, \dots, v_p \in V(\mathcal{A})$ such that $\big(\mathcal{A}, \nu(v_i/x)\big) \models \varphi$ for each v_i, where $\nu(v_i/x)$ is the assignment identical to ν except that $\nu(v_i/x)(x) = v_i$.

An occurrence of a variable x is *free* in a formula φ if it is outside of all subformulae $\exists^{\geq p}x\psi$, otherwise the occurrence is *bound*. We often write $\varphi(x_1, \dots, x_\ell)$ to indicate that the free variables of φ are among $x_1, \dots, x_\ell$. (Not all of these variables are required to be free or even appear in φ.) Then we also denote by $\varphi(y_1, \dots, y_\ell)$ the result of substituting variables $y_1, \dots, y_\ell$ for the free occurrences of $x_1, \dots, x_\ell$. For a formula $\varphi(x_1, \dots, x_\ell) \in \mathsf{C}$ and vertices $u_1, \dots, u_\ell \in V(\mathcal{A})$, let ν be an assignment satisfying $\nu(x_i) = u_i$ for all $i \in [\ell]$. Then for $(\mathcal{A}, \nu) \models \varphi$, we can write $\mathcal{A} \models \varphi(u_1, \dots, u_\ell)$ since the validity of $(\mathcal{A}, \nu) \models \varphi$ only depends on the assignments for the x_i. Moreover, we write $\varphi[\mathcal{A}, u_1, \dots, u_i, x_{i+1}, \dots, x_\ell]$ to denote the set of all $(\ell - i)$-tuples $(u_{i+1}, \dots, u_\ell)$ such that $\mathcal{A} \models \varphi(u_1, \dots, u_\ell)$.

Observe that C is only a syntactical extension of FO and does not add to the expressive power of FO, because $\exists^{\geq p}x\varphi(x)$ is equivalent to

$$\exists x_1 \dots \exists x_p \Big(\bigwedge_{i \neq j} x_i \neq x_j \wedge \bigwedge_i \varphi(x_i) \Big).$$

However, the situation is entirely different when limiting the number of variables in formulae. The k-variable fragment C^k of C consists of the formulae of C which use at most k variables, and it is indeed more expressive than the k-variable fragment of FO: the formula $\exists^{\geq k+1}x(x = x)$ cannot be expressed in the k-variable fragment of FO.

The *quantifier depth* $\mathrm{qd}(\varphi)$ of a formula $\varphi \in \mathsf{C}$ is its depth of quantifier nesting (see the definition of *quantifier rank* in [119]). More formally,

- if φ is atomic, then $\mathrm{qd}(\varphi) = 0$.
- $\mathrm{qd}(\neg\varphi) = \mathrm{qd}(\varphi)$.
- $\mathrm{qd}(\varphi_1 \vee \varphi_2) = \max\{\mathrm{qd}(\varphi_1), \mathrm{qd}(\varphi_2)\}$.
- $\mathrm{qd}(\exists^{\geq p}x\varphi) = \mathrm{qd}(\varphi) + 1$.

We say a formula $\varphi \in \mathsf{C}$ has *width* k if every subformula of φ has at most k free variables. We denote the set of all C-formulae of width k by $\mathsf{C}^k_{\mathsf{w}}$.

Example 2.3. *The following formula in* C^7 *has width* 3*:*

$$\exists x_1(E(x,x_1) \wedge \exists x_2(E(x_1,x_2) \wedge \exists x_3(E(x_2,x_3) \wedge \exists x_4(E(x_3,x_4) \wedge \exists x_5 E(x_5,y))))).$$

It is equivalent to the C^3*-formula*

$$\exists z(E(x,z) \wedge \exists x(E(z,x) \wedge \exists z(E(x,z) \wedge \exists x(E(z,x) \wedge \exists z E(z,y)))))$$

and expresses that there is a walk of length 6 from the first to the second input vertex. That is, if G *is a graph and* $\varphi(x,y)$ *is any of the two presented formulae, then* $G \models \varphi(v,w)$ *if and only if* v *and* w *are connected via a walk of length 6.*

We give another example, which will be useful in the course of this thesis.

Example 2.4. *For every* $k \in \mathbb{N}_0$*, we define a* $\mathsf{C}^3_{\mathsf{w}}$*-formula* $\mathsf{dist}_{\leq k}$ *such that for every graph* G *and all vertices* $u, u' \in V(G)$*, it holds that* $G \models \mathsf{dist}_{\leq k}(u,u')$ *if and only if* u *and* u' *have distance at most* k *in* G*. We let*

$$\mathsf{dist}_{\leq k}(x,x') := \begin{cases} x = x' & \textit{if } k = 0 \\ \exists y_k\big((E(x,y_k) \vee x = y_k) \wedge \mathsf{dist}_{\leq k-1}(y_k,x')\big) & \textit{otherwise.} \end{cases}$$

Note that for $k \in \mathbb{N}_0$*, the* $\mathsf{C}^3_{\mathsf{w}}$*-formula* $\mathsf{dist}_{=k}(x,x') := \mathsf{dist}_{\leq k}(x,x') \wedge \neg\mathsf{dist}_{\leq k-1}(x,x')$ *states that* x *and* x' *have distance exactly* k*. Moreover, in every graph of order at most* n*, the* $\mathsf{C}^3_{\mathsf{w}}$*-sentence* $\mathsf{conn}_n := \forall x \forall x' \mathsf{dist}_{\leq n-1}(x,x')$ *states that the graph is connected.*

We will use the following characterisation of C^k.

Lemma 2.5. *Every* C*-formula of width* k *is equivalent to a* C^k*-formula.*

To translate a C-formula of width k into a C^k-formula, it suffices to rename bound variable occurrences. We omit the straightforward proof. Also note that every C^k-formula has width k.

For a relational structure $\mathcal{A}$ and a logic L, the L-*type* of a tuple $(u_1,\dots,u_\ell) \in V(\mathcal{A})$ is the set of L-formulae $\varphi(x_1,\dots,x_\ell)$ for which $\mathcal{A} \models \varphi(u_1,\dots,u_\ell)$. If $\mathsf{L} \in \{\mathsf{FO}, \mathsf{C}, \mathsf{C}^k\}$, the *atomic* L*-type* of $(u_1,\dots,u_\ell)$ is the set of atomic L-formulae satisfied by $(u_1,\dots,u_\ell)$ in $\mathcal{A}$. We call the atomic FO-type simply the *atomic type.*

For structures $\mathcal{A}$ and $\mathcal{B}$, we say L *distinguishes* $\mathcal{A}$ *and* $\mathcal{B}$ if there is a sentence $\varphi \in \mathsf{L}$ such that $\mathcal{A} \models \varphi$ and $\mathcal{B} \not\models \varphi$. Similarly, L *identifies* $\mathcal{A}$ if for every structure $\mathcal{B} \not\cong \mathcal{A}$, the logic L distinguishes $\mathcal{A}$ and $\mathcal{B}$. If L does not distinguish $\mathcal{A}$ and $\mathcal{B}$, we say $\mathcal{A}$ and $\mathcal{B}$ are L-*equivalent.*

2.5 Partially Oriented Graphs

So far, we have explained how to understand graphs as relational structures. We intend to apply the Weisfeiler-Leman algorithm to graphs, or, a bit more generally, to finite relational structures with just relations of arity at most 2. There is a simple way to model a finite relational structure $\mathcal{A}$ as a so-called partially oriented graph, maintaining all the information about $\mathfrak{A}$ expressible in C by using only two variables (i.e. the C^2-type). We discuss it in this section. As we will see in Chapter3, our model captures all the properties that Colour Refinement is able to detect.

An *edge-coloured partially oriented graph* (in short, an ec-POG) is an edge-coloured directed graph (G, λ) without loops such that for all $(v, w), (v', w') \in E(G)$, it holds that if $(w, v) \in E(G)$ then $\lambda(v, w) = \lambda(w, v)$, and if $\lambda(v, w) = \lambda(v', w')$, then $(w, v) \in E(G)$ if and only if $(w', v') \in E(G)$.

We say that an edge $(v, w) \in E(G)$ is *undirected* if $(w, v) \in E(G)$. We accordingly draw (v, w) and (w, v) as one undirected edge between v and w (see Figure 2.1) and denote it by $\{v, w\}$. Otherwise, i.e. if $(v, w) \in E(G)$, but $(w, v) \notin E(G)$, we call (v, w) *directed*. An ec-POG (G, λ) is *complete* if for all $v, w \in V(G)$ with $v \neq w$, we have $(v, w) \in E(G)$ or $(w, v) \in E(G)$. In particular, if (G, λ) is a complete ec-POG, it might still hold that G is not be a complete graph (see Figure 2.1).

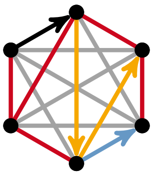


Figure 2.1: A complete ec-POG.

In the following, we consider finite relational structures over a fixed signature $\tau = (R_1, \dots, R_\ell)$ where R_i has arity r_i. Let $\mathfrak{A}$ be a finite relational structure over τ with universe V. For every $i \in [\ell]$, we define a function $\lambda_i \colon V^2 \to \mathcal{P}(\{1,2\}^{r_i})$ via

$$\lambda_i(v_1, v_2) := \big\{(j_1, \dots, j_{r_i}) \in \{1,2\}^{r_i} \;\big|\; \mathfrak{A} \models R_i(v_{j_1}, \dots, v_{j_{r_i}})\big\}$$

where $\mathcal{P}$ denotes the power set and $\{1,2\}^{r_i}$ denotes the set of all r_i-tuples over $\{1,2\}$. For all $v, w \in V$ with $v \neq w$, we let

$$\lambda(v, w) := \big(\lambda_1(v, w), \dots, \lambda_\ell(v, w)\big).$$

Since for each i, the possible images of λ_i come from a set of bounded size, by exploiting the order of the relations R_i in τ, one can easily define a canonical linear ordering $\leq$ on the image of λ (for example, by using the lexicographic order). With the help of this ordering, we define ec-POG$(\mathfrak{A}) := ((V, E_{\mathfrak{A}}), c_{\mathfrak{A}})$ as the complete ec-POG with vertex set V, edge set

$$E_{\mathfrak{A}} := \{(v, w) \mid v, w \in V, v \neq w \text{ and } \lambda(v, w) \leq \lambda(w, v)\}$$

and the edge colouring $\lambda_{\mathfrak{A}} \coloneqq c|_{E_{\mathfrak{A}}}$, the restriction of c to the domain $E_{\mathfrak{A}}$.

Note that $\lambda(v,w)$ uniquely determines the atomic C^2-types of v, w, (v,w) and (w,v). Intuitively, this means that if the universe V has at least two elements, then the colouring c stores all the information about $\mathfrak{A}$ that the logic C^2 can express. Hence, there is no need to include vertex colours or loops in our definition of ec-POGs. (More precisely, the atomic C^2-type of v is encoded in the colour $\lambda(v,w)$ for all $w \neq v$, so does not need to be modelled by loops or vertex colours.) Note also that in ec-POG($\mathfrak{A}$), a directed edge cannot have the same colour as an undirected edge.

A partition Π of the vertex set of an ec-POG is *equitable* if for all $P, Q \in \Pi$, for all $v, v' \in P$ and for every edge colour c, the vertices v and v' have the same numbers of c-coloured outgoing, incoming and undirected edges connecting them to Q. An ec-POG is *colour-regular* if for each edge colour c, every vertex v has the same c-indegree, the same c-outdegree, and the same c-degree for undirected edges, respectively. An edge-coloured undirected biregular graph on bipartition (P,Q) is called *colour-biregular* if for every edge colour c, the subgraph $\big(V(G), \{\{u,v\} \in E(G) \,\big|\, \{u,v\} \text{ has colour } c\}\big)$ is biregular on (P,Q). If the graph is partially oriented, then additionally, for every edge colour c, the vertices in P must have the same number of c-coloured outgoing and also equal numbers of c-coloured incoming edges, and analogous conditions must hold for Q.

3 The Weisfeiler-Leman Algorithm

The Weisfeiler-Leman algorithm is a powerful tool in both theoretical and practical approaches to tackling the graph isomorphism problem. Based on the fact that the multiset of vertex degrees in a graph is a graph property that is preserved under isomorphism, the algorithm repeatedly refines a colouring of vertex tuples of the input graph in an isomorphism-invariant fashion and eventually obtains a colouring that is *stable*. A comparison of the stable colourings obtained by executing the algorithm on each of two given graphs can often be used to prove their non-isomorphism.

For every $k \in \mathbb{N}$, there is a k-dimensional variant of the algorithm, which we call the *k-dimensional Weisfeiler-Leman algorithm*. In the next section, we introduce the 1-dimensional Weisfeiler-Leman algorithm. We discuss its generalisation to higher dimensions in the subsequent section.

3.1 Colour Refinement

The 1-dimensional Weisfeiler-Leman algorithm, which is also called *Colour Refinement* or *Naïve Vertex Classification*, repeatedly reassigns to a vertex the multiset of colours of its neighbours from the previous iteration. It terminates when the reassignment does not refine the partition of the vertex set anymore.

Definition 3.1 (Colour Refinement)**.** Let $\lambda\colon V(G) \to \mathcal{C}$ be a colouring of the vertices of a graph G. The colouring computed by Colour Refinement on input G is defined recursively: we set $\chi^0_{G,1} \coloneqq \lambda$, i.e. the initial colouring is λ. For $i \in \mathbb{N}$, the colouring $\chi^i_{G,1}$ computed by Colour Refinement after i iterations on G is defined as $\chi^i_{G,1}(v) \coloneqq \big(\chi^{i-1}_{G,1}(v), \{\!\!\{\chi^{i-1}_{G,1}(w) \mid w \in N(v)\}\!\!\}\big)$.

That is, $\chi^i_{G,1}$ consists of the colour of v from the previous iteration as well as the multiset of colours of neighbours of v from the previous iteration. Recall that, for a colouring $\lambda\colon V(G) \to \mathcal{C}$, we denote by $\pi(\lambda)$ the vertex partition induced by λ. It is not difficult to see that the reassignment refines the previous colouring, i.e. $\pi(\chi^{i-1}_{G,1}) \succeq \pi(\chi^i_{G,1})$ holds for every graph G and every $i \in \mathbb{N}$. Therefore, there is a unique minimal integer j such that $\pi(\chi^j_G) \equiv \pi(\chi^{j+1}_{G,1})$. For this value j, we define the *output* of Colour Refinement on input G to be $\chi_G \coloneqq \chi^j_{G,1}$ and call $\chi_{G,1}$ and $\pi(\chi_{G,1})$ the *stable colouring* and the *stable partition*, respectively, of G.

Accordingly, executing *i Colour Refinement iterations* on G means computing the colouring $\chi^i_{G,1}$. We call a graph G with colouring λ and the induced partition $\pi(\lambda)$ *stable* if $\pi(\lambda) \equiv \pi(\chi_{G,1})$.

Algorithm 3.1 shows a pseudo-code of Colour Refinement, whereas Figure 3.1 displays an application of the algorithm to a path. Note that in Algorithm 3.1, colours are replaced with values in $\mathbb{N}$ to keep the encoding handleable.

Recall that every colouring function induces a partition in the corresponding graph: a vertex colouring induces a partition of the vertex set, whereas an edge colouring induces a partition of the vertices and edges. Also recall that a partition Π of the vertex set of an uncoloured graph is *equitable* if for all $P, Q \in \Pi$ and for all $v, v' \in P$, the vertices v and v' have the same number of neighbours in Q. It is not hard to see that every graph has a unique coarsest equitable partition, namely the partition obtained by starting with the partition induced by the initial vertex colours and iteratively subdividing partition classes that contain vertices v, v' for which there exists a partition class C in the current partition such that $|N(v) \cap C| \neq |N(v') \cap C|$. This is exactly the partition which Colour Refinement computes on its input graph. In particular, for all $P, Q \in \pi(\chi_{G,1})$ with $P \neq Q$, the graph $G[P]$ is regular and the graph $G[P, Q]$ is biregular. The algorithm can be easily adapted to take also edge colours into account. In that case, the final partition $\pi(\chi_{G,1})$ is equitable on the subgraph induced by every edge colour.

Colour Refinement can be implemented to run in time $O((m + n) \log n)$, where n is the order of the input graph and m is the number of its edges (see, for example, [32, 126, 139]).

Algorithm 3.1 A pseudo-code for Colour Refinement.

Input: A vertex-coloured graph (G, λ).
Output: A colouring $\chi\colon V(G) \to \mathbb{N}$, which induces the coarsest equitable partition on $V(G)$.

```
$\ell := 0$
$\chi(v) := \lambda(v)$ for every $v \in V(G)$
$L := sort(\{\chi(v) \mid v \in V(G)\})$                      // Sort lexicographically
$\chi(v) := L.index(\chi(v))$ for every $v \in V(G)$          // New colours are in $\mathbb{N}$
while $\ell \neq \max(\chi(V(G)))$ do
    $\ell := \max(\chi(V(G)))$
    $\chi^{\mathrm{r}}(v) := (\chi(v), \{\{\chi(w) \mid w \in N(v)\}\})$ for every $v \in V(G)$
    $L := sort(\{\chi^{\mathrm{r}}(v) \mid v \in V(G)\})$             // Sort lexicographically
    $\chi(v) := L.index(\chi^{\mathrm{r}}(v))$ for every $v \in V(G)$   // New colours are in $\mathbb{N}$
end while
return $\chi$
```

For two graphs G and G', the 1-dimensional Weisfeiler-Leman algorithm *distinguishes* G and G' if attaching to each of the vertex sets a single vertex and

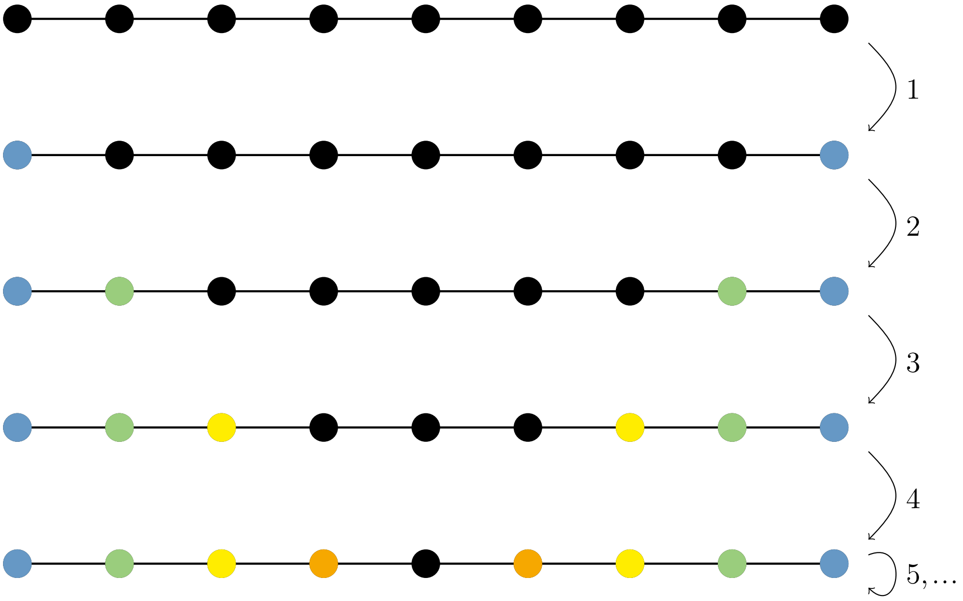


Figure 3.1: The iterations of Colour Refinement when applied to an uncoloured path of length 8.

applying the algorithm to the disjoint union of the resulting graphs yields a colouring with differing colour class sizes in the graphs. More precisely, define $\widehat{G}$ as the graph with vertex set $V(G) \dot\cup \{v\}$, where v is a fresh vertex, and edge set $E(G) \cup \{\{v, w\} \mid w \in V(G)\}$. Define $\widehat{G}'$ analogously with a second fresh vertex v'. Let χ be the output of the 1-dimensional Weisfeiler-Leman algorithm applied to the disjoint union $\widehat{G} \dot\cup \widehat{G}'$. Then the algorithm *distinguishes* G and G' if there is a colour c such that the sets $\{v \mid v \in V(G), \chi(v) = c\}$ and $\{w \mid w \in V(G'), \chi(w) = c\}$ have different cardinalities. Colour Refinement *identifies* G if it distinguishes G from every other, non-isomorphic graph G'.

The following is a direct consequence of the definition of the algorithm.

Observation 3.2. *For every $\ell \in \mathbb{N}$, the 1-dimensional Weisfeiler-Leman algorithm fails to distinguish every pair of ℓ-regular graphs of equal order.*

Even though the 1-dimensional algorithm fails to distinguish basic graphs such as the disjoint union of two triangles and the graph consisting of a single hexagon, it is very powerful. For example, in an inductive fashion ascending from the leaves to the root, one can show that it identifies all vertex-coloured forests. Much more strongly, Colour Refinement identifies random graphs asymptotically almost surely, which is implied by the following theorem due to Babai, Erdős, and Selkow.

Theorem 3.3 ([14]). *On almost all graphs G, the colouring computed by Colour*

Refinement after two iterations induces a discrete partition on $V(G)$.

That is, the fraction of graphs of order n which are not identified by Colour Refinement tends to 0 as n tends to infinity. In fact, it is known that the fraction of graphs which are not identified decreases exponentially in the order n [16].

3.2 Higher Dimensions of the WL Algorithm

We now discuss the generalisation of the 1-dimensional Weisfeiler-Leman algorithm to higher dimensions. As the definition will imply, higher dimensions of the algorithm are more powerful, in the sense that the pairs of graphs that are distinguished by the k-dimensional Weisfeiler-Leman algorithm form a superset of the ones that are distinguished by the $(k-1)$-dimensional algorithm.

Let $k \in \mathbb{N}$ be fixed. Recall that the atomic type of a k-tuple $\bar{u} = (u_1, \dots, u_k)$ of vertices of a coloured graph (G, λ) is the set of all atomic facts satisfied by these vertices, i.e. all facts about edge connections and colours within the tuple. That is, tuples $\bar{u} = (u_1, \dots, u_k)$ and $\bar{v} = (v_1, \dots, v_k)$ of vertices of graphs G and H, respectively, have the same atomic type if and only if the mapping $u_i \mapsto v_i$ is an isomorphism from the induced coloured subgraph $G[\{u_1, \dots, u_k\}]$ to the induced coloured subgraph $H[\{v_1, \dots, v_k\}]$ (with the corresponding restrictions of λ and λ'). We denote the atomic type of $\bar{u} \coloneqq (u_1, \dots, u_k)$ in (G, λ) by $\operatorname{atp}(G, \lambda, \bar{u})$. It can be encoded as a $(k \times k)$-matrix M with

$$M_{ij} = \begin{cases} \bigl(0, \lambda\bigl(u_i, u_i\bigr)\bigr), & \text{if } u_i = u_j \\ \bigl(1, \lambda\bigl(u_i, u_j\bigr)\bigr), & \text{if } (u_i, u_j) \in E(G), u_i \neq u_j \\ (2, \bot), & \text{if } (u_i, u_j) \notin E(G), u_i \neq u_j. \end{cases}$$

We describe the colourings $\chi^i_{G,\lambda,k}$ computed by the k-dimensional Weisfeiler-Leman algorithm on input (G, λ). Since the considered colouring will always be unambiguous from the context, we omit λ in the following notation. The initial colouring $\chi^0_{G,k}$ assigns to each tuple its atomic type:

$$\chi^0_{G,k}(\bar{u}) \coloneqq \operatorname{atp}(G, \lambda, \bar{u}).$$

In the $(i+1)$-st *iteration*, the colouring $\chi^{i+1}_{G,k}$ is defined by

$$\chi^{i+1}_{G,k}(\bar{u}) \coloneqq \bigl(\chi^i_{G,k}(\bar{u}), \mathcal{M}_i(\bar{u})\bigr),$$

where, for $\bar{u} = (u_1, \dots, u_k)$, $\mathcal{M}_i(\bar{u})$ is the multiset

$$\{\!\!\{ \bigl(\operatorname{atp}(G, \chi^i_{G,k}, (u_1, \dots, u_k, v)), \chi^i_{G,k}(u_1, \dots, u_{k-1}, v), \\ \chi^i_{G,k}(u_1, \dots, v, u_k), \dots, \chi^i_{G,k}(v, u_2, \dots, u_k)\bigr) \bigm| v \in V(G) \}\!\!\}$$

if $k \geq 2$, and $\mathcal{M}_i(\bar{u}) := \{\!\{\chi^i_{G,k}(w) \,|\, w \in N(\bar{u})\}\!\}$ if $k = 1$. That is, if $k = 1$, the iteration is only over neighbours of $\bar{u}$ and it coincides with our definition from Section 3.1.

For $i \in \mathbb{N}_0$, let $\pi^i_{G,k}$ be the partition induced by $\chi^i_{G,k}$ on $(V(G))^k$. By definition, it holds that $\pi^i_{G,k} \succeq \pi^{i+1}_{G,k}$. Thus, there is some minimal integer j such that $\pi^j_{G,k} \equiv \pi^{j+1}_{G,k}$, i.e. such that $\pi^{j+1}_{G,k}$ is not strictly finer than $\pi^j_{G,k}$. For this value j, we call $\chi^j_{G,k}$ the *stable* (k-tuple) colouring of G and denote it by $\chi_{G,k}$. Accordingly, we call the partition $\pi^j_{G,k}$ the *stable* (k-tuple) partition of G and denote it by $\pi_{G,k}$.

For $k \in \mathbb{N}$, the *k-dimensional Weisfeiler-Leman algorithm* takes as input a vertex- or edge-coloured graph (G, λ) and returns as output the colouring $\chi_{G,k}$. (More precisely, similarly as Algorithm 3.1, an implementation of the k-dimensional algorithm will only return a colouring equivalent to $\chi_{G,k}$, but this is not an issue for our purposes, since we are mainly concerned with the induced partitions and not with precise colours.) The k-dimensional Weisfeiler-Leman algorithm can be implemented to run in time $O(n^{k+1} \log n)$, where n is the size of the input graph [90].

If the graph G is clear from the context or irrelevant, we omit it in the subscript. For instance, we write χ^i_k instead of $\chi^i_{G,k}$. We often proceed similarly with the dimension of the algorithm. However, since the objects we omit are always known and fixed (or irrelevant) in the specific context, this should cause no confusion.

Similarly as for the 1-dimensional algorithm, to use the k-dimensional Weisfeiler-Leman algorithm as an isomorphism test on two graphs G and G' (possibly with vertex/edge colourings), we apply it to the disjoint union of the graphs $\widehat{G} := \big(V(G) \,\dot\cup\, \{v\}, E(G) \cup \{\{v,w\} \mid w \in V(G)\}\big)$ and $\widehat{G}' := \big(V(G') \,\dot\cup\, \{v'\}, E(G') \cup \{\{v',w\} \mid w \in V(G')\}\big)$ for some fresh vertices v, v' with $v \neq v'$. Moreover, in $\widehat{G}$ and $\widehat{G}'$, we assign a common initial fresh colour to v and v'.

Abusing notation, we also write $\chi_{G,k}$ for the restriction of $\chi_{\widehat{G} \dot\cup \widehat{G}',k}$ to tuples from $(V(G))^k$ and proceed analogously for G'. (This should cause no confusion, since it will always be clear from the context if we are comparing two graphs or if we are just considering the run of the algorithm on a single input graph.) The k-dimensional Weisfeiler-Leman algorithm *distinguishes* G and G' if there is a colour c such that the sets $\{\bar{v} \,|\, \bar{v} \in \big(V(G)\big)^k, \chi_{G,k}(\bar{v}) = c\}$ and $\{\bar{w} \,|\, \bar{w} \in \big(V(G')\big)^k, \chi_{H,k}(\bar{w}) = c\}$ have different cardinalities. If the k-dimensional algorithm does not distinguish G and G', we denote this by $G \simeq_k G'$ and call G and G' *equivalent* with respect to the k-dimensional Weisfeiler-Leman algorithm. The algorithm *identifies* G if it distinguishes G from every second, non-isomorphic graph G'.

In order to talk about the final colours of tuples whose length is smaller than the dimension of the Weisfeiler-Leman algorithm that we are considering, we set

$$\chi_{G,k}(u_1,\dots,u_\ell) \coloneqq \chi_{G,k}(u_1,\dots,u_\ell,\underbrace{u_\ell,\dots,u_\ell}_{k-\ell \text{ times}}),$$

that is, $\chi_{G,k}(u_1,\dots,u_\ell)$ is the colour of the k-tuple resulting from extending the ℓ-tuple by repeating $k-\ell$ times its last entry.

Lemma 3.4. *For all $k \in \mathbb{N}$ and all coloured graphs (G,λ_G), (H,λ_H), it holds that $\{\!\{\chi_{G,k}(\bar{v}) \mid \bar{v} \in (V(G))^k\}\!\} = \{\!\{\chi_{H,k}(\bar{v}) \mid \bar{v} \in (V(H))^k\}\!\}$ or $\{\!\{\chi_{G,k}(\bar{v}) \mid \bar{v} \in (V(G))^k\}\!\} \cap \{\!\{\chi_{H,k}(\bar{v}) \mid \bar{v} \in (V(H))^k\}\!\} = \emptyset$, i.e. both multisets are equal or no colour occurs in both of the multisets. In the first case, it also holds that $\{\!\{\chi_{G,k}(v) \mid v \in V(G)\}\!\} = \{\!\{\chi_{H,k}(v) \mid v \in V(H)\}\!\}$.*

Proof: Let v^*_G be the unique vertex in $V(\widehat{G}) \setminus V(G)$ and define v^*_H analogously for H and recall that the two vertices have an initial colour that distinguishes them from all vertices in $V(G) \cup V(H) \subset V(\widehat{G}) \cup V(\widehat{H})$.

For $k = 1$, suppose there is an iteration i with $\{\!\{\chi^i_{G,1}(v) \mid v \in V(G)\}\!\} \neq \{\!\{\chi^i_{H,1}(v) \mid v \in V(H)\}\!\}$. However, then $\chi^{i+1}_1(v^*_G) \neq \chi^{i+1}_1(v^*_H)$ and thus, from the definition of the algorithm, we know that

$$\{\!\{\chi^{i+2}_{G,1}(v) \mid v \in V(G)\}\!\} \cap \{\!\{\chi^{i+2}_{H,1}(v) \mid v \in V(H)\}\!\} = \emptyset.$$

Hence, the statement from the lemma follows.

Now suppose that $k \geq 2$. For every colour C, define $C_G \coloneqq \{\bar{v} \in (V(G))^k \mid \chi_{G,k}(\bar{v}) = C\}$ and define C_H analogously for H. Now it suffices to show that for every $\bar{v} \in (V(G))^k$ and every $\bar{w} \in (V(G))^k$, the equality $\chi_{G,k}(\bar{v}) = \chi_{H,k}(\bar{w})$ implies for every colour C that $|C_G| = |C_H|$. We first finish the proof under this assumption. If there is a colour that has a positive multiplicity both in $\{\!\{\chi_{G,k}(\bar{v}) \mid \bar{v} \in (V(G))^k\}\!\}$ and in $\{\!\{\chi_{H,k}(\bar{v}) \mid \bar{v} \in (V(H))^k\}\!\}$, this implies that for all colours C, the sets C_G and C_H have equal sizes. Thus, $\{\!\{\chi_{G,k}(\bar{v}) \mid \bar{v} \in (V(G))^k\}\!\} = \{\!\{\chi_{H,k}(\bar{w}) \mid \bar{w} \in (V(H))^k\}\!\}$.

Furthermore, the equality of the two multisets implies $\{\!\{\chi_{G,k}(v) \mid v \in V(G)\}\!\} = \{\!\{\chi_{H,k}(v) \mid \bar{v} \in V(H)\}\!\}$, since $\chi_k(\bar{v})$ takes the atomic type of $\bar{v}$ into account and therefore, the colours of the tuples of the form $(v,\dots,v)$ are distinct from the colours of other tuples.

To show that $\chi_{G,k}(\bar{v}) = \chi_{H,k}(\bar{w})$ implies for every colour C that $|C_G| = |C_H|$, we proceed via induction. In the following, every letter with a bar represents a k-tuple of vertices. For k-tuples $\bar{u}$ and $\bar{x}$, define $S(\bar{u},\bar{x})$ to be the set of positions in $[k]$ at which $\bar{u}$ and $\bar{x}$ differ. We show by induction over $\ell \in \mathbb{N}_0$ that for all tuples $\bar{v} \in (V(G))^k$ and $\bar{w} \in (V(H))^k$, if $\chi_{G,k}(\bar{v}) = \chi_{H,k}(\bar{w})$, then for all colours C and for all sets $P \subseteq [k]$ with $|P| \leq \ell$, it holds that

$$\big|\{\bar{u} \mid \chi_{G,k}(\bar{u}) = C, S(\bar{v},\bar{u}) = P\}\big| = \big|\{\bar{x} \mid \chi_{H,k}(\bar{x}) = C, S(\bar{w},\bar{x}) = P\}\big|. \tag{3.1}$$

Note that the set on the left-hand side of the equation is a subset of $(V(G))^k$ and the set on the right-hand side is a subset of $(V(H))^k$, since $\chi_{G,k}$ and $\chi_{H,k}$ are only defined on tuples of vertices from $V(G)$ and $V(H)$, respectively.

The induction base case $\ell = 0$ follows from the assumption that $\chi_{G,k}(\bar{v}) = \chi_{H,k}(\bar{w})$. For $\ell = 1$, it suffices to see that, if (3.1) did not hold, then χ_k would not be stable. Indeed, the multisets $\mathcal{M}_i(\bar{v})$ and $\mathcal{M}_i(\bar{w})$ of colours computed after one more iteration of the Weisfeiler-Leman algorithm would differ and thus, $\bar{v}$ and $\bar{w}$ would receive different colours by the algorithm. For the induction step from $\ell \geq 1$ to $\ell + 1$, suppose the statement holds for all $P \subseteq [k]$ with $|P| \leq \ell$. Pick arbitrary $\bar{v} \in (V(G))^k$ and $\bar{w} \in (V(H))^k$ with $\chi_{G,k}(\bar{v}) = \chi_{H,k}(\bar{w})$. Let C' be a colour and let $P' \subseteq [k]$ be a set with $|P'| = \ell + 1$. Choose some $m \in P'$. Then by the induction hypothesis, for all colours C, it holds that

$$\big|\{\bar{u} \,|\, \chi_{G,k}(\bar{u}) = C, S(\bar{v},\bar{u}) = \{m\}\}\big| = \big|\{\bar{x} \,|\, \chi_{H,k}(\bar{x}) = C, S(\bar{w},\bar{x}) = \{m\}\}\big|.$$

Call the set on the left-hand side of the equation $C_G(\bar{v},\{m\})$ and the set on the right-hand side $C_H(\bar{w},\{m\})$. Again by the induction hypothesis, for all $\bar{u} \in C_G(\bar{v},\{m\})$ and all $\bar{x} \in C_H(\bar{w},\{m\})$, it holds that

$$\big|\{\bar{y} \,|\, \chi_{G,k}(\bar{y}) = C', S(\bar{u},\bar{y}) = P' \backslash \{m\}\}\big| = \big|\{\bar{y} \,|\, \chi_{H,k}(\bar{y}) = C', S(\bar{x},\bar{y}) = P' \backslash \{m\}\}\big|.$$

In particular, the size of the sets is independent of the precise choice of $\bar{u}$ and $\bar{x}$. Call the set on the left-hand side of the equation $C'_G(\bar{v}, P' \setminus \{m\})$ and the set on the right-hand side $C'_H(\bar{w}, P' \setminus \{m\})$. Since for every k-tuple $\bar{y} \in (V(G))^k$ with $S(\bar{v},\bar{y}) = P'$, there is a unique k-tuple $\bar{u}$ with $S(\bar{v},\bar{u}) = \{m\}$ and $S(\bar{u},\bar{y}) = P' \backslash \{m\}$ and the analogous holds in H, we obtain

$$\begin{aligned}\big|\{\bar{y} \,|\, \chi_{G,k}(\bar{u}) = C', S(\bar{v},\bar{y}) = P'\}\big| &= \sum_C \big|C_G(\bar{v},\{m\})\big| \cdot \big|C'_G(\bar{v}, P' \setminus \{m\})\big| \\ &= \sum_C \big|C_H(\bar{w},\{m\})\big| \cdot \big|C'_H(\bar{w}, P' \setminus \{m\})\big| \\ &= \big|\{\bar{y} \,|\, \chi_{H,k}(\bar{y}) = C', S(\bar{w},\bar{y}) = P'\}\big|,\end{aligned}$$

which concludes the proof of the induction step. Thus, the equality $\chi_{G,k}(\bar{v}) = \chi_{H,k}(\bar{w})$ implies for all colours C that

$$\big|\{\bar{u} \in (V(G))^k \,|\, \chi_{G,k}(\bar{u}) = C\}\big| = \big|\{\bar{x} \in (V(H))^k \,|\, \chi_{H,k}(\bar{x}) = C\}\big|$$

and hence $\{\!\{\chi_{G,k}(\bar{v}) \,|\, \bar{v} \in (V(G))^k\}\!\} = \{\!\{\chi_{H,k}(\bar{v}) \,|\, \bar{v} \in (V(H))^k\}\!\}$. □

Notation 3.5. *As noticed before, for every edge-coloured graph (G, λ) and $k \in \mathbb{N}$, there is a minimal number $i \in \mathbb{N}$ such that $\pi^i_{G,k} \equiv \pi^{i+1}_{G,k}$. We call this number the* iteration number *or* number of iterations *of the k-dimensional Weisfeiler-Leman algorithm on input (G, λ) and write* $\mathrm{WL}_k(G)$ *for it.*

For $k, n \in \mathbb{N}$, we write $\mathrm{WL}_k(n)$ for the maximum iteration number of the k-dimensional Weisfeiler-Leman algorithm on an input graph with n vertices.

Hence, $\mathrm{WL}_k(n) \leq r$ if and only if on every graph of size n, the k-dimensional Weisfeiler-Leman algorithm terminates after at most r iterations.

Observation 3.6. *For all $k, n \in \mathbb{N}$, it holds that*

$$\mathrm{WL}_k(n) \leq n^k - 1.$$

Proof: This follows from the fact that on input G, for all $i < \mathrm{WL}_k(G)$, the partition $\pi^{i+1}_{G,k}$ strictly refines the partition $\pi^{i}_{G,k}$ and the finest partition of a set of size n^k is the discrete one with n^k elements. □

The stable k-tuple colouring induces a partition of the set of k-tuples of the form $(v, \dots, v)$ and thus, by identifying the tuple with the vertex v, it also induces a partition on $V(G)$. The following observation can directly be inferred from the isomorphism invariance of the computed colouring.

Observation 3.7. *For every $k \in \mathbb{N}$ and every input graph G, the partition induced by $\chi_{G,k}$ on $V(G)$ is coarser than the partition into vertex orbits.*

Using the observation, we can show the following statement.

Fact 3.8. *On paths of length n, the 1-dimensional Weisfeiler-Leman algorithm terminates after at most $\lfloor \frac{n}{2} \rfloor$ iterations.*

Proof: The result follows from Observation 3.7 and the circumstance that after at most $\lfloor \frac{n}{2} \rfloor$ iterations, the 1-dimensional Weisfeiler-Leman algorithm induces the vertex orbit partition on a path of length n. □

For uncoloured paths, the bound from the fact is tight. Thus, since paths on n vertices have length $n-1$, we obtain the following corollary.

Corollary 3.9. $\mathrm{WL}_1(n) \geq \lfloor \frac{n-1}{2} \rfloor$.

As stated in Observation 3.2, the 1-dimensional Weisfeiler-Leman algorithm fails to distinguish, for all $n, \ell \in \mathbb{N}$, every pair of ℓ-regular uncoloured graphs on n vertices. In fact, given an uncoloured regular graph, the algorithm immediately terminates because the degree is the only parameter it uses as a refinement criterion in the first iteration.

Random regular graphs are asymptotically almost surely identified by the 2-dimensional Weisfeiler-Leman algorithm [25, 114]. Still, for analogous reasons as for the 1-dimensional algorithm failing to distinguish non-isomorphic regular graphs of the same size, the 2-dimensional Weisfeiler-Leman algorithm immediately terminates if its input is a strongly regular graph, i.e. a regular graph with two parameters μ_1, μ_2 such that every pair of adjacent vertices has μ_1 common neighbours and every pair of non-adjacent vertices has μ_2 common neighbours.

Observation 3.10. *The 2-dimensional Weisfeiler-Leman algorithm fails to distinguish every pair of non-isomorphic strongly regular graphs of the same order and the same degree with equal parameters μ_1, μ_2.*

For example, the 2-dimensional Weisfeiler-Leman algorithm does not distinguish the Shrikhande graph and the 4×4 rook's graph from each other because they both are strongly regular graphs on 16 vertices with the same parameters [80, 131, 150]. In particular, since those graphs are not isomorphic, not every strongly regular graph is identified by the 2-dimensional Weisfeiler-Leman algorithm.

In Section 2.3, we required for all considered colourings λ that $\lambda(v_1, u_1) = \lambda(v_2, u_2)$ if and only if $\lambda(u_1, v_1) = \lambda(u_2, v_2)$ for all vertices u_1, u_2, v_1, v_2, a property which we call *converse equivalence*. We briefly explain why this is not an actual restriction. Suppose that a coloured graph (G, λ) does not satisfy converse equivalence. We define a new colouring $\widehat{\lambda}_G$ which assigns to every arc (u, v) corresponding to an undirected edge $\{u, v\}$ the pair consisting of $\lambda(u, v)$ and $\lambda(v, u)$ (ensuring that the encoding of the new colour does not coincide with other encodings of the λ-colours). The new coloured graph $(G, \widehat{\lambda}_G)$ satisfies converse equivalence. Furthermore, for two graphs (G, λ_G) and (H, λ_H), it is easy to see that

$$(G, \lambda_G) \cong (H, \lambda_H) \iff (G, \widehat{\lambda}_G) \cong (H, \widehat{\lambda}_H).$$

Therefore, in the context of the isomorphism problem, we can always assume converse equivalence. In fact, by the definition of the k-dimensional Weisfeiler-Leman algorithm, the initial colouring of a k-tuple already takes the colours of all arcs between pairs of vertices in the tuple into account and thus ensures converse equivalence. It is then maintained in all future iterations as well.

3.3 The Weisfeiler-Leman Dimension

The Weisfeiler-Leman algorithm can be used to check whether two given graphs are non-isomorphic by computing stable colourings and rejecting if there is a colour c such that the numbers of c-coloured vertex tuples in the two graphs differ. However, even if the stable colourings of the two graphs agree in all colours, the graphs might not be isomorphic. This leads to the notion of the Weisfeiler-Leman dimension of a graph as a measure for its inherent structural complexity (see [64, Definition 18.4.3]).

Definition 3.11. Let G be a graph. The *Weisfeiler-Leman dimension* of G is the smallest k such that the k-dimensional Weisfeiler-Leman algorithm identifies G. Similarly, for a class of graphs, its *Weisfeiler-Leman dimension* is the smallest $k \in \mathbb{N}$ such that every graph in the class has Weisfeiler-Leman dimension at most k, and ∞ if no such k exists.

For example, as discussed in Section 3.2, the Weisfeiler-Leman dimension of the cycle C_6 is at least 2. In fact, it is easy to see that it is exactly 2.

Corollary 3.12. *For every graph G with $|G| \geq 2$, its Weisfeiler-Leman dimension is bounded by $|G| - 1$.*

Proof: For $|G| = 2$, this is obvious. For $|G| \geq 3$, this follows from the definition of the Weisfeiler-Leman algorithm, since for every $(k-1)$-tuple $\bar{v}$ of pairwise distinct vertices, the colour $\chi_{G,|G|-1}(\bar{v})$ encodes the isomorphism type of the entire graph G. □

Unfortunately, the Weisfeiler-Leman dimension of the class of all graphs is ∞ (see [30]). Thus, it is not possible to decide graph isomorphism in general with a fixed dimension of the Weisfeiler-Leman algorithm and no further tools.

Theorem 3.13 ([30]). *For every $n \in \mathbb{N}$, there is a pair of graphs G_n and H_n such that*

1. *G_n and H_n have $O(n)$ vertices.*
2. *G_n and H_n have degree 3 and colour class size 4.*
3. *$G_n \simeq_n H_n$.*
4. *$G_n \ncong H_n$.*

However, we can still hope for bounds on the Weisfeiler-Leman dimension of certain subclasses of graphs. In particular, if we wish to determine the dimension of a hereditary graph class, once we accept Weisfeiler-Leman dimension at least 2, it suffices to only consider the connected graphs in the class.

Lemma 3.14. *Let $\mathcal{G}$ be a hereditary graph class and let $k \in \mathbb{N}_{\geq 2}$. If the k-dimensional Weisfeiler-Leman algorithm identifies every connected graph in $\mathcal{G}$, then the Weisfeiler-Leman dimension of $\mathcal{G}$ is at most k.*

Proof sketch: This follows from the fact that for $k \geq 2$, for $i \in \mathbb{N}$, and for every pair of vertices u and v, the colour which the k-dimensional Weisfeiler-Leman algorithm assigns to (u, v) encodes the number of walks of length i that start in u and end in v. It thus also encodes information about whether u and v are contained in the same connected component or in different ones. Now it is not hard to see that under the assumption that connected components are identified, two graphs whose multisets of connected components are not equal (or, equivalently, two non-isomorphic graphs) are distinguished. □

Note that the lemma does not hold for $k = 1$.

Recall that for a coloured graph (G, λ), we write $\chi_{G,k}$ as a shorthand for $\chi_{G,\lambda,k}$, the colouring which the k-dimensional Weisfeiler-Leman algorithm computes on input (G, λ).

For the inductive proofs we present in Chapters 8–9, we need a stronger requirement on the Weisfeiler-Leman algorithm than to just distinguish graphs from each other.

Definition 3.15. Let $\mathcal{H}$ be a set of graphs and let $k \in \mathbb{N}$. We say that the k-dimensional Weisfeiler-Leman algorithm *determines (vertex) orbits on* $\mathcal{H}$ if for all coloured graphs $(G, \lambda), (G', \lambda')$ with $G, G' \in \mathcal{H}$ and all vertices $v \in V(G)$ and $v' \in V(G')$, the following holds: there exists an isomorphism from (G, λ) to (G', λ') mapping v to v' if and only if $\chi_{G,k}(v) = \chi_{G',k}(v')$.

Note that for $G' = G$, the vertex colour classes obtained by an application of the k-dimensional Weisfeiler-Leman algorithm to a coloured graph (G, λ) are the orbits of the automorphism group of G with respect to λ.

Observation 3.16. *If the k-dimensional Weisfeiler-Leman algorithm determines vertex orbits on a set of graphs $\mathcal{H}$, then it also distinguishes every pair of non-isomorphic graphs in $\mathcal{H}$ from each other: if $G \simeq_k G'$ for some graphs $G, G' \in \mathcal{H}$, using Lemma 3.4, there must be a vertex $v \in V(G)$ and a vertex $v' \in G'$ for which $\chi_{G,k}(v) = \chi_{G',k}(v')$. Thus, due to orbit determination, G and G' must be isomorphic (via an isomorphism mapping v to v').*

The following proposition is a useful correspondence between identification and determination of orbits.

Proposition 3.17. *Let $\mathcal{H}$ be a set of graphs and let $k \in \mathbb{N}$. Suppose that for every coloured graph (G, λ) with $G \in \mathcal{H}$, the k-dimensional Weisfeiler-Leman algorithm identifies all vertex-coloured versions of (G, λ). Then the $(k+1)$-dimensional Weisfeiler-Leman algorithm determines vertex orbits on $\mathcal{H}$.*

Proof: Let (G, λ) be a coloured graph with $G \in \mathcal{H}$. Suppose there are a graph H and vertices $v \in V(G)$, $v' \in V(H)$ such that $\chi_{G,k+1}(v) = \chi_{H,k+1}(v')$ holds. Then we can individualise v in G and v' in H (i.e. assign to them a colour which distinguishes them from all other vertices in G and H) and apply the k-dimensional Weisfeiler-Leman algorithm to these coloured graphs G_v and $H_{v'}$. Since $\chi_{G,k+1}(v) = \chi_{H,k+1}(v')$, we have $\{\!\!\{\chi_{G,k+1}(v, w_1, \ldots, w_k) \mid (w_1, \ldots, w_k) \in \big(V(G)\big)^k\}\!\!\} = \{\!\!\{\chi_{H,k+1}(v', w'_1, \ldots, w'_k) \mid (w'_1, \ldots, w'_k) \in \big(V(H)\big)^k\}\!\!\}$. Thus, the graphs G_v and $H_{v'}$ obtain equal multisets of colours with respect to the k-dimensional Weisfeiler-Leman algorithm. By assumption, this implies $G_v \cong H_{v'}$, which is equivalent to the existence of an isomorphism from G to H mapping v tov'. □

3.4 Connections to Logic and Games

In particular, when analysing the mechanisms of higher dimensions of the Weisfeiler-Leman algorithm on certain graphs or graph classes, it is cumbersome to always refer to the definition of the algorithm directly. However, the algorithm has correspondences to plenty of other research fields, two of which we present here. We refer the reader to existing literature (for example [30, 64]) for more information.

Let $k \in \mathbb{N}$. For graphs G and H with equal numbers of vertices and vertex colourings λ and λ', respectively, we define the *bijective k-pebble game* $\mathrm{BP}_k(G,H)$ as follows (see also [77]). The game is played by two players called *Spoiler* and *Duplicator*. It proceeds in rounds, each of which is associated with a pair of configurations $(\bar{v},\bar{w})$ with $\bar{v} \in \big(V(G)\big)^{\ell}$ and $\bar{w} \in \big(V(H)\big)^{\ell}$, where $\ell \in [0,k]$. The initial configuration is a pair of vertex tuples of equal length ℓ with $[0,k]$. If not specified otherwise, the initial configuration is the pair $\big((),()\big)$ of empty tuples.

We describe one round of the game. Suppose the current configuration is $(\bar{v},\bar{w}) = \big((v_1,\dots,v_\ell),(w_1,\dots,w_\ell)\big)$ with $\ell \in [0,k]$. Now Spoiler chooses an $i \in [k]$.

Then the following steps are performed.

(D) Duplicator picks a bijection $f\colon V(G) \to V(H)$.

(S) Spoiler chooses $v \in V(G)$ and sets $w \coloneqq f(v)$.

If $i \in [\ell]$, the new configuration of the game is the tuple

$$\big((v_1,\dots,v_{i-1},v,v_{i+1},\dots,v_\ell),(w_1,\dots,w_{i-1},w,w_{i+1},\dots,w_\ell)\big).$$

Otherwise, the new configuration is $\big((v_1,\dots,v_\ell,v),(w_1,\dots,w_\ell,w)\big)$.

If for the new configuration $\big((v_1,\dots,v_{\ell'}),(w_1,\dots,w_{\ell'})\big)$, the induced ordered subgraphs of G and H are not isomorphic, Spoiler wins the play (after the current round). More precisely, Spoiler wins if there is an $i \in [\ell]$ such that $\lambda(v_i) \neq \lambda'(w_i)$, or there are $i,j \in [\ell]$ such that $v_i = v_j \not\Leftrightarrow w_i = w_j$ or $v_iv_j \in E(G) \not\Leftrightarrow w_iw_j \in E(H)$. If there is no configuration of the play such that Spoiler wins, then Duplicator wins. ⌟

The interpretation of a configuration $\big((v_1,\dots,v_\ell),(w_1,\dots,w_\ell)\big)$ is that the first pebble pair is placed on v_1 and w_1, the second pebble pair is placed on v_2 and w_2, and so on.

We say that Spoiler (and Duplicator, respectively) *wins the game* $\mathrm{BP}_k(G,H)$ *with r moves* if Spoiler (and Duplicator, respectively) has a strategy to win the game after r rounds[1].

Recalling the definition of the counting logics from Section 2.4, we can now phrase the correspondence between the Weisfeiler-Leman algorithm, counting logics, and pebble games.

Theorem 3.18 ([30, 90]). *Let $k \in \mathbb{N}$. Let G and H be graphs, possibly vertex-coloured, with $|G| = |H|$ and let $\bar{u} \coloneqq (u_1,\dots,u_k) \in (V(G))^k$ and $\bar{v} \coloneqq (v_1,\dots,v_k) \in (V(H))^k$. Then for all $i \in \mathbb{N}$, the following are equivalent:*

[1] The pebble games in [77] are defined slightly differently. Still, a player has a winning strategy with r moves in the game described there if and only if they have one with r moves in our game and thus, Theorems 3.18 and 3.19 hold for both versions of the game.

1. $\chi^i_{G,k}(\bar{u}) = \chi^i_{H,k}(\bar{v})$;
2. $G \models \varphi(u_1, \ldots, u_k) \iff H \models \varphi(v_1, \ldots, v_k)$ *holds for every* C^{k+1}*-formula* $\varphi(x_1, \ldots, x_k)$ *of quantifier depth at most* i.
3. *Spoiler does not win the game* $\mathrm{BP}_{k+1}(G, H)$ *with the initial configuration* $(\bar{u}, \bar{v})$ *after at most* i *moves.*

Letting i be an integer such that $\chi^i_{G,k} = \chi_{G,k}$ and $\chi^i_{H,k} = \chi_{H,k}$, we can state the following result, which is due to Cai, Fürer, and Immerman.

Theorem 3.19 ([30, 90])**.** *Let* $k \in \mathbb{N}$. *Let* G *and* H *be graphs, possibly vertex-coloured, with* $|G| = |H|$ *and let* $\bar{u} := (u_1, \ldots, u_k) \in (V(G))^k$ *and* $\bar{v} := (v_1, \ldots, v_k) \in (V(H))^k$. *Then the following are equivalent:*

1. $\chi_{G,k}(\bar{u}) = \chi_{H,k}(\bar{v})$;
2. $G \models \varphi(u_1, \ldots, u_k) \iff H \models \varphi(v_1, \ldots, v_k)$ *holds for every* C^{k+1}*-formula* $\varphi(x_1, \ldots, x_k)$.
3. *Duplicator wins the game* $\mathrm{BP}_{k+1}(G, H)$ *with the initial configuration* $(\bar{u}, \bar{v})$.

We will see in the following chapters that it is very convenient to combine the three characterisations of the Weisfeiler-Leman algorithm when reasoning about its expressive power.

Recall that we say a graph G is *identified* by the logic C^k if for every second graph H, there is a sentence $\varphi \in \mathsf{C}^k$ which distinguishes G and H. This is equivalent to the existence of a sentence $\mathsf{iso}_G \in \mathsf{C}^k$ such that for all graphs H, we have $H \models \mathsf{iso}_G$ if and only if H is isomorphic to G. As direct consequences from Theorem 3.19, we obtain the following two corollaries.

Corollary 3.20. *For two graphs* G *and* H, *it holds that* $G \simeq_k H$ *if and only if* G *and* H *are* C^{k+1}*-equivalent.*

Corollary 3.21. *A graph has Weisfeiler-Leman dimension at most* k *if and only if it is identified by* C^{k+1}.

To start off, in the following two chapters, we examine the number of iterations required for the k-dimensional Weisfeiler-Leman algorithm to compute the stable colouring of its input. By Theorem 3.18, for two graphs G, H, the number of iterations required to distinguish G and H with the k-dimensional Weisfeiler-Leman algorithm essentially equals the quantifier depth of a distinguishing C^{k+1}-formula.

4 The Iteration Number of 1-WL

As described in Chapter 3, the Weisfeiler-Leman algorithm computes a stable colouring of its input graph, which, when compared with the colouring computed for a second graph, can be used to detect non-isomorphism of the two. To obtain the stable colouring, the algorithm proceeds in iterations. In this chapter, we investigate how long it takes for the algorithm to terminate, i.e. to obtain the stable colouring. As was shown by Immerman and Lander, the k-dimensional algorithm can be implemented to run in time $O(n^{k+1} \log n)$ [90]. However, here we are concerned rather with the number of iterations required to stabilise than with the total running time.

More specifically, for $n, k \in \mathbb{N}$, we are interested in $\mathrm{WL}_k(n)$, the maximum number of iterations required to reach stabilisation of the k-dimensional Weisfeiler-Leman algorithm among all graphs of order n. While not directly linked to the running time on a sequential machine, the iteration number corresponds to the parallel running time of Colour Refinement (on a standard PRAM model) [73, 109]. Furthermore, via the correspondence to counting logics (see Theorem 3.18), a bound on the iteration number for graphs of a fixed size directly translates into a bound on the descriptive complexity of the difference between the two graphs, namely into a bound on the quantifier depth of a distinguishing formula in the logic C^{k+1}.

For example, by definition of the refinement step of the Weisfeiler-Leman algorithm, on every regular uncoloured graph G as input, the 1-dimensional algorithm immediately stops: every vertex has the same colour and the same multiset of colours of neighbours. Thus, the vertices all look the same towards the algorithm. In fact, if G is vertex-transitive, then the unit partition is also the vertex orbit partition. Therefore, in this case, the vertex partition computed by Colour Refinement is the finest isomorphism-invariant one. However, for a regular graph that has at least two non-isomorphic connected components, the vertex orbit partition is strictly finer than the one computed by Colour Refinement.

On random graphs, Colour Refinement terminates asymptotically almost surely after 2 iterations [14]. Still, due to simple examples, it holds that $\mathrm{WL}_1(n) \geq \frac{n}{2} - 1$ (see Corollary 3.9). Krebs and Verbitsky improved this bound to $\mathrm{WL}_1(n) \geq n - 8\sqrt{n}$ [113]. However, concerning the upper bound, no improvement over the trivial $\mathrm{WL}_1(n) \leq n - 1$ (see Observation 3.6) has been shown. Note that this bound holds for every repeated partitioning of a set of size n and does not take into account any further properties of the input graph or of the algorithm used to execute the partitioning.

In this chapter, we prove via explicit constructions that, indeed, the bound is tight for infinitely many n (see Theorems 4.13 and 4.16), thus closing the gap between upper and lower bound.

Theorem 4.1. *There are infinitely many $n \in \mathbb{N}$ with* $\mathrm{WL}_1(n) = n - 1$.

Using the theorem, we can determine the iteration number up to an additive constant of 1 for all $n \in \mathbb{N}$ (where the precise numbers for $n \leq 9$ can easily be determined computationally), see Theorem 4.22.

The results in Sections 4.2 and 4.3 are based on joint work with Brendan McKay.

4.1 Previously Known Bounds

We recapitulate the known results and proof techniques for lower bounds on the iteration number for Colour Refinement, i.e. the 1-dimensional Weisfeiler-Leman algorithm.

For $n \in \mathbb{N}$, consider the uncoloured path on n vertices. In the first iteration of the execution of the algorithm on the path, the two end vertices are distinguished from all others because they are the only ones with degree 1. Then in each iteration, the information of being adjacent to a special vertex, i.e. the information about the distance to a vertex of degree 1, is propagated one step closer to the vertices in the centre of the path. This procedure takes $\lfloor \frac{n-1}{2} \rfloor$ iterations.

In 2015, Krebs and Verbitsky improved on this explicit linear lower bound by constructing pairs of graphs of order n which in C^2 can only be distinguished by formulae with quantifier depth larger than $n - 8\sqrt{n}$, resulting in the lower bound $\mathrm{WL}_1(n) \geq n - 8\sqrt{n}$ (see [113, Theorem 4.6]).

The general idea behind their construction is to exploit the linearity of the spreading of information on a path, as discussed above. However, in order to improve on the linear bound obtained from considering paths, the graphs are modified such that there are vertices that are far away from every (actually, *the*) end point and thus, the number of iterations is increased. By choosing the parameters of this modification optimally, the new lower bound is then obtained.

Together with the trivial upper bound from Observation 3.6, we can state that

$$n - 8\sqrt{n} \leq \mathrm{WL}_1(n) \leq n - 1$$

and it has remained open whether any of the two bounds is tight.

In his Master's thesis, Gödicke presented simplifications of the construction by Krebs and Verbitsky, still maintaining the asymptotic bound on the iteration number of $n - O(\sqrt{(n)})$ [52]. Preliminary research conducted together with Gödicke had led to the concept of combinatorial *split procedures* as a means to find graphs of a given size with a specific number of Colour Refinement iterations until stabilisation. A split procedure is an abstract model for the splitting of colour classes that happens during the execution of Colour Refinement on a graph. Thus, in order to find a graph on n vertices on which Colour Refinement takes j iterations to stabilise, we can improve upon the brute-force approach that would simply simulate the algorithm on every graph on n vertices. From every combinatorially possible split procedure for the given values n and j, by considering the regularity conditions that the colour classes impose on the graph, we can derive a linear program. If the program is solvable, there is a graph for the given split procedure and thus also a graph on n vertices with j Colour Refinement iterations.

We used the split procedure approach to reverse-engineer the splitting and check whether there are graphs for which the trivial upper bound on the iteration number of Colour Refinement is tight. Gödicke's implementation yields the following positive answer.

Theorem 4.2 ([52])**.** *For every $n \in \{1, 10, 11, 12\}$, there exists a graph of order n on which Colour Refinement stabilises only after $n - 1$ iterations. For $n \in [2, 9]$, there is no such graph.*

Unfortunately, due to computational exhaustion, it was not possible to test for larger graph sizes. Also, the obtained graphs do not exhibit any structural properties that would lend themselves for a generalisation to larger orders.

Using a fast implementation of Colour Refinement, we could verify that there are exactly 16 graphs G with $|G| - 1$ Colour Refinement iterations of order 10, 24 graphs of order 11, 32 of order 12, and 36 of order 13. However, again, with simple brute-force approaches, we could not go beyond this number exhaustively.

4.2 Compact Representations of Long-Refinement Graphs

In the light of the previous section, the question whether the lower bound obtained by Krebs and Verbitsky is asymptotically tight has remained open. With the brute-force approach, it becomes infeasible to test all graphs of orders much larger than 10 exhaustively for their number of Colour Refinement iterations until stabilisation. Still, knowing that there are graphs G with $|G| - 1$ Colour Refinement iterations, it is natural to ask whether the ones presented in [52] are just exceptions or whether there exist such long-refinement graphs also for larger graph sizes (and, in that case, if they can be generated systematically). In this section, we show that the latter is the case.

In the following, we call every graph G with $\mathrm{WL}_1(G) = |G| - 1$ a *long-refinement graph.* We also set $\deg(G) := \{\deg(v) \mid v \in V(G)\}$.

When the input is a coloured graph with at least two vertex colours, the initial partition already has two classes. Hence, all graphs G with $\mathrm{WL}_1(G) = |G| - 1$ are monochromatic. Therefore, in the following, all initial input graphs are considered to be monochromatic.

Proposition 4.3. *Let G be a graph and let $n := |G|$. If there exists an $i \in \mathbb{N}_0$ such that $|\{\chi_{G,1}^{i+1}(v) \mid v \in V(G)\}| - |\{\chi_{G,1}^{i}(v) \mid v \in V(G)\}| \geq 2$ holds, then $\mathrm{WL}_1(G) < n - 1$.*

Proof: For a partition π, every partition π' with $\pi \gneq \pi'$ satisfies $|\pi'| \geq |\pi| + 1$. Thus, for a sequence of partitions of the form

$$\pi_1 := \big\{\{v_1, \dots, v_n\}\big\} \gneq \pi_2 \gneq \dots \gneq \big\{\{v_1\}, \dots \{v_n\}\big\} =: \pi_n,$$

it must hold that $|\pi_i| = |\pi_{i-1}| + 1$ for every $i \in [2, n]$. □

The proposition implies that in order to find long-refinement graphs, we have to look for graphs in which in every Colour Refinement iteration, only one additional colour class appears. That is, in each iteration, only one colour class is split and the splitting creates exactly two new colour classes.

Corollary 4.4. *Let G be a long-refinement graph with at least two vertices. Then there exist $d_1, d_2 \in \mathbb{N}_0$ with $d_1 \neq d_2$ and such that $\deg(G) = \{d_1, d_2\}$.*

Proof: This is a direct consequence of Proposition 4.3: Every (uncoloured) regular graph G satisfies $\mathrm{WL}_1(G) = 0$ and if there were more than two vertex degrees present in G, we would have $|\{\chi_{G,1}^{1}(v) \mid v \in V(G)\}| - |\{\chi_{G,1}^{0}(v) \mid v \in V(G)\}| \geq 3 - 1 \geq 2$. □

We can thus restrict ourselves to graphs with exactly two vertex degrees. It is similarly easy to see that it suffices to consider connected graphs. The only connected graphs G with vertex degrees only 1 and 2 are paths and, by Fact 3.8, they are not long-refinement graphs.

Thus, the smallest degree pairs for a search for candidates are $\{1, 3\}$ and $\{2, 3\}$.

For the remainder of this chapter, we introduce the following notation.

Notation 4.5. *For a graph G and $i \in \mathbb{N}_0$, we let π_G^i denote the partition induced by $\chi_{G,1}^i$ on $V(G)$, i.e. after i Colour Refinement iterations on G. If G is clear from the context, we omit it in the expression.*

Recall that for vertex sets $V_1, V_2 \subseteq V(G)$, we denote by $G[V_1, V_2]$ the graph with vertex set $V_1 \cup V_2$ and edge set $E(G) \cap \big\{\{v_1, v_2\} \,\big|\, v_1 \in V_1, v_2 \in V_2\big\}$.

Lemma 4.6. *Let G be a long-refinement graph. Then $|\{v \in V(G) \mid \deg(v) = 1\}| \leq 2$.*

Proof: Suppose the lemma does not hold. Let G be a long-refinement graph with at least three vertices of degree 1. Consider the execution of Colour Refinement on input G and let $n := |G|$. In π^1, there are two vertex colour classes, namely a class V_1 containing the vertices of degree 1 and a class V_d containing the vertices of the second vertex degree $d \neq 1$. In the partition π^2, the class V_d is replaced with $N(V_1)$ and $V_d \setminus N(V_1)$.

Suppose that $|V_1| \geq 2$. The class V_1 is not split before $N(V_1)$ has been split. Thus, consider the iteration j after which $N(V_1)$ has been subdivided into two classes W_1 and W_2. This induces the splitting of V_1 into $N(W_1) \cap V_1$ and $N(W_2) \cap V_1$, which by Proposition 4.3 implies in particular that for all pairs of partition classes $C, C' \in \pi^{j-1}$ with $C \cap V_1 = \emptyset = C' \cap V_1$, the graph induced between the two classes is biregular. Therefore, however, now for every pair of classes $C, C' \in \pi^j$, the graph $G[C, C']$ is biregular and thus, the partition is equitable. Hence, $j = n - 1$, i.e. the splitting of V_1 must happen in the $(n-1)$-st iteration. In particular, $N(W_1) \cap V_1$ and $N(W_2) \cap V_1$ must be singletons, i.e. $|V_1| = 2$. □

Table 4.1: Adjacency lists of long-refinement graphs G with $\deg(G) = \{1, 5\}$ (left) and $\deg(G) = \{1, 3\}$ (right), respectively.

v	$N(v)$
0	1
1	0,2,3,4,5
2	1,3,5,7,10
3	1,2,4,6,10
4	1,3,5,9,11
5	1,2,4,8,11
6	3,7,8,9,11
7	2,6,8,9,10
8	5,6,7,10,11
9	4,6,7,10,11
10	2,3,7,8,9
11	4,5,6,8,9

v	$N(v)$
0	1
1	0,2,3
2	1,11,13
3	1,10,12
4	5,7,10
5	4,6,10
6	5,9,11
7	4,8,11
8	7,9,13
9	6,8,12
10	3,4,5
11	2,6,7
12	3,9,13
13	2,8,12

Table 4.1 displays the adjacency lists of two long-refinement graphs on 12 and 14 vertices, respectively, which each have exactly one vertex of degree 1.

As the following corollary states, the lemma allows us to reduce the decision problem whether there are long-refinement graph with degrees in $\{1, 2, 3\}$ to the question whether there are such graphs with degrees in $\{2, 3\}$.

Corollary 4.7. *If there is a long-refinement graph G with $\deg(G) = \{1, 3\}$, then there is also a long-refinement graph $\widehat{G}$ with $\deg(\widehat{G}) = \{2, 3\}$ and $|\widehat{G}| \in \{|G|-1, |G|\}$.*

Proof: Let G be a long-refinement graph with $\deg(G) = \{1, 3\}$. Then $\pi^1 = \{V_1, V_3\}$, where $V_1 = \{v \in V(G) \mid \deg(v) = 1\}$ and $V_3 = \{v \in V(G) \mid \deg(v) = 3\}$. By Lemma 4.6, it holds that $|V_1| \in \{1, 2\}$.

First suppose $|V_1| = 2$. Consider the graph $\widehat{G}$ with $V(\widehat{G}) = V(G)$ and $E(\widehat{G}) = E(G) \cup \{V_1\}$, i.e. obtained from G by inserting an edge between the two vertices in

V_1. In the following, we identify the vertices of $\widehat{G}$ with their counterparts in G. For $i \in \mathbb{N}_0$, let $\widehat{\pi}^i$ be the partition of $V(\widehat{G})$ induced by $\chi^i_{\widehat{G},1}$. Let $n := |G|$. Then, for $i \in [0, n-1]$, it holds that

$$\widehat{\pi}^i = \pi^i.$$

This follows from $\widehat{G} - V_1 = G - V_1$ and $\widehat{G}[V_1, N(V_1)] = G[V_1, N(V_1)]$, the regularity of $\widehat{G}[V_1]$ is regular and that there is only one way to split V_1, which results in two singletons. In particular, it holds that $\mathrm{WL}_1(\widehat{G}) = \mathrm{WL}_1(G) = |\widehat{G}| - 1$.

Now suppose $|V_1| = 1$. In π^1, there are only the two partition classes V_1 and V_3. In π^2, the set V_3 is subdivided into the singleton $N(V_1)$ and $V_3 \setminus N(V_1)$. Define $\widehat{G} := G - V_1]$ and again, for $i \in \mathbb{N}_0$, let $\widehat{\pi}^i$ be the partition of $V(\widehat{G})$ induced by $\chi^i_{\widehat{G}}$. Then $\widehat{\pi}^1 = \{N(V_1), V_3 \setminus N(V_1)\} = \pi^2 \setminus \{V_1\}$ and, more generally, for $i \in \mathbb{N}$, we obtain $\widehat{\pi}^i = \pi^{i+1} \setminus \{V_1\}$. This can be deduced from the equality $\widehat{G} - V_1 = G - V_1$. Thus, $\mathrm{WL}_1(\widehat{G}) = (n-1) - 1 = |\widehat{G}| - 1$. □

Thus, in order to decide whether there are infinitely many graphs G with $\mathrm{WL}_1(G) = |G| - 1$ and degrees in $\{1, 2, 3\}$, it suffices to look for graphs with degrees $\{2, 3\}$. In the pursuit of that goal, with the help of the tool `Nauty` [126], our quest for long-refinement graphs with degrees 2 and 3 was successful. Exploiting the degree restrictions and some other conditions that we imposed to render the search tractable, it was possible to test for graphs up to orders around 60. We found graphs G with $n - 1$ Colour Refinement iterations, where $n = |G|$, for all even $n \in [10, 64] \setminus \{24, 30, 42, 48, 60\}$ and for all odd $n \in [11, 63] \setminus \{21, 27, 39, 45, 57, 63\}$.

In the following, in order to generalise the results to bigger graph sizes, we analyse the obtained graphs. Among our computational results, the even-size graphs G with $|G| \geq 12$ have the following property in common: there is an iteration j such that for every $c \in \chi^j_{G,1}(V(G))$, it holds that

$$|\{v \in V(G) \mid \chi^j_{G,1}(v) = c\}| = 2.$$

That is, with respect to their assigned colours, the vertices remain in pairs until there are no larger colour classes left. Then the first such pair is split into singletons, which must induce a splitting of another pair, and so on, until the discrete partition is obtained. (Similar statements hold for the odd-size graphs, but are more technical. We therefore first focus on the even-size graphs.) In the following, a *pair* is a set of two vertices which occurs as a colour class during the execution of Colour Refinement. That is, vertices v, v' form a pair if and only if $\{v, v'\}$ is contained in π^i for some $i \in \mathbb{N}_0$.

As just argued, there is a splitting order on the pairs, i.e. a linear order $\prec$ induced by the order in which pairs are split into singletons. We now examine the possible connections between pairs. As a result of the regularity conditions that must hold for the graph induced within and between colour classes, we make the following observation. It implies that, in a long-refinement graph, to determine the class C

that is split in iteration i, it suffices to consider the neighbourhood of an arbitrary class obtained in the preceding iteration.

Lemma 4.8. *Let G be a graph, let $i \in \mathbb{N}$ and suppose there are C_1, C_2 with $\{C_1, C_2\} = \pi^i \setminus \pi^{i-1}$. Suppose $\pi^i \setminus \pi^{i+1} \neq \emptyset$ and let $C' \in \pi^i \setminus \pi^{i+1}$.*

Then there are vertices $v_1', v_2' \in C'$ such that $|N(v_1') \cap C_1| \neq |N(v_2') \cap C_1|$.

Proof: Note that there must be a $C \in \pi^{i-1} \setminus \pi^i$ with $C_1 \cup C_2 = C$. Since $C' \in \pi^i$ and $C \in \pi^{i-1}$, there is $d \in \mathbb{N}_0$ such that for every $v \in C'$, it holds that $d = |N(v) \cap C|$. Since $\{C_1, C_2\} = \pi^i \setminus \pi^{i-1}$ and $C' \notin \pi^{i+1}$, there are vertices $v_1', v_2' \in C'$ such that $|N(v_1') \cap C_1| \neq |N(v_2') \cap C_1|$ or $|N(v_1') \cap C_2| \neq |N(v_2') \cap C_2|$. In the first case, we are done. In the second case, letting $d_i := |N(v_i') \cap C_2|$, we obtain $|N(v_1') \cap C_1| = d - d_1 \neq d - d_2 = |N(v_2') \cap C_1|$. □

Note that the validity of the lemma depends on the assumption $\{C_1, C_2\} = \pi^i \setminus \pi^{i-1}$, which by Proposition 4.3 is always fulfilled in long-refinement graphs as long as $\pi^{i-1} \neq \pi^i$. (If C is split into an arbitrary number of vertex colour classes, it also suffices to consider the classes obtained in the previous iteration. However, in that case, C_1 cannot be chosen arbitrarily in general.)

From now on, we make the following assumption.

Assumption 4.9. *G is a long-refinement graph with $\deg(G) = \{2, 3\}$ and such that there is an $i \in \mathbb{N}_0$ for which π^i contains only pairs. Let $\prec$ be the splitting order of these pairs.*

We call pairs $P_1, P_2 \subseteq V(G)$ *successive* if P_2 is the successor of P_1 with respect to $\prec$. Note that for successive pairs P_1, P_2, in the graph $G[P_1, P_2]$, every $v_2 \in P_2$ must have the same number of neighbours in P_1, otherwise it would hold that $P_2 \prec P_1$. By a simple case analysis, together with an application of Lemma 4.8, this rules out all connections but matchings for successive pairs.

Corollary 4.10. *Let P_1 and P_2 be successive pairs. Then $G[P_1, P_2]$ is a matching.*

Towards a compact representation of the graphs, we further examine the connections between pairs P_1 and P_2 with $S(P_1) \prec P_2$, where $S(P_1)$ is the successor of P_1 with respect to $\prec$.

Lemma 4.11. *Let P_1 be a pair. Then exactly one of the following holds.*

- *$P_1 \neq \min(\prec)$ and for every pair P_2 which satisfies $S(P_1) \prec P_2$, it holds that $E(G[P_1, P_2]) = \emptyset$.*
- *$P_1 = \min(\prec)$ and there are exactly two choices P_2, P_2' for a pair P' with $S(P_1) \prec P'$ such that $E(G[P_1, P']) \neq \emptyset$. Furthermore, there is a vertex $v_1 \in P_1$ such that $G[\{v_1\}, P_2]$ and $G[P_1 \setminus \{v_1\}, P_2']$ are complete bipartite and $E(G[\{v_1\}, P_2']) = E(G[P_1 \setminus \{v_1\}, P_2]) = \emptyset$.*

Proof: Suppose $P_1 \neq \min(\prec)$. If $P_1 = \max(\prec)$, the statement trivially holds. Otherwise, by Corollary 4.10, every vertex $v_1 \in P_1$ has exactly one neighbour in $S(P_1)$ and exactly one neighbour in the predecessor of P_1, i.e. in the unique pair $A(P_1)$ such that $P_1 = S(A(P_1))$. Thus, due to the degree restrictions, v_1 has at most one neighbour in a pair P' with $P_1 \prec P'$ and $P' \neq S(P_1)$. However, if v_1 had a neighbour in such a P', the graph $G[\{v_1\}, P']$ would not be biregular, implying that $P' = S(P_1)$, a contradiction. Therefore, $N(v_1) \subseteq A(P_1) \cup P_1 \cup S(P_1)$ and thus, $N(P_1) \subseteq A(P_1) \cup S(P_1)$. In particular, for every pair P_2 with $P_1 \prec P_2$ and $P_2 \neq S(P_1)$, it holds that $E(G[P_1, P_2]) = \emptyset$.

Now suppose that $P_1 = \min(\prec)$. Since the splitting of P_1 must be induced by a splitting of a union of two pairs and $G[P_1, S(P_1)]$ is biregular and $G[P_1]$ is regular, we cannot have $N(P_1) \subseteq S(P_1)$. Thus, there is a pair P_2 with $S(P_1) \prec P_2$ and such that $E(G[P_1, P_2]) \neq \emptyset$. Let $v_1 \in P_1$ be a vertex with $N(v_1) \cap P_2 \neq \emptyset$. Then $P_2 \subseteq N(v_1)$, otherwise $P_2 = S(P_1)$. Thus, $G[\{v_1\}, P_2]$ is complete bipartite. Therefore and due to the degree restrictions, v_1 has exactly three neighbours: one in $S(P_1)$ and two in P_2. In particular, for every pair P_2' with $P_2 \neq P_2' \neq S(P_1)$, it holds that $E(G[\{v_1\}, P_2']) = \emptyset$.

Let $v_1' \neq v_1$ be the second vertex in P_1. Since the splitting of P_1 induces the splitting of $S(P_1)$, by Proposition 4.3, for every pair P' with $P_1 \neq P' \neq S(P_1)$, the graph $G[\{v_1'\}, P']$ must be biregular, i.e. either empty or complete bipartite.

Moreover, since $\deg(v_1) = 3$, also $\deg(v_1') = 3$. By Corollary 4.10, $|N(v_1') \cap S(P_1)| = 1$. Therefore, there is exactly one pair P_2' such that $G[\{v_1'\}, P_2']$ is complete bipartite and for all other pairs P' with $P_1 \neq P' \neq S(P_1)$, the graph $G[\{v_1'\}, P']$ is empty.

Suppose $P_2' = P_2$. Choose i such that $\pi^i \setminus \pi^{i+1} = \{P_1\}$. Then the unique element in $\pi^{i-1} \setminus \pi^i$ is a union of two pairs, whose splitting induces the splitting of P_1. However, $N(P_1) = S(P_1) \cup P_2$ and both graphs $G[P_1, S(P_1)]$ and $G[P_1, P_2]$ are biregular.

Thus, $P_2' \neq P_2$, which concludes the proof. □

Corollary 4.10 and Lemma 4.11 characterise $G[P_1, P_2]$ for all pairs $P_1 \neq P_2$. Thus, all additional edges must be between vertices from the same pair. Hence, we can use the following compact graphical representation to fully describe the graphs of order at least 12 that we found. As vertex set, we take the pairs. We order those according to $\prec$ and connect successive pairs with an edge representing the matching. If the two vertices of a pair are adjacent, we indicate this with a loop at the corresponding vertex. The only other type of connection between pairs is constituted by the edges from $\min(\prec)$ to two other pairs which form the last colour class of size 4, i.e. a colour class of size 4 in the partition π^i for which $\pi^{i+1} \setminus \pi^{i+2} = \{\min(\prec)\}$. We indicate this type of edge with a dotted line.

An example graph as well as the evolution of the colour classes computed by Colour Refinement on the graph is depicted in Figure 4.1.

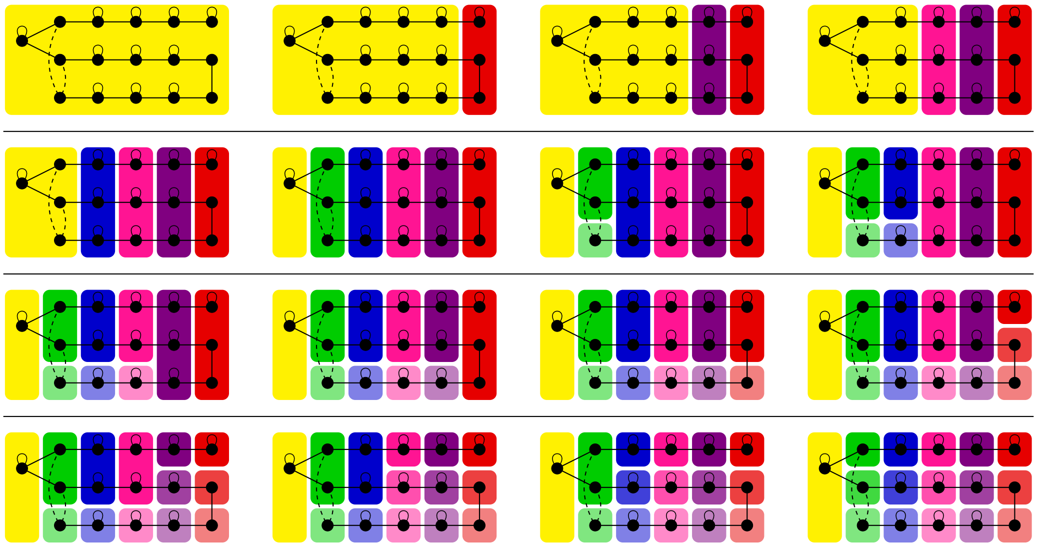


Figure 4.1: Top left: An input graph G with $\mathrm{WL}_1(G) = |G| - 1$. The subsequent pictures show the partitions of $V(G)$ after the first 15 Colour Refinement iterations. There are 16 further iterations not depicted here, which consist in the splitting of the pairs into singletons.

Notation 4.12. *Since $\prec$ is a linear order, we can also use a string representation to fully describe the graph. For this, we introduce the following notation, letting $A(P)$ and $S(P)$ be the predecessor and successor of P, respectively, with respect to $\prec$.*

- *0 represents a pair of vertices of degree 2.*
- *1 represents a pair P of vertices of degree 3 that is not the minimum of $\prec$ and for which $N(P) \subseteq A(P) \cup S(P)$. (This implies that $P \in E(G)$.)*
- *X represents a pair P of vertices of degree 3 that is not the minimum of $\prec$ and for which $N(P) \not\subseteq A(P) \cup S(P)$.*
- *S represents the minimum of $\prec$.*

Thus, by Lemma 4.11, there are exactly two pairs of type X, namely P_2 and P_2' from the lemma. Now we can use the alphabet $\Sigma = \{0, 1, \mathrm{S}, \mathrm{X}\}$ and the order $\prec$ to encode the graphs in strings. The i-th letter of a string is the i-th element of $\prec$. Note that S is always a pair of non-adjacent vertices of degree 3 due to the degree restrictions. For example, the string representation for the graph in Figure 4.1 is S11100111X1X1110.

Formally, for every $\ell \geq 2$ and every string $\Xi\colon [\ell] \to \{0, 1, \mathrm{S}, \mathrm{X}\}$ with $\Xi(1) = \mathrm{S}$ and $\Xi^{-1}(\mathrm{X}) = \{r, r'\}$ for some $r, r' \in [\ell]$ with $r < r'$, we define the corresponding graph

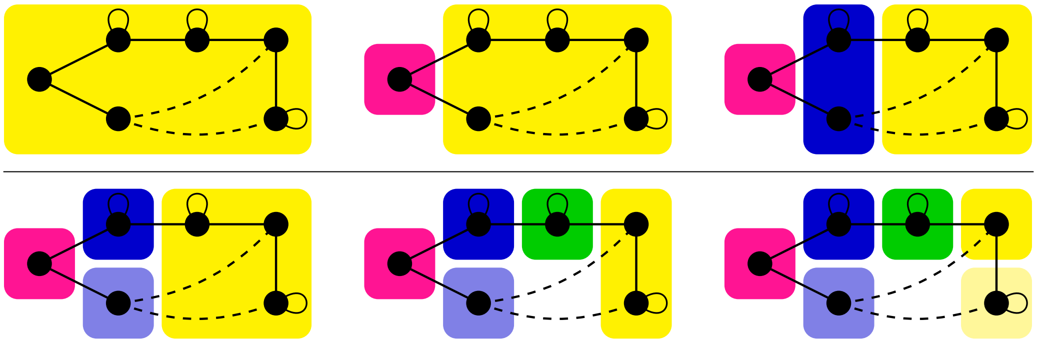


Figure 4.2: A visualisation of the graph with string representation S011XX and the evolution of the colour classes in the first 5 Colour Refinement iterations on the graph.

$G \coloneqq G(\Xi)$ with $V(G) = \{v_{i,j} \mid i \in [\ell], j \in [2]\}$ and

$$\begin{aligned} E(G) = \ &\big\{\{v_{i,1}, v_{i,2}\} \,\big|\, i \in [\ell], \Xi(i) = 1\big\} \cup \\ &\big\{\{v_{i,j}, v_{i+1,j}\} \,\big|\, i \in [\ell - 1], j \in [2]\big\} \cup \\ &\big\{\{v_{r,j}, v_{1,1}\} \,\big|\, j \in [2]\big\} \cup \\ &\big\{\{v_{r',j}, v_{1,2}\} \,\big|\, j \in [2]\big\}. \end{aligned}$$

We use this encoding in the next section, which contains the main results of this chapter.

4.3 Infinite Families of Long-Refinement Graphs

Here we present infinite families of graphs G satisfying $\mathrm{WL}_1(G) = |G| - 1$. We adapt them further to deduce that $\mathrm{WL}_1(n) \geq n - 2$ holds for all $n \in \mathbb{N}_{\geq 10}$.

For $w \in \{0,1\}^*$, the notation $(w)^k$ abbreviates the k-fold concatenation of w. We let $1^k \coloneqq (1)^k$.

Theorem 4.13. *For every string Ξ contained in the following sets, the graph $G \coloneqq G(\Xi)$ satisfies $\mathrm{WL}_1(G) = |G| - 1$.*

- $\{\mathrm{S011XX}\}$
- $\{\mathrm{S}1^k001^k\mathrm{X1X}1^k0 \mid k \in \mathbb{N}_0\}$
- $\{\mathrm{S11}1^k001^k\mathrm{XX}1^k0 \mid k \in \mathbb{N}_0\}$
- $\{\mathrm{S}1^k001^k\mathrm{1XX1}1^k0 \mid k \in \mathbb{N}_0\}$
- $\{\mathrm{S011}(011)^k00(110)^k\mathrm{XX}(011)^k0 \mid k \in \mathbb{N}_0\}$

- $\{\mathrm{S}(011)^k00(110)^k1\mathrm{X}0\mathrm{X}1(011)^k0 \mid k \in \mathbb{N}_0\}$

Proof: Let $G \coloneqq G(\mathrm{S}011\mathrm{XX})$ (cf. Figure 4.2). The vertices $v_{2,1}$ and $v_{2,2}$ are the only ones of degree 2. Thus,

$$\pi^1 = \big\{\{v_{2,1}, v_{2,2}\}, V(G) \setminus \{v_{2,1}, v_{2,2}\}\big\},$$
$$\pi^2 = \big\{\{v_{2,1}, v_{2,2}\}, \{v_{i,j} \mid i \in \{1,3\}, j \in [2]\}, \{v_{i,j} \mid i \in [4,6], j \in [2]\}\big\}.$$

Then

$$\pi^3 = \big\{\{v_{1,1}, v_{1,2}\}, \{v_{2,1}, v_{2,2}\}, \{v_{3,1}, v_{3,2}\}, \{v_{i,j} \mid i \in [4,6], j \in [2]\}\big\},$$

since the vertices in the S-pair have no neighbours in $\{v_{i,j} \mid i \in \{1,3\}, j \in [2]\}$. Similarly,

$$\pi^4 = \big\{\{v_{1,1}, v_{1,2}\}, \{v_{2,1}, v_{2,2}\}, \{v_{3,1}, v_{3,2}\}, \{v_{4,1}, v_{4,2}\}, \{v_{i,j} \mid i \in [5,6], j \in [2]\}\big\},$$
$$\pi^5 = \big\{\{v_{i,j} \mid j \in [2]\} \mid i \in [6]\big\}.$$

Now the splitting of the last colour class of size 4 into two X-pairs induces the splitting of the S-pair into singletons, which is propagated linearly according to $\prec$, adding 6 further iterations, thus summing up to 11 iterations.

We now consider the various infinite families of graphs. The proofs for them work similarly by induction over k. Therefore, we only present the full detailed proof for the family $\{\mathrm{S}1^k001^k\mathrm{X}1\mathrm{X}1^k0 \mid k \in \mathbb{N}_0\}$, which includes the graph from Figure 4.1.

For $k = 0$, the graph $G_0 \coloneqq G(\mathrm{S}00\mathrm{X}1\mathrm{X}0)$ has 14 vertices. It is easy to verify that it indeed takes 13 Colour Refinement iterations to stabilise. We sketch how Colour Refinement processes the graph: for this, for $i \in \mathbb{N}_0$, we let π_0^i denote the partition of $V(G_0)$ induced by $\chi^i_{G_0,1}$, i.e. after i iterations of Colour Refinement on G_0. First, vertices are assigned colours indicating their degrees. That is,

$$\pi_0^1 = \big\{\{v_{i,j} \mid i \in \{2,3,7\}, j \in [2]\}, \{v_{i,j} \mid i \in \{1,4,5,6\}, j \in [2]\}\big\}.$$

Now

$$\pi_0^2 = \big\{\{v_{i,j} \mid i \in \{2,3,7\}, j \in [2]\}, \{v_{i,j} \mid i \in \{1,4,6\}, j \in [2]\}, \{v_{5,i} \mid i \in [2]\}\big\},$$

since the vertices contained in the 1-pair are not adjacent to vertices from 0-pairs. Since no vertex contained in the S-pair is adjacent to any vertex from the 1-pair, we obtain

$$\pi_0^3 = \big\{\{v_{i,j} \mid i \in \{2,3,7\}, j \in [2]\}, \{v_{i,j} \mid i \in \{4,6\}, j \in [2]\}, \{v_{5,i} \mid j \in [2]\},$$
$$\{v_{1,j} \mid j \in [2]\}\big\}.$$

Furthermore,

$$\begin{aligned}\pi_0^4 &= \big\{\{v_{i,j} \mid i \in \{3,7\}, j \in [2]\}, \{v_{i,j} \mid i \in \{4,6\}, j \in [2]\}, \{v_{5,i} \mid j \in [2]\},\\ &\qquad \{v_{1,j} \mid j \in [2]\}, \{v_{2,j} \mid j \in [2]\}\big\},\\ \pi_0^5 &= \big\{\{v_{7,j} \mid j \in [2]\}, \{v_{i,j} \mid i \in \{4,6\}, j \in [2]\}, \{v_{5,i} \mid j \in [2]\},\\ &\qquad \{v_{1,j} \mid j \in [2]\}, \{v_{2,j} \mid j \in [2]\}, \{v_{3,j} \mid j \in [2]\}\big\},\\ \pi_0^6 &= \big\{\{v_{i,j} \mid j \in [2]\} \mid i \in [7]\big\},\end{aligned}$$

i.e. with respect to the order $\prec$ induced by the string representation, the first 0-pair, the second 0-pair and the first X-pair are separated from the others. Once the two X-pairs form separate colour classes, this induces the splitting of S into two singletons, which is propagated linearly through the entire string, adding 7 further iterations, thus summing up to 13 iterations.

For general $k \geq 1$, let $G_k := G(\mathrm{S}1^k001^k\mathrm{X}1\mathrm{X}1^k0)$. To count the iterations of Colour Refinement, we introduce some vocabulary for the pairs in G_k (see also Figure 4.1). We let $V := \{v_{i,j} \mid i \in [2, k+1] \cup [k+4, 2k+3] \cup [2k+7, 3k+6], j \in [2]\}$. Note that V is the set of vertices contained in the subgraphs corresponding to the substrings 1^k in the string representation. Furthermore, for all $i \in [k+2]$, we call the set $\{v_{i',j} \mid i' \in \{i, 2k+5-i, 2k+5+i\}, j \in [2]\}$ the *i-th column* and denote it by V_i. The *0-th column* is the set $\{v_{2k+5,j} \mid j \in [2]\}$. Thus,

$$V = \bigcup_{2 \leq i \leq k+1} V_i.$$

For every $j \in [2]$, the sets $\{v_{i,j} \mid i \in [1, k+2]\}$, $\{v_{i,j} \mid i \in [k+3, 2k+4]\}$, and $\{v_{i,j} \mid i \in [2k+6, 3k+7]\}$ are called *rows*. In accordance with Figure 4.1, we fix an ordering on the rows: the *first row* is $\{v_{i,1} \mid i \in [2k+6, 3k+7]\}$, the *second row* is $\{v_{i,2} \mid i \in [2k+6, 3k+7]\}$, ..., the *sixth row* is $\{v_{i,2} \mid i \in [1, k+2]\}$. To be able to refer to the vertices in V and the adjacent columns more easily, we relabel them: for $i \in [k+2], j \in [6]$, the vertex $w_{i,j}$ is defined to be the unique vertex in the i-th column and the j-th row.

The following observation is the crucial insight for counting the iterations of Colour Refinement on G_k. We will use it to show that, informally stated, the subgraph $G_k[V]$ delays the propagation of the splitting of the colour classes in the remainder of the graph by k iterations whenever the splitting of a colour class contained in V_1 or V_{k+2} initiates a splitting of a colour class contained in V.

Claim 1. Consider a colouring λ of G_k and its induced partition π_k of $V(G_k)$. For $t \in \mathbb{N}_0$, let π_k^t be the partition induced by $\chi^t_{G_k,1}$ on input (G_k, λ). Suppose G_k, λ, π_k satisfy the following conditions.

1. There exist $\ell \in [6]$ and $I_1, \ldots, I_\ell \subseteq [6]$ such that $\bigcup_{i \in [\ell]} I_i = [6]$ and $I_i \cap I_j = \emptyset$ for $1 \leq i < j \leq \ell$ and for every $i \in [\ell]$, it holds that $\{w_{k+2,j'} \mid j' \in I_i\} \in \pi_k^0$. That is, V_{k+2} is a union of colour classes with respect to λ.

2. $\pi_k^1 = \big\{\{w_{k+1,j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\big\} \cup \big\{C \setminus V_{k+1} \,\big|\, C \in \pi_k, C \setminus V_{k+1} \neq \emptyset\big\}$.
3. For all $C, C' \subseteq V_{k+1}$ with $C, C' \in \pi_k^1$, the graph $G_k[C]$ is regular and $G_k[C, C']$ is biregular.

Then for every $t \in [k]$, it holds that

$$\begin{aligned}\pi_k^t = \bigcup_{i' \in [t]} \big\{\{w_{k+2-i',j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\big\} \cup \\ \left\{C \setminus \Big(\bigcup_{i' \in [t]} V_{k+2-i'}\Big) \,\middle|\, C \in \pi_k^0, C \setminus \Big(\bigcup_{i' \in [t]} V_{k+2-i'}\Big) \neq \emptyset\right\}.\end{aligned}$$

Proof of the claim: We show the claim via induction. For $t = 1$, the statement is exactly the second item from the assumptions. For the inductive step, suppose the statement holds for all $t' \leq t$ for some $t \in [k-1]$. We show that it also holds for $t + 1$.

Note that the right-hand side of the equation is a partition of $V(G_k)$. Thus, it suffices to show "$\supseteq$", i.e. that the right-hand side is contained in the left-hand side of the equation.

Since V_{k+2} is a union of elements of π_k^0, it is also a union of elements of π_k^t. Thus, by the induction hypothesis,

$$\begin{aligned}\{C \mid C \in \pi_k^0, C \cap V_{k+2} \neq \emptyset\} &= \{C \mid C \in \pi_k^0, C \subseteq V_{k+2}\} \\ &\subseteq \left\{C \setminus \Big(\bigcup_{i' \in [t]} V_{k+2-i'}\Big) \,\middle|\, C \in \pi_k^0, C \setminus \Big(\bigcup_{i' \in [t]} V_{k+2-i'}\Big) \neq \emptyset\right\} \\ &\subseteq \pi_k^t. \qquad (4.1)\end{aligned}$$

Similarly, using the induction hypothesis for $t-1$, all other V_i with $i \geq k+2-(t-1)$ are unions of elements of π_k^{t-1} (if $t = 1$, this holds trivially). Thus,

$$\begin{aligned}\left\{C \,\middle|\, C \in \pi_k^{t-1}, C \cap \bigcup_{i' \in [t-1]} V_{k+2-i'} \neq \emptyset\right\} &= \left\{C \,\middle|\, C \in \pi_k^{t-1}, C \subseteq \bigcup_{i' \in [t-1]} V_{k+2-i'}\right\} \\ &= \bigcup_{i' \in [t-1]} \big\{\{w_{k+2-i',j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\big\} \\ &\subseteq \pi_k^t.\end{aligned}$$

This means that all elements from π_k^{t-1} that have a non-empty intersection with the set $\bigcup_{i' \in [0,t-1]} V_{k+2-i'}$ are also present in π_k^t. Therefore, for all $C, C' \in \pi_k^t$ with $C \cap \bigcup_{i' \in [0,t-1]} V_{k+2-i'} \neq \emptyset \neq C' \cap \bigcup_{i' \in [0,t-1]} V_{k+2-i'}$, the graph $G_k[C]$ must be regular and $G_k[C, C']$ must be biregular. (Otherwise, at least one of these classes would have been split in the t-th iteration.)

Actually, this also holds when relaxing the restriction for C' to have a non-empty intersection with $\bigcup_{i' \in [0,t]} V_{k+2-i'}$. Indeed, $G_k[V_{k+2-t}, V_{k+2-(t-1)}]$ is a matching between vertices contained in equal rows and, by the induction hypothesis for t, it holds that $\{C \mid C \in \pi_k^t, C \subseteq V_{k+2-(t-i')}\} = \big\{\{w_{k+2-(t-i'),j'} \mid j' \in I_i\} \mid i \in [\ell]\big\}$ for $i' \in \{0,1\}$. Thus, for all $C, C' \in \pi_k^t$ with $C \subseteq V_{k+2-(t-1)}$ and $C' \subseteq V_{k+2-t}$, the graph $G_k[C, C']$ is either a perfect matching or empty. In particular, it is biregular.

Note that there are no edges between vertices from columns V_i, $V_{i'}$ with $|i - i'| \geq 2$. Hence, in fact, for every $C \in \pi_k^t$ with $C \cap \bigcup_{i' \in [0,t-1]} V_{k+2-i'} \neq \emptyset$ and for all $C' \in \pi_k^t$, the graph $G_k[C]$ is regular and $G_k[C, C']$ is biregular. Thus, every $C \in \pi_k^t$ with $C \cap \bigcup_{i' \in [0,t-1]} V_{k+2-i'} \neq \emptyset$ is present in π_k^{t+1}. This shows that

$$\bigcup_{i' \in [t-1]} \big\{\{w_{k+2-i',j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\big\} \subseteq \pi_k^{t+1}. \tag{4.2}$$

and, using (4.1), that

$$\big\{C \,\big|\, C \in \pi_k^0, C \cap V_{k+2} \neq \emptyset\big\} \subseteq \pi_k^{t+1}. \tag{4.3}$$

We have seen that for $C, C' \in \pi_k^t$ with $C \subseteq \bigcup_{i' \in [0,t-1]} V_{k+2-i'}$ and $C' \subseteq V_{k+2-t}$, the graph $G[C]$ is regular and $G[C, C']$ is biregular. We now show that we can actually relax the location restriction for C to $C \subseteq \bigcup_{i' \in [0,t]} V_{k+2-i'}$. For this, note that all subgraphs $G_k[V_i]$ with $i \in [2, k+1]$ have the same structure. That is, for all $r, s \in [6]$ and all $i, j \in [2, k+1]$, the vertices in V_i in rows r and s are adjacent if and only if the corresponding ones are adjacent in V_j. Furthermore, by the induction assumption for t, it holds that $\{C \mid C \in \pi_k^t, C \subseteq V_{i'}\} = \big\{\{w_{i',j'} \mid j' \in I_i\} \mid i \in [\ell]\big\}$ for $i' \in \{k+2-t, k+1\}$. Thus, by Conditions (2) and (3) from the prerequisites of the claim, for all $C, C' \subseteq V_{k+2-t}$ with $C, C' \in \pi_k^t$, the graph $G_k[C]$ is regular and $G_k[C, C']$ is biregular.

In order to determine the colour classes of $\chi_{G_k,1}^{t+1}$ contained in V_{k+2-t}, we still need to analyse the structure of the graph $G_k[V_{k+1-t}, V_{k+2-t}]$ with respect to π_k^t. To this end, for $i \in [t+1]$ and $j \in [0, k+2]$, set $M_j^i := \{C \cap V_j \mid C \in \pi_k^i, C \cap V_j \neq \emptyset\}$. That is, M_j^i is the partition of V_j induced by π_k^i.

The graph $G_k[V_{k+2-(t+1)}, V_{k+2-t}]$ is a matching between vertices contained in equal rows and furthermore, by the induction assumption, we know $\{C \mid C \in \pi_k^t, C \subseteq V_{k+2-t}\} = \big\{\{w_{k+2-t,j'} \mid j' \in I_i\} \mid i \in [\ell]\big\}$. Therefore,

$$M_{k+1-t}^{t-1} \succeq \big\{\{w_{k+1-t,j'} \mid j' \in I_i\} \mid i \in [\ell]\big\}.$$

However, the induction assumption yields $M_j^{i-1} = M_j^i$ for all $i \in [t], j \in [0, k+1-t]$. In particular, the partition of V_{k+1-t} induced by π_k^t is not strictly finer than the one induced by π_k^{t-1}. Thus,

$$M_{k+1-t}^{t} \succeq \big\{\{w_{k+1-t,j'} \mid j' \in I_i\} \mid i \in [\ell]\big\}. \tag{4.4}$$

Therefore, again using the induction hypothesis for t, we obtain

$$\big\{\{w_{k+2-t,j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\big\} \subseteq \pi_k^{t+1}. \tag{4.5}$$

Since $M_{k+1-t}^{t-1} = M_{k+1-t}^t$ and $M_{k-t}^{t-1} = M_{k-t}^t$, from (4.4), we know $M_{k-t}^t \succeq M_{k+1-t}^t \succeq M_{k+2-t}^t$. It follows that

$$M_{k+1-t}^{t+1} = \big\{\{w_{k+1-t,j'} \mid j' \in I_i\} \mid i \in [\ell]\big\}.$$

Moreover, since $V_{k+1-t} \subseteq N(V_{k+2-t})$ and V_{k+2-t} is a union of elements of π_k^t, the column $V_{k+2-(t+1)} = V_{k+1-t}$ must be a union of elements of π_k^{t+1}. Hence,

$$\big\{\{w_{k+2-(t+1),j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\big\} \subseteq \pi_k^{t+1}. \tag{4.6}$$

Putting (4.2), (4.5), and (4.6) together, we get

$$\bigcup_{i' \in [t+1]} \big\{\{w_{k+2-i',j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\big\} \subseteq \pi_k^{t+1}. \tag{4.7}$$

Recall that by the induction hypothesis, it holds that $\big\{C \setminus (\bigcup_{i' \in [t]} V_{k+2-i'}) \,\big|\, C \in \pi_k^0, C \setminus (\bigcup_{i' \in [t]} V_{k+2-i'}) \neq \emptyset\big\} \subseteq \pi_k^t$. Thus, for every $C' \in \pi_k^t$ and for all vertices $u, v \in \bigcup_{i=0}^{k-t} V_i$ for which there is a $C \in \pi_k^0$ with $u, v \in C$, it holds that $|N(u) \cap C'| = |N(v) \cap C'|$. (To see this, recall that $N(u), N(v) \subseteq \bigcup_{i=0}^{k+1-t} V_i$.) This implies $\chi_{G_k,1}^{t+1}(u) = \chi_{G_k,1}^{t+1}(v)$. Together with (4.3), this yields that

$$\Big\{C \setminus \Big(\bigcup_{i' \in [t+1]} V_{k+2-i'}\Big) \,\Big|\, C \in \pi_k^0, C \setminus \Big(\bigcup_{i' \in [t+1]} V_{k+2-t}\Big) \neq \emptyset\Big\} \subseteq \pi_k^{t+1},$$

which, together with (4.7), concludes the proof of the claim. ⌟

We call the property described in the claim *path propagation from right to left*. The proof for the following statement, *path propagation from left to right*, is completely analogous. Therefore, we skip it.

Claim 2. Consider a colouring λ of G_k and its induced partition π_k of $V(G_k)$. For $t \in \mathbb{N}_0$, let π_k^t be the partition induced by $\chi_{G_k,1}^t$ on input (G_k, λ). Suppose G_k, λ, π_k satisfy the following conditions.

1. There exist $\ell \in [6]$ and $I_1, \dots, I_\ell \subseteq [6]$ such that $\bigcup_{i \in [\ell]} I_i = [6]$ and $I_i \cap I_j = \emptyset$ for $1 \leq i < j \leq \ell$ and for every $i \in [\ell]$, it holds that $\{w_{1,j'} \mid j' \in I_i\} \in \pi_k$. That is, the first column is a union of colour classes with respect to λ.
2. $\pi_k^1 = \big\{\{w_{2,j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\big\} \cup \big\{C \setminus V_2 \,\big|\, C \in \pi_k, C \setminus V_2 \neq \emptyset\big\}$.
3. For all $C, C' \subseteq V_2$ with $C, C' \in \pi_k^1$, the graph $G_k[C]$ is regular and $G_k[C, C']$ is biregular.

Then for every $t \in [k]$, it holds that

$$\begin{aligned}\pi_k^t = &\bigcup_{i' \in [t]} \{\{w_{i'+1,j'} \mid j' \in I_i\} \,\big|\, i \in [\ell]\} \\ &\cup \Big\{C \setminus \Big(\bigcup_{i' \in [t]} V_{i'+1}\Big) \,\Big|\, C \in \pi_k^0, C \setminus \Big(\bigcup_{i' \in [t]} V_{i'+1}\Big) \neq \emptyset\Big\}.\end{aligned} \qquad \lrcorner$$

We are now ready to analyse the run of Colour Refinement on input G_k. Recall that π_0^t denotes the partition induced by $\chi_{G_0,1}^t$ on $V(G_0) \subseteq V(G_k)$. For the following arguments, see also Figure 4.1.

In π_k^1, the vertices are distinguished according to their degrees. We can then use path propagation from right to left to deduce that

$$\begin{aligned}\pi_k^{k+1} &= \pi_0^1 \cup \big\{V_i \,\big|\, i \in [2, k+1]\big\} \quad \text{and thus} \quad \pi_k^{k+2} = \pi_0^2 \cup \big\{V_i \,\big|\, i \in [2, k+1]\big\}, \\ \pi_k^{k+3} &= \pi_0^3 \cup \big\{V_i \,\big|\, i \in [2, k+1]\big\} \quad \text{and also} \quad \pi_k^{k+4} = \pi_0^4 \cup \big\{V_i \,\big|\, i \in [2, k+1]\big\}.\end{aligned}$$

Now path propagation from left to right yields that

$$\pi_k^{2k+4} = \pi_0^4 \cup \big\{\{w_{i,j} \mid j \in [4]\} \,\big|\, i \in [2, k+1]\big\} \cup \big\{\{w_{i,j} \mid j \in \{5,6\}\} \,\big|\, i \in [2, k+1]\big\}$$

and $\pi_k^{2k+5} = \pi_0^5 \cup (\pi_k^{2k+4} \setminus \pi_0^4)$. Again using path propagation from right to left, we get that

$$\begin{aligned}\pi_k^{3k+6} = \pi_0^6 &\cup \big\{\{w_{i,j} \mid j \in \{1,2\}\} \,\big|\, i \in [2, k+1]\big\} \\ &\cup \big\{\{w_{i,j} \mid j \in \{3,4\}\} \,\big|\, i \in [2, k+1]\big\} \\ &\cup \big\{\{w_{i,j} \mid j \in \{5,6\}\} \,\big|\, i \in [2, k+1]\big\}\end{aligned}$$

and $\pi_k^{3k+7} = \pi_0^7 \cup (\pi_k^{3k+6} \setminus \pi_0^6)$. Similarly, we obtain

$$\begin{aligned}\pi_k^{4k+8} &= \pi_0^8 \cup \big\{\{w_{i,j} \mid j \in \{1,2\}\} \,\big|\, i \in [2, k+1]\big\} \\ &\qquad \cup \big\{\{w_{i,j} \mid j \in \{3,4\}\} \,\big|\, i \in [2, k+1]\big\} \\ &\qquad \cup \big\{\{w_{i,j}\} \,\big|\, i \in [2, k+1], j \in \{5,6\}\big\}, \\ \pi_k^{4k+9} &= \pi_0^9 \cup (\pi_k^{4k+8} \setminus \pi_0^8), \\ \pi_k^{5k+10} &= \pi_0^{10} \cup \big\{\{w_{i,j} \mid j \in \{1,2\}\} \,\big|\, i \in [2, k+1]\big\} \\ &\qquad \cup \big\{\{w_{i,j}\} \,\big|\, i \in [2, k+1], j \in [3,6]\big\}, \\ \pi_k^{5k+11} &= \pi_0^{11} \cup (\pi_k^{5k+10} \setminus \pi_0^{10}), \\ \pi_k^{5k+12} &= \pi_0^{12} \cup (\pi_k^{5k+11} \setminus \pi^{11}), \\ \pi_k^{6k+13} &= \pi_0^{13} \cup \big\{\{w_{i,j}\} \,\big|\, i \in [2, k+1], j \in [6]\big\},\end{aligned}$$

which is the discrete partition by the induction assumption for $k = 0$.

This implies that on input G_k, Colour Refinement takes $6k + 13$ iterations to stabilise and, since $|G_k| = 6k + 14$, it holds that $\mathrm{WL}_1(G_k) = n - 1$, where $n = |G_k|$.

In the third and fourth infinite family from the theorem, the three 0-pairs take on the role of V_{k+2}, the $(k + 2)$-nd column, which initiates the first path propagation. The proofs for those families are up to index changes essentially analogous to the one just presented.

In the fifth and sixth family, there are more 0-pairs. We sketch the splitting. In those families, in π^1, there is one partition class formed by all vertices contained in 0-pairs. The second partition class contains all other vertices. In π^2, the vertices contained in the two adjacent 0-pairs as well as the vertices contained in $\max(\prec)$ (i.e. the rightmost 0) form a separate partition class, while the class consisting of all vertices not contained in 0-pairs is not split. Now, like in the other families, those three 0-pairs take up the role of V_{k+2}, initiate the first path propagation and the proof proceeds similarly as above. □

The theorem immediately implies Theorem 4.1.

Corollary 4.14. *For every even* $n \in \mathbb{N}_{\geq 12}$ *such that* $n = 12$ *or* $n \bmod 18 \notin \{6, 12\}$, *there is a long-refinement graph* G *with* $|G| = n$. *The graph* G *can be chosen to satisfy* $\deg(G) = \{2, 3\}$.

Proof: The string representation S0011XX covers $n = 12$. The first infinite family from Theorem 4.13 covers all even $n \in \mathbb{N}_{\geq 14}$ with $n = 2 \mod 6$, i.e. with $n \bmod 18 \in \{2, 8, 14\}$. The second and the third infinite family both cover all even $n \in \mathbb{N}_{\geq 16}$ with $n = 4 \mod 6$, i.e. with $n \bmod 18 \in \{4, 10, 16\}$. The fourth and the fifth infinite family cover all even $n \in \mathbb{N}_{\geq 18}$ with $n = 0 \mod 18$. Thus, among the even sizes larger than 10, only the ones with $n \bmod 18 \in \{6, 12\}$ remain not covered. □

We now turn to the long-refinement graphs G of odd order with vertex degrees in $\{2, 3\}$. If the graph has odd size, we cannot represent it just with pairs. For this, we relax Assumption 4.9 as follows.

Assumption 4.15. *G is a long-refinement graph with* $\deg(G) = \{2, 3\}$ *and such that there is an* $i \in \mathbb{N}_0$ *for which* π^i *contains only pairs and at most one singleton.*

We maintain the vocabulary and notation from the long-refinement graphs of even order, i.e. 0, 1, S, X will be used in the same way as before. However, in order to fully describe the odd-size graphs via strings, we have to extend the string alphabet by fresh letters $\widehat{1}$ and $\widehat{\mathrm{X}}$, which represent particular pairs with attached vertices as follows. For a string $\widehat{\Xi}\colon [\ell] \to \{0, 1, \mathrm{S}, \mathrm{X}, \widehat{1}, \widehat{\mathrm{X}}\}$, we define the *base string* Ξ as the string obtained by removing hats, i.e. by replacing every $\widehat{1}$ with a 1 and every $\widehat{\mathrm{X}}$ with an X. Let $I(\widehat{\Xi}) \subseteq [\ell]$ be the set of positions i with $\widehat{\Xi}(i) \in \{\widehat{1}, \widehat{X}\}$. If in the base graph $G(\Xi)$, every vertex pair corresponding to a position in $I(\widehat{\Xi})$ (a *hat vertex pair*) is adjacent, we call $\widehat{\Xi}$ a *hat string.*

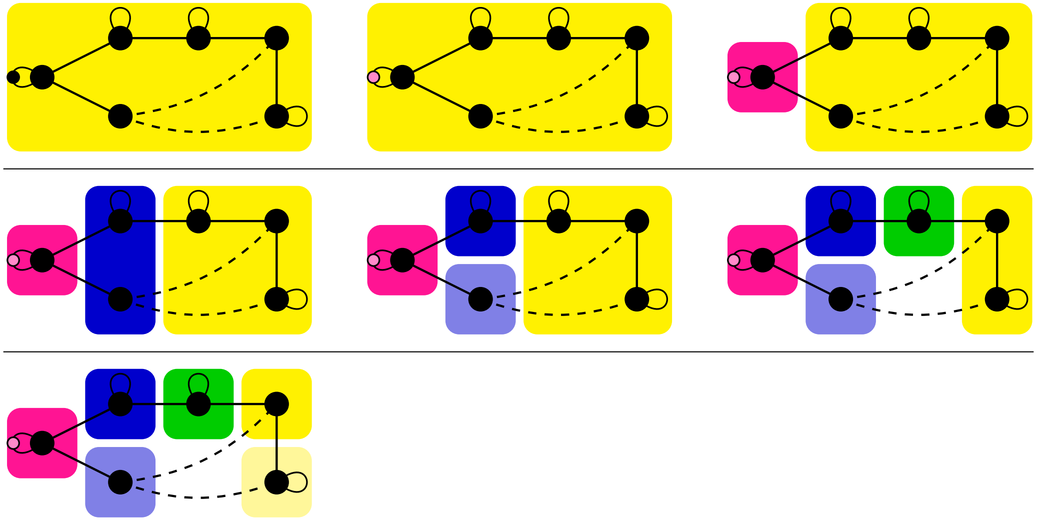


Figure 4.3: A visualisation of the graph with string representation $\mathrm{S}\widehat{1}11\mathrm{XX}$ and the evolution of the colour classes in the first 6 Colour Refinement iterations on the graph.

Similarly as for the even-size long-refinement graphs, to every hat string $\widehat{\Xi}$, we assign a graph $G(\widehat{\Xi})$. We obtain the graph $G(\widehat{\Xi})$ by subdividing in $G(\Xi)$ each edge connecting a hat vertex pair with a new fresh vertex, which we call a *hat*. For a hat $\widehat{v}$, we call the neighbourhood $N(\widehat{v}) \subseteq V\big(G(\widehat{\Xi})\big)$ the *hat base* of $\widehat{v}$. Note that every vertex in the hat base has degree 3 since it already has degree 3 in $G(\Xi)$ (cf. Notation 4.12). Also, a hat always has degree 2 and thus, with respect to $\chi^1_{G(\widehat{\Xi})}$, it has a different colour than its hat base.

Graphically, we represent a hat by a loop attached to the corresponding hat vertex pair, which we subdivide by inserting a small vertex that represents the hat (see Figure 4.3). It is not difficult to see that every graph $G(\widehat{\Xi})$ corresponding to a hat string $\widehat{\Xi}$ has exactly one hat and that the hat is the first vertex forming a singleton colour class during the execution of Colour Refinement on $G(\widehat{\Xi})$. Thus, $|G(\widehat{\Xi})| = |G(\Xi)| + 1$.

Theorem 4.16. *For every string $\widehat{\Xi}$ contained in the following sets, the graph $G := G(\widehat{\Xi})$ satisfies* $\mathrm{WL}_1(G) = |G| - 1$.

- $\{\mathrm{S}\widehat{1}11\mathrm{XX}\}$
- $\{\mathrm{S0X1}\widehat{\mathrm{X}}\} \cup \{\mathrm{S1}^k\mathrm{1011}^k\mathrm{X1X1}^k\widehat{1} \mid k \in \mathbb{N}_0\}$
- $\{\mathrm{S110X}\widehat{\mathrm{X}}\} \cup \{\mathrm{S111}^k\mathrm{1011}^k\mathrm{XX1}^k\widehat{1} \mid k \in \mathbb{N}_0\}$
- $\{\mathrm{S1}^k\mathrm{01}^k\mathrm{1XX1}^k\widehat{1} \mid k \in \mathbb{N}_0\}$
- $\{\mathrm{S(011)}^k\mathrm{00(110)}^k\mathrm{X}\widehat{1}\mathrm{X(011)}^k\mathrm{0} \mid k \in \mathbb{N}_0\}$

Proof sketch: The proof techniques for the infinite families are very similar to the ones presented for the families from Theorem 4.13. Therefore, we only sketch the proof on two concrete examples containing $\widehat{1}$ and $\widehat{X}$, respectively. To be able to refer to vertices explicitly, recall the formal definition of $G \coloneqq G(\Xi)$ from Section 4.2. Since for a hat string $\widehat{\Xi}$, it holds that $|G(\widehat{\Xi})| = |G(\Xi)| + 1$, we can use the same indexing of vertices, additionally letting $\widehat{v}$ be the unique hat in $G(\widehat{\Xi})$.

In the graph $G \coloneqq G(\mathrm{S\widehat{1}11XX})$ (cf. Figure 4.3), the only vertex of degree 2 is $\widehat{v}$. Thus, in π^1, it forms a singleton colour class. In π^2, the hat base $\{v_{2,1}, v_{2,2}\}$ forms a new colour class.

In general, for $i \in \mathbb{N}$, we have that $\pi^i_G = \pi^{i-1}_{G'} \cup \{\widehat{v}\}$, where $G' = G(\mathrm{S011XX})$ (cf. Theorem 4.13 and Figure 4.2).

Thus,

$$\begin{aligned}
\pi^1 &= \big\{\{\widehat{v}\}, V(G) \setminus \{\widehat{v}\}\big\},\\
\pi^2 &= \big\{\{\widehat{v}\}, \{v_{2,1}, v_{2,2}\}, V(G) \setminus \{\widehat{v}, v_{2,1}, v_{2,2}\}\big\},\\
\pi^3 &= \big\{\{\widehat{v}\}, \{v_{2,1}, v_{2,2}\}, \{v_{i,j} \mid i \in \{1,3\}, j \in [2]\}, \{v_{i,j} \mid i \in [4,6], j \in [2]\}\big\},\\
\pi^4 &= \big\{\{\widehat{v}\}, \{v_{1,1}, v_{1,2}\}, \{v_{2,1}, v_{2,2}\}, \{v_{3,1}, v_{3,2}\}, \{v_{i,j} \mid i \in [4,6], j \in [2]\}\big\},\\
\pi^5 &= \big\{\{\widehat{v}\}, \{v_{1,1}, v_{1,2}\}, \{v_{2,1}, v_{2,2}\}, \{v_{3,1}, v_{3,2}\}, \{v_{4,1}, v_{4,2}\}, \{v_{i,j} \mid i \in [5,6], j \in [2]\}\big\},\\
\pi^6 &= \big\{\{\widehat{v}\}\big\} \cup \big\{\{v_{i,j} \mid j \in [2]\} \mid i \in [6]\big\}.
\end{aligned}$$

Now in 6 further iterations, the splitting of the pairs is propagated linearly according to the order $\prec$.

Now let G be the graph $G(\mathrm{S0X1\widehat{X}})$, i.e. a member of the first infinite family. It has three vertices of degree 2, namely $\widehat{v}$, $v_{2,1}$, and $v_{2,2}$, which therefore form a colour class in π^1. Also, the vertices contained in the 1-pair are the only vertices of degree 3 that are not adjacent to any vertex of degree 2. Thus,

$$\pi^2 = \big\{\{\widehat{v}, v_{2,1}, v_{2,2}\}, \{v_{4,1}, v_{4,2}\}, V(G) \setminus \{\widehat{v}, v_{2,1}, v_{2,2}, v_{4,1}, v_{4,2}\}\big\},$$

and similarly,

$$\pi^3 = \big\{\{\widehat{v}, v_{2,1}, v_{2,2}\}, \{v_{4,1}, v_{4,2}\}, \{v_{1,1}, v_{1,2}\}, \{v_{i,j} \mid i \in \{3,5\}, j \in [2]\}\big\}.$$

Now the hat forms a singleton since, in contrast to the vertices of the 0-pair, it is not adjacent to any vertex in the S-pair. We obtain:

$$\begin{aligned}
\pi^4 &= \big\{\{\widehat{v}\}, \{v_{2,1}, v_{2,2}\}, \{v_{4,1}, v_{4,2}\}, \{v_{1,1}, v_{1,2}\}, \{v_{i,j} \mid i \in \{3,5\}, j \in [2]\}\big\},\\
\pi^5 &= \big\{\{\widehat{v}\}\big\} \cup \big\{\{v_{i,j} \mid j \in [2]\} \mid i \in [5]\big\}.
\end{aligned}$$

Then in 5 further iterations, the splitting of the pairs is propagated linearly according to the order $\prec$. □

With the odd-size families from the theorem, we can deduce the following.

Corollary 4.17. *For every odd $n \in \mathbb{N}_{\geq 11}$ with $n \bmod 18 \notin \{3, 9\}$, there is a long-refinement graph G with $|G| = n$. The graph G can be chosen to satisfy $\deg(G) = \{2, 3\}$.*

Proof: The string representation $\mathrm{S}\widehat{1}\mathrm{11XX}$ covers $n = 13$. The first infinite family covers all odd $n \in \mathbb{N}_{\geq 11}$ with $n = 5 \mod 6$, i.e. with $n \bmod 18 \in \{5, 11, 17\}$. The second and the third infinite family both cover all odd $n \in \mathbb{N}_{\geq 13}$ with $n = 1 \mod 6$, i.e. with $n \bmod 18 \in \{1, 7, 13\}$. The fourth infinite family covers all odd $n \in \mathbb{N}_{\geq 15}$ with $n = 15 \mod 18$. Thus, among the odd orders larger than 10, only the ones with $n \bmod 18 \in \{3, 9\}$ are not covered. □

We summarise the results from Corollaries 4.14 and 4.17.

Corollary 4.18. *For every $n \in \mathbb{N}_{\geq 11}$ such that $n = 12$ or $n \bmod 18 \notin \{3, 6, 9, 12\}$, there is a long-refinement graph G with $|G| = n$. The graph G can be chosen to satisfy $\deg(G) = \{2, 3\}$.*

The following lemma allows to cover more graph sizes.

Lemma 4.19. *Let $n \in \mathbb{N}$ be arbitrary. Suppose there is a long-refinement graph G such that $|G| = n$. If there is a $d \in \mathbb{N}$ with $\deg(G) = \{d, d + 1\}$ such that $|\{v \in V(G) \mid \deg(v) = d\}| \neq d + 1$, then there is also a long-refinement graph G' with $|G'| = n + 1$.*

Proof: We can insert an isolated vertex w into G and insert edges from w to every vertex $v \in V(G)$ with $\deg_G(v) = d$. In the new graph G', the vertex w has a degree other than $d + 1$, whereas all other vertices have degree $d + 1$. Thus, the colour classes in $\pi^1_{G'}$ are $\{w\}$ and $V(G') \setminus \{w\}$. After the second iteration, the neighbours of w are distinguished from all other vertices, just like they are in π^1_G. Inductively, it is easy to see that for $i \in \mathbb{N}$, it holds that $\pi^i_{G'} = \pi^{i-1}_G \cup \big\{\{w\}\big\}$. Thus, Colour Refinement takes $n - 1 + 1 = n$ iterations to stabilise on G'. □

We can thus cover all odd graph sizes larger than 10.

Corollary 4.20. *For every odd $n \in \mathbb{N}_{\geq 11}$, there is a long-refinement graph G with $|G| = n$.*

Proof: By Corollary 4.17, it suffices to provide long-refinement graphs of order n for every odd $n \in \mathbb{N}_{\geq 11}$ with $n \bmod 18 \in \{3, 9\}$. We will accomplish this by applying Lemma 4.19 to suitable graphs of orders n' with $n' \bmod 18 \in \{2, 8\}$. Every graph G with a string representation contained in one of the infinite families from Theorem 4.13 has an even number of vertices of degree 2. In particular, it satisfies $|\{v \in V(G) \mid \deg(v) = 2\}| \neq 3$. Furthermore, every even graph order larger than 10 excluded in Corollary 4.14 is a multiple of 6. Hence, since $N := \{n \in \mathbb{N}_{\geq 18} \mid n \bmod 18 \in \{2, 8\}\}$ contains only even numbers and no multiples of 6, for every $n' \in N$, there is a graph of order n' that satisfies the prerequisites of

Lemma 4.19 with $d = 2$. (Actually, we can cover all of these sizes with the family $\{\mathrm{S}1^k001^k\mathrm{X}1\mathrm{X}1^k0 \mid k \in \mathbb{N}_0\}$.)

Thus, applying the lemma, we can construct for every $n \in \mathbb{N}_{\geq 18}$ with $n \bmod 18 \in \{3, 9\}$ a graph G' with $\mathrm{WL}_1(G') = |G'| - 1$. □

Note that, since we apply Lemma 4.19 to close the gaps, we cannot guarantee anymore that the vertex degrees are 2 and 3, as we could in Corollary 4.18.

Corollary 4.21. *For every $n \in \{1\} \cup \mathbb{N}_{\geq 10}$ with $n = 12$ or $n \bmod 18 \notin \{6, 12\}$, there is a long-refinement graph G with $|G| = n$.*

Proof: This follows from combining Corollaries 4.14 and 4.20 with Theorem 4.2. □

Although the corollary leaves some gaps, we can deduce a new lower bound on the number of Colour Refinement iterations until stabilisation, which is optimal up to an additive constant of 1.

Theorem 4.22. *For every $n \in [3] \cup \mathbb{N}_{\geq 10}$, it holds that $\mathrm{WL}_1(n) \in \{n-2, n-1\}$.*

Proof: The statement trivially holds for $n \leq 2$. For $n = 3$, one can take a path of length 2. For $n \geq 10$, by Corollary 4.21, we only need to consider the numbers $n \geq 24$ for which $n \bmod 18 \in \{6, 12\}$. Since these numbers are all even, for every such n, we can use Corollary 4.20 to obtain a graph G' with $|G'| = n - 1$ and $\mathrm{WL}_1(G') = |G'| - 1$. Let v be a fresh vertex, not contained in $V(G')$. Now define $G := (V, E)$ with $V := V(G') \cup \{v\}$ and $E := E(G')$. Then $\pi^i_G = \pi^i_{G'} \cup \{\{v\}\}$ for every $i \in \mathbb{N}$. In particular, $\mathrm{WL}_1(G) = \mathrm{WL}_1(G') = n - 2$. □

4.4 Discussion

With Theorem 4.22, it holds for all $n \in \mathbb{N}_{\geq 10}$ that $\mathrm{WL}_1(n) \geq n - 2$. In particular, this proves that the trivial upper bound $\mathrm{WL}_1(n) = n - 1$ is tight, up to an additive constant of 1.

For infinitely many graph sizes, the graph G can even be chosen to have vertex degrees 2 and 3, as Theorems 4.13 and 4.16 show. We applied Lemma 4.19 to cover some of the remaining sizes. However, no order $n \in \mathbb{N}_{\geq 18}$ with $n \bmod 18 \in \{6, 12\}$ is covered by Theorem 4.13. Also, for $|G| \in \{n \in \mathbb{N}_{\geq 18} \mid n \bmod 18 \in \{5, 11\}\}$, the only long-refinement graphs G we have found are the members of the first infinite family in Theorem 4.16. For every such odd-size graph G, it holds that $|\{v \in V(G) \mid \deg(v) = 2\}| = 3$. Thus, G does not satisfy the prerequisites of Lemma 4.19. Note that we cannot apply the construction from the proof if $|\{v \in V(G) \mid \deg(v) = d\}| = d + 1$, since the new graph G' would be $(d+1)$-regular and would thus satisfy $\mathrm{WL}_1(G') = 0$. Hence, it is not clear how to apply our techniques to construct a graph of order 24 with 23 Colour Refinement iterations. Altogether, the values $n \in \mathbb{N}_{\geq 24}$ with $n \bmod 18 \in \{6, 12\}$ are precisely the graph orders for which it remains open whether there is a graph G with $\mathrm{WL}_1(G) = |G| - 1$.

Also, in the light of the graph isomorphism problem, it is a natural follow-up task to find for each order n pairs of non-isomorphic graphs G, H for which it takes $n - 1$ Colour Refinement iterations to distinguish the graphs from each other. A first step towards this goal is the search for pairs of long-refinement graphs of equal order. It is easy to see that for infinitely many n, Theorems 4.13 and 4.16 yield such pairs of graphs. Still, for example, when evaluating the colourings computed by Colour Refinement on the graphs with string representations S1100XX0 and S001XX10, they differ after less than $n - 1$ iterations. (To see this, observe that in $G(\mathrm{S1100XX0})$, all vertices of degree 2 have paths of length 3 to a vertex of degree 2 whose inner vertices only have degrees other than 2. This is not the case for $G(\mathrm{S001XX10})$ and this property is detected by Colour Refinement after at most 4 iterations, which can be shown using the correspondence to pebble games from Theorem 3.18.) Thus, finding for $n \in \mathbb{N}_{\geq 10}$ two graphs of order n which Colour Refinement only distinguishes after $n - 1$ iterations remains a challenge.

5 Upper Bounds on the Iteration Number for 2-WL

We have seen that there are infinitely many graphs on which the 1-dimensional Weisfeiler-Leman algorithm takes $n-1$ iterations to stabilise. Now it is natural to ask whether the trivial upper bound $\mathrm{WL}_k(n) \leq n^k - 1$ on the number of iterations of the k-dimensional Weisfeiler-Leman algorithm on graphs with n vertices is tight also for $k \geq 2$.

On the one hand, using graphs of bounded colour class size, Fürer shows that $\mathrm{WL}_k(n) \in \Omega(n)$, which is still the best known lower bound for general k [48]. On the other hand, building on work from [85], Berkholz and Nordström present pairs of n-element structures that are not C^k-equivalent and thus distinguished by the $(k-1)$-dimensional Weisfeiler-Leman algorithm, but for which each distinguishing C^k-formula has a quantifier depth in $n^{\Omega(k/\log k)}$ [21]. This implies by Theorem 3.18 the near-optimal lower bound of $n^{\Omega(k/\log k)}$ on the number of iterations that the k-dimensional Weisfeiler-Leman algorithm needs to distinguish the two structures from each other. Although the result does not yield a new lower bound when the structures are required to be graphs, it can be regarded as an indication that the bound $\mathrm{WL}_k(n) \leq n^k - 1$ might in fact be tight.

In this chapter, we prove that this is not the case for $k = 2$. More precisely, we show that on every graph with n vertices, the 2-dimensional Weisfeiler-Leman algorithm terminates after $O(n^2/\log n)$ iterations (see Theorem 5.19).

For this, we define a game in which two players alternate in their turns on the input graph. The game is designed such that it mimics some of the mechanisms of the 2-dimensional Weisfeiler-Leman algorithm. Player Maxi assumes the role of an adversary. Being allowed arbitrary refinements, Maxi is even more powerful than the Weisfeiler-Leman algorithm. After each of Maxi's turns, the second player, Mini, "tidies" the colouring of the graph in so-called clean-up steps. We show that the costs of this game form an upper bound on the iteration number of the 2-dimensional Weisfeiler-Leman algorithm. Therefore, to obtain our bound on the iteration number, it suffices to show that the costs of the game, assuming the players play optimally, are in $O(n^2/\log n)$.

The game allows us to consider the dynamics of the 2-dimensional Weisfeiler-Leman algorithm from a sequential point of view. Since one turn in the game consists of elementary actions which can be performed one after the other and not all at the same time as happens in one iteration of the Weisfeiler-Leman algorithm, the costs of the game are easier to analyse.

When choosing her actions, Mini distinguishes between large and small vertex colour classes in the graph with the current colouring. This way, we can show that if the size of each colour class in the coloured input graph is bounded by a constant, then the number of iterations of the 2-dimensional Weisfeiler-Leman algorithm on the graph is linear in the graph order (see Lemma 5.18). In fact, this yields the first improvement on the trivial upper bound of $n^2 - 1$ for graphs of bounded colour class size.

In this chapter, it is convenient for us to imagine the colourings computed by the 2-dimensional Weisfeiler-Leman algorithm as colours of edges in a directed graph. More precisely, we consider the intermediate results obtained after iterations of the algorithm as coloured directed graphs. In order for this view to be consistent with our definition of coloured directed graphs from Section 2.3, all edges must be present in the graphs (since colourings in coloured graphs are defined only on the arcs stemming from single vertices or edges). Thus, all graphs G we consider in this chapter are coloured directed complete graphs, i.e. they satisfy $E(G) = (V(G))^2$. Note that this is not a restriction because we can reserve a specific colour for non-edges and in this way model every coloured graph as a complete coloured graph. The graph colourings will mostly be clear from the context (or irrelevant) and in that case, we do not mention them explicitly.

The results presented in this chapter were published in [102] and [103].

5.1 A Game for a Sequential Perspective

For two coloured graphs (G, λ) and (G', λ') with $G = (V, E)$ and $G' = (V', E')$ and corresponding partitions $\pi(\lambda)$ and $\pi(\lambda')$, we say that G' *refines* G (and, equivalently, that G' is a *refinement* of G) if $V = V'$ and $\pi(\lambda) \succeq \pi(\lambda')$. Slightly abusing notation, we write $G \succeq G'$ in this case. If both $G \succeq G'$ and $G' \succeq G$ hold, we write $G \equiv G'$ and call the two graphs *equivalent*. Similarly, we say that a player *refines* their input graph (G, λ) if they recolour the edges of G in such a way that the new induced partition of $(V(G))^2$ is finer than the partition induced by λ.

For a graph G, let $G^{(i)}$ be the edge-coloured graph obtained from G after i iterations of the 2-dimensional Weisfeiler-Leman algorithm and let $\widetilde{G}$ be the graph that the algorithm produces on input G. That is, the graph $\widetilde{G}$ equals $G^{(j)}$, where j is the smallest integer for which $G^{(j)} \equiv G^{(j+1)}$. We call G and the induced partition of the 2-tuples of its vertices *stable* if $\widetilde{G} \equiv G^{(0)}$. Accordingly, we call $\widetilde{G}$ the *stabilisation* of G.

In the following, we describe the game that enables us to take a sequential perspective on the 2-dimensional Weisfeiler-Leman algorithm in detail. The input of the game is a coloured graph. On this graph, the two players Maxi and Mini alternate turns and Maxi plays first. Each turn consists in choosing a refinement of the current colouring of the graph. If in his turn, Maxi is faced with the coloured graph G, then he must choose a proper refinement of G, that is, he must return a graph G' with $G \succneqq G'$. The coloured graph that Maxi chooses is the input for Mini's next turn.

Faced with a graph G, Mini has to return a graph G' with $G \succeq G' \succeq \widetilde{G}$. Thus, the refinement that Mini chooses must still be at least as coarse as $\widetilde{G}$, but it may be equivalent to G or to $\widetilde{G}$.

The game ends when the colouring of the graph induces the discrete partition, i.e. the partition $\{\{(u,v)\} \mid u,v \in V(G)\}$. Note that the discrete partition is not necessarily the coarsest stable partition, i.e. the final graph of the game starting with an initial graph G may be a proper refinement of the output of the 2-dimensional Weisfeiler-Leman algorithm on input G.

Each turn of Maxi adds 1 to the total cost of the game, whereas the cost for a turn of Mini playing G' as a response to G is the smallest integer j such that $G' \succeq G^{(j)}$. Thus, a turn of Mini costs the number of iterations of the 2-dimensional Weisfeiler-Leman she needs to refine the graph she returns to Maxi. In the game, every modification of a graph that increases the total costs by 1 is called a *move*. Note that the costs of the game are in $O(n^2)$.

As the names indicate, Maxi intends to maximise the total cost whereas his opponent Mini aims at minimising it. Given a strategy S_1 of Maxi and a strategy S_2 of Mini, let $\mathrm{val}(S_1, S_2)$ denote the costs of the game when Maxi uses S_1 and Mini uses S_2. Since the game is finite and deterministic and has perfect information, it is not hard to see that the value $c(G)$ of the game (i. e., the costs assuming both players play optimally) is well-defined and that

$$\max_{S_1} \min_{S_2} \mathrm{val}(S_1, S_2) = c(G) = \min_{S_2} \max_{S_1} \mathrm{val}(S_1, S_2).$$

Note that we can also view the game as a zero-sum game (when considering the costs as the gain for Maxi and the loss for Mini).

The following lemma states an important monotonicity fact for our game.

Lemma 5.1. *Let G and H be two coloured graphs with $G \succeq H$. Then the following hold.*

1. $G^{(i)} \succeq H^{(i)}$ *for every* $i \in \mathbb{N}_0$.
2. $\widetilde{G} \succeq \widetilde{H}$.
3. *If additionally* $H \succeq \widetilde{G}$, *then for every* i *with* $G^{(i)} = \widetilde{G}$, *we have that* $H^{(i)} \equiv \widetilde{G}$. *In particular* $\widetilde{H} \equiv \widetilde{G}$.

Proof: Part 1 follows straight from the definition of the Weisfeiler-Leman algorithm using induction on i: the induction basis $i = 0$ simply restates the assumption that $G \succeq H$. The induction step follows from a combination of the induction assumption and the fact that the colouring computed by the 2-dimensional Weisfeiler-Leman algorithm in iteration $i + 1$ refines the colouring from iteration i. Indeed, suppose that $G^{(i)} \succeq H^{(i)}$ for a fixed i. Let χ_G^i and χ_H^i denote the colourings in $G^{(i)}$ and $H^{(i)}$, respectively, and let χ_G^{i+1} and χ_H^{i+1} denote the colourings in $G^{(i+1)}$ and $H^{(i+1)}$. Let $V := V(G) = V(H)$. Let (u_1, u_2) and (v_1, v_2) be two tuples with $u_1, u_2, v_1, v_2 \in V$ and such that $\chi_G^{i+1}(u_1, u_2) \neq \chi_G^{i+1}(v_1, v_2)$. We show that $\chi_H^{i+1}(u_1, u_2) \neq \chi_H^{i+1}(v_1, v_2)$.

If $\chi_G^i(u_1, u_2) \neq \chi_G^i(v_1, v_2)$, the statement follows from $G^{(i)} \succeq H^{(i)}$ and $\pi(\chi_H^i) \succeq \pi(\chi_H^{i+1})$. Otherwise $\chi_G^i(u_1, u_2) = \chi_G^i(v_1, v_2)$ and therefore, it must hold that

$$\{\!\!\{ \big(\chi_G^i(w, u_2), \chi_G^i(u_1, w)\big) \mid w \in V \}\!\!\} \neq \{\!\!\{ \big(\chi_G^i(w, v_2), \chi_G^i(v_1, w)\big) \mid w \in V \}\!\!\}.$$

However, since $G^{(i)} \succeq H^{(i)}$, the same inequality holds when substituting χ_H^i for χ_G^i. Thus, the tuples $\chi_H^{i+1}(u_1, u_2)$ and $\chi_H^{i+1}(v_1, v_2)$ differ in their second components and hence, they are distinct.

Therefore, if i is the minimal integer with $G^{(i)} = \widetilde{G}$, then it holds that $\widetilde{G} \succeq H^{(i)} \succeq \widetilde{H}$, which proves Part 2. Part 3 is an immediate consequence of Part 1, using the relation $G^{(i)} \succeq H^{(i)} \succeq \widetilde{G}$. □

Hence, the game is monotone in the following sense: given an input graph H, the costs for Mini to obtain the stabilisation $\widetilde{H}$ are at most as high as the costs for her to obtain the stabilisation of any other input graph G with $G \succeq H$ and $\widetilde{G} \equiv \widetilde{H}$.

We first show that it is never optimal for Mini to execute partial iterations of the 2-dimensional Weisfeiler-Leman algorithm on her input.

Lemma 5.2. *Suppose that in one of her turns, Mini is given a coloured graph G. If she plays optimally, she returns $G^{(i)}$ for some integer i.*

Proof: Consider a strategy S for Mini which dictates to her for at least one input G for her turn to play a graph G' such that $G' \not\equiv G^{(j)}$ for every j. In the game tree T with this fixed strategy for Mini, consider a counterexample situation that is maximal with respect to the number of moves performed so far. That is, consider a node $x \in V(T)$ at which, on input G, Mini finishes her turn with a graph G' such that $G' \not\equiv G^{(j)}$ for every j and for which the subtree of T rooted at x contains no such node. Thus, in her future turns, Mini always returns a graph corresponding to full iterations of the 2-dimensional Weisfeiler-Leman algorithm.

Let i be the maximal j with $G^{(j)} \succneqq G'$. Then Maxi can define $G^{(i+1)}$ as the new input for Mini. By the maximality of the counterexample, Mini now finishes this turn with a graph $G^{(i+\ell)}$ with $\ell \geq 1$. This conversion of G into $G^{(i+\ell)}$ contributes the costs $(i + 1) + 1 + \big(i + \ell - (i + 1)\big) = i + \ell + 1$, whereas Mini could have reached $G^{(i+\ell)}$ with $i+\ell$ moves by simply executing the $i+\ell$ necessary iterations. Thus, the strategy S is not optimal. Hence, in every optimal strategy, Mini only returns graphs corresponding to full iterations of the 2-dimensional Weisfeiler-Leman algorithm. □

It is even optimal for Mini to return the stabilisation of her input.

Corollary 5.3. *Given a coloured graph G, it is optimal for Mini to return $\widetilde{G}$.*

Proof: Similarly to the proof of Lemma 5.2, consider the counterexamples in which Mini is given a graph G and finishes her turn with a graph $G^{(i)} \succneqq \widetilde{G}$ and it would be worse for her to return $\widetilde{G}$ instead. Take such a counterexample situation that is maximal with respect to the number of moves performed so far. Given $G^{(i)}$, Maxi can define $G^{(i+1)}$ as the new input for Mini (including the case that $i+1=k$). The conversion costs i for the moves of Mini and 1 for Maxi, so altogether $i+1$. By the maximality of the counterexample, from now on, we can assume that Mini always returns the stabilisation of her input. If $i+1=k$, that is $G^{(i+1)} = \widetilde{G}$, the costs are the same as if Mini had chosen to stabilise G herself. If $i+1<k$, Mini now needs $k-(i+1)$ additional moves to stabilise $G^{(i+1)}$, because $\widetilde{G^{(i+1)}} = \widetilde{G}$ by Lemma 5.1. So the conversion costs are $i+1+\big(k-(i+1)\big)=k$. That is, in all cases, Maxi can choose to pursue a strategy such that the costs are at least as high as when Mini always returns stable graphs. □

To see that the total costs of the 2-player game yield an upper bound on the number $\mathrm{WL}_2(G)$ of iterations of the 2-dimensional Weisfeiler-Leman algorithm on input G, it suffices to describe a strategy for Maxi with which the game has costs at least $\mathrm{WL}_2(G)$. For this, Maxi can simply recolour the edges according to the first iteration of the Weisfeiler-Leman algorithm and define the graph $G^{(1)}$ as the first input for Mini. Corollary 5.3 states that it is optimal for Mini to perform the entire Weisfeiler-Leman algorithm on her input. Thus, she could not do better but to perform the remaining iterations of the algorithm on $G^{(1)}$, resulting in costs that are at least as high as $\mathrm{WL}_2(G)$. Note that they could be higher because the game does not end before the discrete partition is reached.

We therefore obtain the following corollary, which allows us to analyse the costs of the described game in order to obtain an upper bound on the number of iterations of the 2-dimensional Weisfeiler-Leman algorithm.

Corollary 5.4. Let G be a graph and let $c(G)$ be the value of the described 2-player game on input G. Then $\mathrm{WL}_2(G) \leq c(G)$.

As a consequence of Corollary 5.3, we can draw conclusions about optimal strategies for Maxi.

Lemma 5.5. *If playing G' in response to G is optimal for Maxi, then playing G'' with $G \succneqq G'' \succeq G'$ is also optimal for Maxi.*

Proof: Given a coloured input graph G, a turn of Maxi consists in choosing a set $\mathcal{C}'$ of colour classes in G that he wants to refine. We denote by $G_{\mathcal{C}'}$ the graph that Maxi returns to Mini.

We show that it is optimal for Maxi to refine only one colour class per turn. Suppose Maxi is given a graph G and decides to refine a set $\mathcal{C}'$ consisting of ℓ colour classes, i.e. $\mathcal{C}' = \{C_i \mid 1 \leq i \leq \ell\}$ with $\ell > 1$.

Let k be the minimal number with $\widetilde{G}_{\mathcal{C}'} = G_{\mathcal{C}'}^{(k)}$. Due to Corollary 5.3, we can assume Mini always returns the stabilisation of her input. Hence, the costs for Mini to return $\widetilde{G}_{\mathcal{C}'}$ are k. We show by induction that the costs do not decrease when Maxi only refines one colour class per turn.

Suppose Maxi plays the graph G_{C_1}, which only refines C_1. Mini will then stabilise G_{C_1} with, say, j iterations. That is, she returns $\widetilde{G}_{C_1} = G_{C_1}^{(j)}$. From Part 1 of Lemma 5.1, we obtain that $\widetilde{G}_{C_1} \succeq G_{\mathcal{C}'}^{(j)}$. If $j \geq k$, Maxi can refine $\widetilde{G}_{C_1}$ to the graph $\widetilde{G}_{\mathcal{C}'}$ and continue as if $\widetilde{G}_{\mathcal{C}'}$ was his input graph for this round.

Assume therefore that $j < k$. We know that

$$\widetilde{G}_{C_1} = G_{C_1}^{(j)} \succeq G_{\mathcal{C}'}^{(j)} \succeq G_{\mathcal{C}'}^{(k)} = \widetilde{G}_{\mathcal{C}'}.$$

The second "$\succeq$" is proper since $j < k$ and k is minimal. From Part 3 of Lemma 5.1, we know that the stabilisation of $G_{\mathcal{C}'}^{(j)}$ is $\widetilde{G}_{\mathcal{C}'}$. This implies that the first "$\succeq$" is also proper because $\widetilde{G}_{C_1}$ is stable, whereas $G_{\mathcal{C}'}^{(j)}$ is not.

This shows that $\widetilde{G}_{C_1}$ is strictly coarser than $G_{\mathcal{C}'}^{(j)}$. Thus, Maxi can proceed by refining $\widetilde{G}_{C_1}$ to $G_{\mathcal{C}'}^{(j)}$, on which Mini needs $k - j$ moves to obtain the stabilisation.

In a similar way, one can show that it is optimal for Maxi to split the selected colour class into only two colour classes. □

From now on, we can thus assume that Maxi only refines one colour class per turn and that he splits this class into exactly two new colour classes. What is the effect of such a turn of Maxi? By recolouring loops and edges, Maxi refines vertex colour classes and edge colour classes. As a consequence, some regularity conditions may no longer hold in the obtained graph G. We are in particular interested in the following two conditions.

(C1) The colour of an edge determines the colour of its head and of its tail. More precisely, for all $u, u', v, v' \in V(G)$, if (u, v) and (u', v') have the same colour, then (u, u) and (u', u') have equal colours and similarly, (v, v) and (v', v') have equal colours as well.

(C2) For every edge colour class C and every two vertex colour classes C_1 and C_2, the graph induced by C between C_1 and C_2 is biregular. This means that for every two vertices $v_1, v_2 \in C_1$, we have $|N_C^+(v_1) \cap C_2| = |N_C^+(v_2) \cap C_2|$

and $|N_C^-(v_1) \cap C_2| = |N_C^-(v_2) \cap C_2|$ and similarly for the C-neighbourhoods in C_1 of vertices in C_2.

In her reaction to a turn of Maxi, we let Mini "clean up" the graph: By refining edge colour classes, first she reestablishes Condition C1 and, after that, Condition C2, i. e., the colour-biregularity. We claim that the costs of this procedure only amount to 2. Indeed, to satisfy Condition C1, Mini can simply perform one iteration of the Weisfeiler-Leman algorithm on her input and ignore various refinements that would be made in this iteration. More precisely, given G coloured with λ, she can return the colouring that assigns to (u, v) the colour

$$\big(\lambda(u, v), \lambda(v, v), \lambda(u, u)\big).$$

Thus, in the first move of her turn, she only refines (real) edge colour classes whose heads (and tails, respectively) are not all of the same colour. In the second move, she reestablishes colour regularity, i. e., she refines vertex colour classes according to their colour degrees. We call these two moves a *clean-up step.*

It is possible that the attempt to establish Condition C2 causes Condition C1 to be violated again. Indeed, if the move splits vertex colour classes, Condition C1 might need to be reestablished afterwards due to the splitting, i. e., Mini might need to continue cleaning up. In turn, this might cause Condition C2 to be violated, and so on. We call a shortest succession of clean-up steps that reestablishes both Condition C1 and Condition C2 simultaneously a *complete clean-up step.* We denote by clean(G) the coloured graph resulting from a complete clean-up step on G.

Recall that our goal is to show that the costs of the game are bounded by $O(n^2/\log n)$. Therefore, the following observation allows Mini to always perform complete clean-up steps without causing critical extra costs.

Observation 5.6. There are only $O(n)$ splittings of vertex colour classes in the game.

A clean-up step that is not yet a complete clean-up step consists of at least one splitting of a vertex colour class. Therefore, the total costs for a complete clean-up step can be bounded by $2s + 2 = O(s)$, where s is the number of vertex colour class splittings appearing in the complete clean-up step. (The additional 2 is for the costs of the last clean-up step, which does not need to incur a vertex colour class splitting.) By Observation 5.6, there are only $O(n)$ vertex colour class splittings in the whole game and each of them appears in at most one complete clean-up step. We obtain the following bound on the costs for clean-up steps that split vertex colour classes.

Corollary 5.7. *The total costs for complete clean-up steps that split vertex colour classes amount to $O(n)$.*

A complete clean-up step in which no vertex colour classes are split consists only of one clean-up step. Therefore, such a complete clean-up step has constant costs of at most 2. This means that we can assign these costs to the preceding move (either performed by Maxi or Mini) since every move entails at most one complete clean-up step. Thus, the extra costs for complete clean-up steps do not have an effect on the asymptotic bound that we want to obtain for the costs of the game.

Now we know optimal strategies for both players and we have seen that Mini can perform complete clean-up steps whenever she wants without asymptotically increasing the costs. We can thus assume that Mini always performs a complete clean-up step on her input graph before and after manipulating it.

5.1.1 Large Colour Classes

To give a bound on the costs of the game, we distinguish between large vertex colour classes and small vertex colour classes with respect to some threshold function t.

Let $t\colon \mathbb{N} \to \mathbb{N}$ be a function. We call a vertex colour class of G *large* with respect to t if it consists of at least $t(|V(G)|)$ vertices. A *small* vertex colour class is one that is not large.

The choice of the threshold function t for our purposes will be a tradeoff between the costs for operations on edges incident with large vertex colour classes and edges incident with small vertex colour classes. We will later see that setting $t(n) := \log_2(n)/2$ suffices to prove our main result (see Theorem 5.19).

We first treat edges that are incident with large vertex colour classes. Due to Observation 5.6 and Corollary 5.7, we only have to consider moves in which no vertex colour classes are split and which are not part of clean-up steps.

For a colouring λ, we define a potential function by setting

$$f(\lambda) := \sum_{v \in V(G)} \bigl|\bigl\{\lambda(v,w) \bigm| w \in V(G)\bigr\}\bigr|.$$

Obviously, $f(\lambda) \leq n^2$ for all colourings λ, and f is strictly monotonically increasing. That is, if $\pi(\lambda) \gneq \pi(\lambda')$, then $f(\lambda) < f(\lambda')$.

Let $B_1, B_2, \dots$ be an enumeration of the large vertex colour classes and let $B := \dot{\bigcup}_k B_k$ be the set of vertices in large vertex colour classes.

Lemma 5.8. *Suppose that the current colouring in the game is λ and that the current player chooses a refinement which, for some $v \in B$, induces a strictly finer partition on $\{(v,w) \mid w \in V\}$ than λ. If a subsequent complete clean-up step does not split any large vertex colour class, then f increases by at least $t(n)$.*

Proof: Let $v \in B$ be as in the assumptions of the lemma and let B_k be the large vertex colour class that contains v. Denote by λ' the colouring obtained after a subsequent clean-up step. Then, in particular, there is a certain, not necessarily large, vertex colour class C such that on the set $\{(v,w) \mid w \in C\}$, the colouring λ' induces a strictly finer partition than λ. Thus, the number of colour classes occurring in the set $\{(v,w) \mid w \in C\}$ increases by at least 1.

By assumption, the vertex colour class B_k is not split in this round and we can assume that Mini always performs complete clean-up steps. Therefore, to ensure that all vertices in B_k have identical colour degrees (Condition C2), also on every set $\{(v',w) \mid w \in C\}$ with $v' \in B_k$, the colouring λ' must induce a partition strictly finer than the one induced by λ. Hence, for each of these sets, of which there are at least $t(n) = t(|G|)$, the number of occurring colour classes increases by at least 1.

Let $V := V(G)$. For every vertex $v \in V$ and every vertex set V', we have

$$|\{\lambda'(v,w) \mid w \in V'\}| \geq |\{\lambda(v,w) \mid w \in V'\}|$$

and so

$$\begin{aligned}
f(\lambda') &= \sum_{v \in V} |\{\lambda'(v,w) \mid w \in V\}| \\
&= \sum_{v \in B_k} |\{\lambda'(v,w) \mid w \in V\}| + \sum_{v \in V \setminus B_k} |\{\lambda'(v,w) \mid w \in V\}| \\
&\overset{\text{C1}}{=} \sum_{v \in B_k} |\{\chi'(v,w) \mid w \in C\}| + \sum_{v \in B_k} |\{\lambda'(v,w) \mid w \in V \setminus C\}| + \\
&\quad \sum_{v \in V \setminus B_k} |\{\lambda'(v,w) \mid w \in V\}| \\
&\geq \sum_{v \in B_k} |\{\lambda(v,w) \mid w \in C\}| + t(n) + \sum_{v \in B_k} |\{\lambda(v,w) \mid w \in V \setminus C\}| + \\
&\quad \sum_{v \in V \setminus B_k} |\{\lambda(v,w) \mid w \in V\}| \\
&\geq t(n) + f(\lambda),
\end{aligned}$$

and this gives the claim. □

Corollary 5.9. The costs for the moves (and the following complete clean-up steps) in which the colour set of edges incident with large vertex colour classes is properly refined are in $O\big(n^2/t(n)\big)$.

This bound already includes the necessary complete clean-up steps. In the following, we treat the moves in which edge colour classes that are only incident with small vertex colour classes are refined.

5.1.2 Small Colour Classes

To analyse the costs for small vertex colour classes, we describe a strategy for Mini on them. For this, we define auxiliary graphs which she uses to derive moves in the original game. For a coloured graph G from the original game, its auxiliary graph is denoted by $\mathrm{Aux}(G)$. Just like with the clean-up steps in the original game, Mini will pursue a strategy according to information obtained from auxiliary graphs in order to maintain certain invariants.

Notation 5.10. *We describe all graphs coming from the original game using the letter G and all auxiliary graphs using the letter H or, when derived from a specific graph G, by* $\mathrm{Aux}(G)$.

Every auxiliary graph is undirected, simple, uncoloured, and not necessarily complete. Whereas in the original game, a move *recolours* edges, in the auxiliary graphs edges are *inserted.*

Next we describe how to obtain the auxiliary graph for a graph G appearing during a run of the game. Let $G_1, G_2, \dots, G_\ell = G$ be the graphs that have been played by the two players so far. Let $\mathcal{T}$ be the collection of vertex sets $C \subseteq V(G)$ that each form a small vertex colour class in some G_i. The graph $\mathrm{Aux}(G)$ is constructed as follows.

The vertices of $\mathrm{Aux}(G)$ form a partition into two sets called the *upper* and the *lower vertices* (V_u and V_ℓ, respectively). They are two identical copies of the set defined as

$$\big\{(C, M) \mid C \in \mathcal{T}, M \subseteq C\big\},$$

that is, the set that contains all pairs of a small vertex colour class C that has appeared so far in the game and a subset of C. Thus, as the game progresses, the vertex sets of the auxiliary graphs grow.

For every small vertex colour class $C \in \mathcal{T}$, the graph $\mathrm{Aux}(G)$ contains $2^{|C|}$ upper vertices and also $2^{|C|}$ lower vertices. Thus, letting $n := |G|$, there are at most $4n \cdot 2^{t(n)}$ vertices in $\mathrm{Aux}(G)$. (To see this, one observes that there are at most $2n$ elements in $\mathcal{T}$.) We have an undirected edge between an upper vertex $(C, M) \in V_u$ and a lower vertex $(D, N) \in V_\ell$ if there exists a set of colours $\mathcal{C}'$ such that, in G, every vertex $v \in C$ satisfies

$$v \in M \iff N^+_{\mathcal{C}'}(v) = N.$$

Another way of formulating this is that the vertices of M are exactly the ones whose $\mathcal{C}'$-neighbourhood is N.

Between two upper vertices (C, M) and (C', M'), we insert an undirected edge if such a condition holds in both directions—more precisely, if there are sets of colours $\mathcal{C}', \mathcal{C}''$ such that every vertex $v \in C'$ satisfies

$$v \in M \iff N^+_{\mathcal{C}'}(v) = M',$$

and every vertex $v \in C''$ satisfies

$$v \in M' \iff N^{+}_{C''}(v) = M.$$

The resulting object is the auxiliary graph $\mathrm{Aux}(G)$.

Remark 5.11. Let G and G' with $G \succeq G'$ be two graphs appearing during the original game (i. e., the graph G' appears later than G). Then it holds that $\mathrm{Aux}(G') \supseteq \mathrm{Aux}(G)$, since by definition, the graph $\mathrm{Aux}(G')$ contains all the vertices and all the edges from previous auxiliary graphs.

To describe the strategy that Mini will pursue in the original game, we need the notion of a *triangle completion*, an operation on undirected uncoloured graphs, which we define next. Whereas the clean-up steps are designed to simulate information essentially collected by the 1-dimensional Weisfeiler-Leman algorithm, the triangle completion is designed to capture certain dynamics of the 2-dimensional algorithm, as will become apparent.

For an undirected uncoloured graph H with $V(H) = V_u \mathbin{\dot\cup} V_\ell$, its *triangle completion* $\triangle(H)$ is obtained by applying the following rules once.

- Insert an edge between every two upper vertices that have a common neighbour in V_ℓ or in V_u.
- Insert an edge between every upper vertex and every lower vertex that have a common neighbour in V_u.

Note that no edges between any vertices in V_ℓ are inserted. On the obtained graph $\triangle(H)$, new applications of the above rules may be possible. The graph obtained after i repetitions of the triangle completion on H is denoted by $\triangle^i(H)$. We call H *stable* if $\triangle(H) = H$.

Remark 5.12. An undirected uncoloured graph H with $V(H) = V_u \mathbin{\dot\cup} V_\ell$ is stable if and only if every connected component of H has the following properties:

- The graph induced on V_u is a clique.
- The graph induced by the edges between V_u and V_ℓ is complete bipartite.

Now we describe the strategy that Mini derives from the auxiliary graphs for the original game. It is also shown in Algorithm 5.2. Mini first performs a complete clean-up step on her input graph. Then, as long as the auxiliary graph of her obtained graph is not stable, she iterates the following process on it: first she performs an iteration of the Weisfeiler-Leman algorithm on G, then a complete clean-up step. She returns the first completely cleaned-up graph G' for which $\mathrm{Aux}(G')$ is stable to Maxi, who then performs his next turn. (We will see that Algorithm 5.2 always terminates.)

We claim that if Mini follows this strategy, then she only computes $O(n \cdot 2^{t(n)})$ pairwise different auxiliary graphs. To show this, we need the following lemma.

Algorithm 5.2 One turn of Mini in the 2-player game on input G.

Input: A coloured graph G.
Output: A refinement G' of G satisfying $G \succeq G' \succeq \widetilde{G}$.

```
G ← clean(G)
while $\triangle\big(\mathrm{Aux}(G)\big) \neq \mathrm{Aux}(G)$ do
    $G \leftarrow G^{(1)}$
    $G \leftarrow \mathrm{clean}(G)$
end while
return G
```

Lemma 5.13. Let $m \in \mathbb{N}$ and let $H^1, \dots, H^k$ be a sequence of graphs such that for all $i \in [k]$, the following hold.

- $V(H^i) = V^i_\ell \mathbin{\dot\cup} V^i_u$.
- $|V^i_\ell| = |V^i_u| \leq m$.
- $V^{i+1}_\ell \supseteq V^i_\ell$ and $V^{i+1}_u \supseteq V^i_u$.
- $\triangle(H^i) \subseteq H^{i+1}$.
- $H^i \neq H^{i+1}$.

Furthermore, assume that for all i, the edge set of $H^i[V^i_\ell]$ is empty. Then $k \leq 10m$.

Proof: Without loss of generality, we can assume $m = |V^k_\ell| = |V^k_u|$. Note that every counterexample satisfies $k \leq 2m + \frac{(2m)^2}{2}$. Let $H^1, \dots, H^k$ be a longest counterexample to the statement of the lemma (i.e. there is no $k' > k$ for which there is a counterexample of length k'). By inserting $V(H^k) \setminus V(H^1)$ as a set of isolated vertices into H^1 and adjusting the vertex sets of all other H^i accordingly, some of the new graphs H^i might become equal their successor in the sequence. When deleting such copies, we shorten the sequence by at most $2m$. Thus, it suffices to show that $k \leq 8m$ holds under the assumption that $|V^i_\ell| = |V^i_u| = m$.

Furthermore, we can assume that the graphs H^i have as few edges as possible. More formally, we can assume that H^1 is empty and for the cases that $\triangle(H^i) \neq H^i$, we have $H^{i+1} = \triangle(H^i)$ and, for every other i, we have $|E(H^{i+1}) \setminus E(H^i)| = 1$. (If the sequence $H^1, \dots, H^k$ does not have this property, we can easily transform it into a longer counterexample by performing the insertions one by one, contradicting the maximality of k.)

We claim that, for every j with $\triangle(H^j) = H^j$, there is some number $r \leq 4$ such that $\triangle(H^{j+r}) = H^{j+r}$, i. e., the graph H^{j+r} is stable again.

Suppose $\triangle(H^j) = H^j$, that is, H^j is stable and has the structure described in Remark 5.12. If $H^{j+1} = \triangle(H^{j+1})$, the claim is true. Thus assume $H^{j+1} \neq \triangle(H^{j+1})$. Remember that, by the minimality of our counterexample, we have $H^{j+2} =$

$\triangle(H^{j+1})$. By the assumptions of the lemma, the new edge e in H^{j+1} either connects two upper vertices or an upper and a lower vertex. Suppose first that $e = \{v, v'\}$ with $v, v' \in V_u$. Let $U(v)$ and $U(v')$ be the cliques containing v and v' among V_u, respectively, and let $L(v)$ and $L(v')$ be the vertices in V_ℓ that are connected to $U(v)$ and $U(v')$, respectively. In $\triangle(H^{j+1})$, the edges between v and $U(v')$ and between v' and $U(v)$ are inserted, as well as the edges between $L(v)$ and v' and between $L(v')$ and v. Now if $\triangle^2(H^{j+1}) = \triangle^1(H^{j+1})$, the statement holds. Otherwise, in $\triangle^2(H^{j+1})$, the graph induced by $U(v) \,\dot\cup\, U(v')$ is rendered a clique and all edges between $L(v)$ and $U(v')$ and between $L(v')$ and $U(v)$ are inserted, resulting in a connected component that is complete bipartite between its upper and its lower vertices.

Suppose now that $e = \{v, v'\}$ with $v \in V_u$ and $v' \in V_\ell$. If, in H^j, the vertex v' is not adjacent to any upper vertex, then, in $\triangle(H^{j+1})$, all edges between v' and $U(v)$ are inserted and $\triangle(H^{j+1})$ is stable. Otherwise, let $U(v')$ be the clique among V_u that is connected to v'. Similarly as in the case that e is an edge between upper vertices, in $\triangle^2(H^{j+1})$, the graph induced by $U(v) \,\dot\cup\, U(v')$ is a clique and all edges between $L(v)$ and $U(v')$ are present. The vertices in $L(v')$ are adjacent to all vertices in $U(v')$ and to v. A third triangle completion then accounts for the missing edges between $L(v')$ and $U(v)$.

Therefore, $\triangle^3(H^{j+1})$ is stable. Thus, the gap between two successive stable graphs is at most 4.

For every i such that H^i is stable, the graph H^{i+1} contains an edge that connects two connected components from H^i. We know that H^1 is stable and we have shown that the gap between stable graphs is at most 4. Thus, also the gap between the graphs H^{i+1} that contain an extra edge connecting two connected components from H^i is at most 4. There are at most $2m$ isolated vertices in H^1, which yields $k \leq 4 \cdot 2m$. □

Lemma 5.14. Let G be a graph that is completely cleaned up (i. e., G satisfies clean$(G) = G$). Then $\mathrm{Aux}(G^{(1)}) \supseteq \triangle\big(\mathrm{Aux}(G)\big)$.

Proof: Suppose G satisfies the assumption and let χ be its colouring. Let $V_u \dot\cup V_\ell$ be the partition of the vertices of $\mathrm{Aux}(G)$ into upper and lower vertices. Let (C_1, M_1) and (C_2, M_2) be two vertices of V_u both adjacent to the vertex $(D, N) \in V_\ell$. We prove that (C_1, M_1) and (C_2, M_2) are adjacent in $\mathrm{Aux}(G^{(1)})$.

By definition of the auxiliary graph, there exists a set of colours $\mathcal{C}_1$ such that in G, every vertex $v \in C_1$ satisfies

$$v \in M_1 \iff N^+_{\mathcal{C}_1}(v) = N.$$

There is also a set $\mathcal{C}_2$ such that in G, every vertex $v \in C_2$ satisfies

$$v \in M_2 \iff N^-_{\mathcal{C}_2}(v) = N.$$

Here, we use the fact that edge colourings respect converse equivalence (see Section 2.3).

For $m \in M_1$, we have that $m' \in M_2$ if and only if, for every vertex $w \in D$, it holds that

$$\chi(m, w) \in \mathcal{C}_1 \iff \chi(w, m') \in \mathcal{C}_2.$$

Since G is completely cleaned up, this does not only hold for every $w \in D$, but for every $w \in V(G)$. This implies that, in $G^{(1)}$, there is a set of colours $\mathcal{C}'$ such that every vertex $v \in C_1$ satisfies

$$v \in M_1 \iff N^{+}_{\mathcal{C}'}(v) = M_2.$$

To see this, suppose $v_1 \in M_1$. The new colour of an edge (v_1, v_2) contains in its second component the multiset that, for $w \in V(G)$, consists of the tuples $\bigl(\chi(w, v_2), \chi(v_1, w)\bigr)$ of colours from the previous iteration.

For $v_2 \in M_2$, we have

$$\bigl(\chi(w, v_2), \chi(v_1, w)\bigr) \in \begin{cases} \mathcal{C}_2 \times \mathcal{C}_1 & \text{if } w \in N \\ (\mathcal{C} \setminus \mathcal{C}_2) \times (\mathcal{C} \setminus \mathcal{C}_1) & \text{if } w \notin N, \end{cases}$$

where $\mathcal{C}$ is the set of all edge colours in G.

However, if $v_1 \in M_1$ and $v_2 \in (C_2 \setminus M_2)$, then the multiset in the new colour of the edge (v_1, v_2) will contain some element from $\bigl((\mathcal{C} \setminus \mathcal{C}_2) \times \mathcal{C}_1\bigr) \cup \bigl(\mathcal{C}_2 \times (\mathcal{C} \setminus \mathcal{C}_1)\bigr)$.

Note that if $v_2 \notin C_2$, then the new colour of (v_1, v_2) is different from all colours of edges between C_1 and C_2 anyway, since G is completely cleaned up. Thus, we obtain that, in $G^{(1)}$, the colours of edges from M_1 to M_2 are distinct from the colours of edges from M_1 to the complement of M_2.

By symmetry, this shows that, in $\mathrm{Aux}(G^{(1)})$, the vertices (C_1, M_1) and (C_2, M_2) are adjacent.

A similar argument shows that if (C_1, M_1) is adjacent to (C_2, M_2) and an upper or lower vertex (D, N), then, in $\mathrm{Aux}(G^{(1)})$, there is an edge between (C_2, M_2) and (D, N).

We conclude that $\mathrm{Aux}(G^{(1)}) \supseteq \triangle\bigl(\mathrm{Aux}(G)\bigr)$. □

After each complete clean-up step that Mini performs in the original game, she computes an auxiliary graph. We show that two auxiliary graphs differ by at least one edge if between their computations, in the original game, at least one (vertex or edge) colour class that is only incident with small vertex colour classes is refined—either by a move of Maxi or during the subsequent complete clean-up step performed by Mini.

Lemma 5.15. Let G, G' with $G \succeq G'$ be two graphs appearing in the game that have been completely cleaned up. If there is a colour class in G that has been refined in G' and is only incident with small vertex colour classes, then it holds that $\mathrm{Aux}(G) \subsetneqq \mathrm{Aux}(G')$.

Proof: Suppose that, after a complete clean-up step, a small vertex colour class of G has been split into at least two new small vertex colour classes C' and C'' in G'. Thus, because of Condition C1 of a completely cleaned-up graph, the graph $\mathrm{Aux}(G')$ contains an edge between the upper and the lower copy of (C', C'), whereas none of these two vertices is present in $\mathrm{Aux}(G)$.

Now suppose that after a complete clean-up step, an edge colour class C of G only incident with small vertex colour classes has been split in G' into at least two new edge colour classes C' and C'' without causing any splitting of small vertex colour classes. Consequently, there exists a vertex v in a small vertex colour class A of G' which satisfies that $\emptyset \subsetneqq N^+_{C'}(v) \subsetneqq N^+_C(v)$. Now we define

$$M := \big\{u \in A \mid N^+_{C'}(u) = N^+_{C'}(v)\big\}.$$

Let N be the vertex colour class in G containing $N^+_{C'}(v)$. It suffices to observe that the edge $\big\{(A, M), \big(N, N^+_{C'}(v)\big)\big\}$ is present in $\mathrm{Aux}(G')$, but not in $\mathrm{Aux}(G)$. □

We wish to bound the costs that Mini incurs within Algorithm 5.2. To do so, we need to bound the costs that are incurred before and within the while loop. Let $G_1, G_2, \ldots$ be the sequence containing the following two types of graphs in their order of appearance during the game. On the one hand, it contains the graphs that are played by Maxi and cleaned up by Mini and have incurred some refinement of edges only incident with small vertex colour classes. On the other hand, it contains the graphs that are results of computations of Line 4 of Algorithm 5.2 across the entire play of the game. Thus, the sequence consists of all completely cleaned-up graphs played in the game right after a refinement of colour classes that are only incident with small vertex colour classes.

Lemma 5.16. The sequence of graphs $G_1, G_2, \ldots$ has length at most $O(2^{t(n)}n)$.

Proof: We set $m := 4n \cdot 2^{t(n)}$ for the threshold function t. Each of the graphs $\mathrm{Aux}(G_1), \mathrm{Aux}(G_2), \ldots$ has size at most m. Thus, since $G_{i+1} \preceq G_i$, it holds that $\mathrm{Aux}(G_i) \subseteq \mathrm{Aux}(G_{i+1})$ by Remark 5.11. The while loop in Algorithm 5.2 is only entered in the case that $\triangle\big(\mathrm{Aux}(G)\big) \neq \mathrm{Aux}(G)$. Also, in that situation, the graph G is completely cleaned up. We conclude with Lemma 5.14 that $\triangle\big(\mathrm{Aux}(G_i)\big) \subseteq \mathrm{Aux}(G_{i+1})$. By Lemma 5.15, we obtain the proper inclusion $\mathrm{Aux}(G_i) \subsetneqq \mathrm{Aux}(G_{i+1})$.

Therefore, the sequence of the auxiliary graphs $\mathrm{Aux}(G_1), \mathrm{Aux}(G_2), \ldots$ fulfils the conditions of Lemma 5.13 with $m = O(2^{t(n)}n)$, implying that the length of the graph sequence $G_1, G_2 \ldots$ is $O(2^{t(n)}n)$. □

The lemma in particular implies that Algorithm 5.2 terminates. Now we can bound the number of iterations for small colour classes.

Corollary 5.17. The number of iterations in which colour classes that are only incident with small vertex colour classes are refined is $O(2^{t(n)}n)$.

Proof: By Lemma 5.16, the sequence of the computed auxiliary graphs satisfies the conditions from Lemma 5.13. Thus, every subsequence of it also satisfies these conditions. If we only consider the subsequence of auxiliary graphs computed after moves refining a colour class that is only incident with small vertex colour classes, we know that the sequence has length $O(m)$, where m is the maximum number of upper vertices in an auxiliary graph. Thus, the sequence has length $O(n \cdot 2^{t(n)})$.

With Lemma 5.15, the length of the sequence is an upper bound on the number of iterations in which colour classes that are only incident with small vertex colour classes are refined. □

From the corollary, we can directly conclude the following lemma, which proves that, on graphs of bounded colour class size, the 2-dimensional Weisfeiler-Leman algorithm only takes a linear number of iterations to terminate.

Lemma 5.18. *The number of iterations of the 2-dimensional Weisfeiler-Leman algorithm on graphs with n vertices of colour class size at most t is $O(2^t n)$.*

We have assembled all the required tools to prove our main theorem concerning the upper bound on the number of iterations.

Theorem 5.19. *The number of iterations of the 2-dimensional Weisfeiler-Leman algorithm on graphs of order n is at most $O(n^2/\log(n))$.*

Proof: Let G be a graph with n vertices. To show a bound on the number of iterations of the Weisfeiler-Leman algorithm applied to G, it suffices to show an upper bound on the costs of the 2-player game that we defined. Assume that both players play optimally. By Corollary 5.9 the total costs of moves in which a colour class incident with a large vertex colour class is split is $O\big(n^2/t(n)\big)$. By Corollary 5.17, the total costs of moves in which a small colour class is split are $O(2^{t(n)} \cdot n)$. Therefore, the entire game costs $O\big(n^2/t(n)\big) + O(2^{t(n)} \cdot n)$ and setting $t := \log_2(n)/2$ yields the upper bound of $O\big(n^2/\log_2(n)\big)$. □

By the correspondence from Theorem 3.18, our theorem translates to a new bound on the quantifier depth for formulae in C that define a graph.

Corollary 5.20. If two n-vertex graphs can be distinguished by the 3-variable first-order logic with counting C^3, there is also a formula in C^3 with quantifier depth at most $O(n^2/\log(n))$ distinguishing the two graphs.

5.2 Logics without Counting

As stated in Corollary 5.20, the bound we have proven for the 2-dimensional Weisfeiler-Leman algorithm leads to bounds on the quantifier depth in the 3-variable fragment of first-order logic with counting C^3. One may wonder what happens for the 3-variable fragment of first-order logic without counting L^3. A priori, it is not clear that there should be a relation between the depths of formulae distinguishing graphs in C^3 and in L^3. We can thus not draw conclusions about L^3 from our theorems.

However, a careful analysis of our proof reveals that we could have obtained the same results for the logic without counting.

For this, one has to redefine the colouring computed by the Weisfeiler-Leman algorithm so as to obtain a non-counting version of the 2-dimensional Weisfeiler-Leman algorithm (see [138, Subsection 2.2.4]). In this version, we replace the multiset $\mathcal{M}$ with a set with the same elements. Refinement and stability is then defined with respect to this new operator. Most lemmas apply to the new situation with verbatim proofs. Most notably, for the potential function for large colour classes, the definition does not use a multiset.

Condition C2 in the clean-up steps of Mini then has to be replaced with a condition which requires, for all $v_1, v_2 \in C_1$, that $\{\chi(v_1, u) \mid u \in C_2\} = \{\chi(v_2, u) \mid u \in C_2\}$ and also $\{\chi(u, v_1) \mid u \in C_2\} = \{\chi(u, v_2) \mid u \in C_2\}$, where χ is the current colouring.

Finally, we highlight the fact that in the auxiliary graph, the definition of adjacency does not require counting either. For example, the existence of a set of colours $\mathcal{C}' \subseteq \mathcal{C}$ such that every vertex v satisfies

$$v \in M \iff N^+_{\mathcal{C}'}(v) = N,$$

can obviously be expressed in a L^2-formula, i.e. without counting quantifiers. We conclude that analogous statements to our theorems also hold for the 3-variable first-order logic without counting L^3.

5.3 Discussion

In the game we considered in this chapter, we basically want to allow Maxi to refine to arbitrary colourings that do not necessarily arise from the application of the 2-dimensional Weisfeiler-Leman algorithm. This might violate the converse equivalence of the colourings, i.e. it might happen that $\lambda(v_1, u_1) \neq \lambda(v_2, u_2)$ but $\lambda(u_1, v_1) = \lambda(u_2, v_2)$ (see Section 2.3). However, this is not a problem, since we can actually drop the requirement for converse equivalence. In this case, at the expense of a more technical clean-up procedure, Mini could then (under the new definition of the Weisfeiler-Leman algorithm) restore converse equivalence. Overall, we obtain essentially the same results.

We have shown that the number of iterations of the 2-dimensional Weisfeiler-Leman algorithm is in $O\big(n^2/\log(n)\big)$, which is the first improvement over the trivial upper bound. The factor $1/\log(n)$ arises from a trade-off between considerations in large and small vertex colour classes, and thus it stems from the factor of 2^t appearing in Lemma 5.18. However, an improvement of the bound on the iteration number for graphs with bounded colour class size would not directly improve the overall iteration number. Indeed, since refinements within small colour classes may be caused by refinements of large colour classes, and also since small colour classes might appear only over time, we need to bound the number of all refinements involving small colour classes also within graphs that may not have bounded colour class size.

The work presented in this chapter laid the foundation for further research on the iteration number of the higher-dimensional Weisfeiler-Leman algorithm. In subsequent work three years after the publication of the results from this chapter, Lichter, Ponomarenko, and Schweitzer take a highly algebraic approach to analyse the iteration number of the 2-dimensional Weisfeiler-Leman algorithm and show the new upper bound $\mathrm{WL}_2(n) \in O(n \log n)$ [120]. They introduce so-called walk logic and walk refinements and obtain a version of the algorithm whose refinement steps are based on counting walks in a graph. Via the correspondence to logics, this implies that two n-vertex graphs that are distinguishable in C^3 can be distinguished via a formula of quantifier depth in $O(n \log n)$. Still, since the arguments in [120] rely heavily on the counting, no improvement over our bound of $O(n^2/\log n)$ on the quantifier depth of a distinguishing formula in the 3-variable logic without counting L^3 has been found so far.

Our proof for the 3-variable logics C^3 and L^3 already requires a careful analysis of the interaction between the small vertex colour classes. Still, it remains an interesting open question whether our techniques or the ones presented in [120] can be generalised to also show bounds on the depth of formulae with more variables.

6 Graphs Identified by Colour Refinement

In the previous two chapters, we have presented upper and lower bounds on the number of iterations of certain dimensions of the Weisfeiler-Leman algorithm. By Theorem 3.18, these correspond to bounds on the quantifier depth of C-formulae that identify the input graph.

A second parameter that can be regarded as a measure for the complexity of a C-formula and has a precise connection to the Weisfeiler-Leman algorithm is the number of variables. For a graph, the number of variables needed to define it in C essentially corresponds to the Weisfeiler-Leman dimension needed to distinguish the graph from every other, non-isomorphic graph (see Corollary 3.21). By Theorem 3.13, even when allowing arbitrary quantifier depth, for every $k \in \mathbb{N}$, there are pairs of non-isomorphic graphs which are not distinguished by any formula in C^k. Thus, also the $(k+1)$-dimensional Weisfeiler-Leman algorithm fails to distinguish those graphs. So, as a general graph isomorphism test, the algorithm is incomplete. Nevertheless, the situation can change when we are guaranteed some structural properties about the input. Accordingly, in the remainder of this thesis, we concern ourselves with the Weisfeiler-Leman dimension of certain graph classes.

We first examine the expressive power of the 1-dimensional Weisfeiler-Leman algorithm. To this end, we will exploit the correspondence to the logic C^2. In this chapter, we give a complete characterisation of the graphs and the finite relational structures that are identified by C^2, i.e. the 1-dimensional Weisfeiler-Leman algorithm (see Theorems 6.18 and 6.29). Thanks to the classification, it can be decided in almost linear time whether a relational structure is identified.

To show that our classification is correct, i.e. that every graph described in the characterisation is indeed identified, we employ explicit constructions that solve the inversion problem and the canonisation problem for the logic C^2. The inversion problem for an invariant $\mathcal{I}$ consists in the question whether, to a particular $\mathcal{I}$-value y, a graph G (or more generally, a finite structure) with $\mathcal{I}(G) = y$ can be constructed. In the case of C^2, such an invariant is the multiset of C^2-types of pairs of vertices. In [137], Otto shows that the inversion problem and the canonisation problem for C^2 can be solved in polynomial time. Our direct constructions provide an alternative proof for this. In fact, they show that the inversion problem for C^2 can be solved in linear time.

The classification that we obtain can also be used to draw several conclusions about properties of C^2-identified graphs. For example, we show that if an undirected graph is identified by C^2, every vertex-coloured version of it is identified as well (see Corollary 6.20). This statement holds for finite relational structures, too, but is not true for C^k with $k > 2$, i.e. higher dimensions of the Weisfeiler-Leman algorithm.

The results presented in this chapter were published in [105].

6.1 Inversion and Canonisation

In this section, we treat the so-called *inversion problem* that is closely related to the question which graphs are identified by C^2 (i.e. Colour Refinement). It will provide some first insight about the structure of identified graphs.

A *complete invariant* of an equivalence relation $\equiv$ on an isomorphism-closed class $\mathcal{C}$ of relational structures is a mapping $\mathcal{I}$ from $\mathcal{C}$ to some set S, such that $\mathfrak{A} \equiv \mathfrak{B}$ if and only if $\mathcal{I}(\mathfrak{A}) = \mathcal{I}(\mathfrak{B})$. We say that $\mathcal{I}$ *admits linear-time inversion on* $\mathcal{C}$ if, given any $I \in S$, one can construct in time linear in the output size a structure $\mathfrak{A} \in \mathcal{C}$ with $\mathcal{I}(\mathfrak{A}) = I$ or decide that no such structure exists. We call the underlying algorithmic challenge, i.e. the task to construct such an $\mathfrak{A} \in \mathcal{C}$ (or to decide that none exists), the *inversion problem.* In the following, the equivalence relation we consider is C^2-equivalence. Note that a structure $\mathfrak{A}$ is identified by C^2 if and only if its C^2-equivalence class is a singleton. Thus, if a structure $\mathfrak{A}$ is identified by C^2, then a solution to the inversion problem must construct a structure that is isomorphic to $\mathfrak{A}$ when given $\mathcal{I}(\mathfrak{A})$.

Given an uncoloured graph G, one can define a linear ordering $P_1 \leq \ldots \leq P_t$ on the classes of its coarsest equitable partition, which only depends on the C^2-equivalence class of G (see [137]). This ordering allows us to define the natural invariant $\mathcal{I}^2_{\mathsf{C}}$, which maps G to $(\bar{s}, M)$, where $\bar{s}$ is the tuple $(|P_1|, \ldots, |P_t|)$ and M is a $t \times t$ matrix, such that every vertex in P_i has exactly M_{ij} neighbours in P_j. For coloured graphs, the invariant can also be adapted such that its value contains not only sizes of C^2-partition classes and degrees between them, but also actual encodings of the final colours and degrees with respect to each colour. The obtained mapping $\mathcal{I}^2_{\mathsf{C}}$ is a complete graph invariant of C^2 (see [137]). It is easy to see that $\mathcal{I}^2_{\mathsf{C}}$ is a complete invariant of C^2-equivalence.

In the following, we show that $\mathcal{I}^2_{\mathsf{C}}$ admits linear-time inversion. Otto proved that this invariant admits polynomial-time inversion [137].

Definition 6.1. A graph is *circulant* if it has a full cycle automorphism, i.e. an automorphism φ for which the cyclic group $\langle\varphi\rangle$ has just one single orbit (called the *cycle* of φ). A graph on vertex set $P \mathbin{\dot\cup} Q$ is *doubly-circulant* with respect to P and Q if it has an automorphism φ whose cycles are P and Q.

Analogously, a graph is *multi-circulant* with respect to a partition $\{P_1, \ldots, P_\ell\}$ of its vertex set if it has an automorphism with exactly ℓ cycles, each on one of the P_i.

A graph G is *vertex-transitive* if for every pair of vertices $v, v' \in V(G)$, there is an automorphism of G mapping v to v'.

Note that in a doubly-circulant graph, it may hold that $|P| \neq |Q|$. By definition, every circulant graph is vertex-transitive. However, the converse does not hold. For example, the 1-skeleton of a 3-dimensional cube is vertex-transitive, but not circulant [117]. It is well-known that circulant graphs of order n can be constructed by numbering the vertices from 0 to $n-1$, picking a set $S \subseteq \{1, \ldots, \lfloor \frac{n}{2} \rfloor\}$ of distances and inserting all edges between all pairs of vertices whose distance of indices in the cyclic ordering is contained in S.

For a k-regular graph on n vertices, the product $k \cdot n$ equals $2m$, where m is the number of edges of the graph. In particular, $k \cdot n$ is even. Moreover, for all k, $n \in \mathbb{N}$ with $k < n$ and for which $k \cdot n$ is even, a circulant k-regular graph on n vertices can be constructed in time linear in the output size. Indeed, to obtain a k-regular graph in case k is even, one must only ensure that $|S| = \frac{k}{2}$. If k is odd (which implies that n is even), it suffices to have $\frac{n}{2} \in S$ and $|S| = \frac{k+1}{2}$ (see Figure 6.1a for $n = 12$ and $S = \{1, 2, 3, 6\}$). We call this the *circulant construction*.

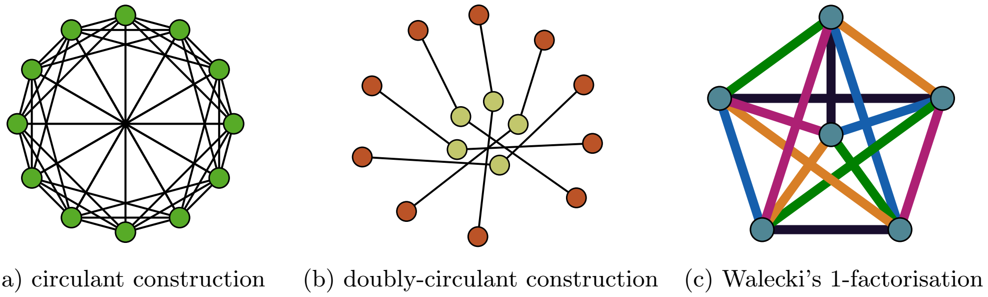

(a) circulant construction (b) doubly-circulant construction (c) Walecki's 1-factorisation

Figure 6.1: The graph constructions used in the proof of our characterisations.

As we argued above, the condition of $k \cdot n$ being even is necessary and sufficient for the existence of a k-regular circulant graph on n vertices (with $k < n$). Similarly, there is a simple criterion for the existence of doubly-circulant graphs. By double-counting, every (k, ℓ)-biregular graph on a bipartition (P, Q) with $|P| = m$ and $|Q| = n$ satisfies $k \cdot m = \ell \cdot n$. The following lemma says that this condition is essentially sufficient to guarantee the existence of a doubly-circulant (k, ℓ)-biregular graph and that such a graph can be constructed in linear time.

Lemma 6.2. *Given k, ℓ, m, $n \in \mathbb{N}$ with $k \leq n$, $\ell \leq m$ and $k \cdot m = \ell \cdot n$, one can construct in linear time a (k, ℓ)-biregular graph on a bipartition (P, Q) with $|P| = m$ and $|Q| = n$ which is doubly-circulant with respect to P and Q.*

Moreover, fix arbitrary cyclic orderings $\prec_P$ and $\prec_Q$ on P and Q. The graph G can be chosen such that the automorphism witnessing G being doubly-circulant maps every vertex in P and Q to its successor with respect to $\prec_P$ and $\prec_Q$, respectively.

Proof: Let $P \coloneqq \{v_0, \dots, v_{m-1}\}$ and $Q \coloneqq \{v'_0, \dots, v'_{n-1}\}$. We insert edges according to the following algorithm (see also Figure 6.1b): Let r be the least integer such that the edge $\{v_0, v'_r\}$ has not yet been inserted. (So in the beginning, $r = 0$.) For every v_i, we insert edges to all vertices $v'_{(i+r+s) \bmod n}$, where s ranges over all multiples of m up to $\operatorname{lcm}(m, n)$, the least common multiple of m and n. Thus, a vertex of P in the obtained graph has degree $i_{P,Q} = \frac{\operatorname{lcm}(m,n)}{m}$. Let $E_{P,Q}$ be the set of edges in the final graph we intend to construct. We have $|E_{P,Q}| = m \cdot k = n \cdot \ell$ edges, so $|E_{P,Q}|$ is a multiple of both m and n. Thus, $k = \frac{n \cdot \ell}{m} = \frac{|E_{P,Q}|}{m} \geq \frac{\operatorname{lcm}(m,n)}{m} = i_{P,Q}$. If the inequality is proper, we can repeat the above process.

In one iteration, the degree of a vertex in P increases by $\frac{\operatorname{lcm}(m,n)}{m}$, a divisor of $\operatorname{lcm}(m, n)$. Since $|E_{P,Q}|$ is a multiple of $\operatorname{lcm}(m, n)$, throughout the iterations, the degree of a vertex in P never exceeds k. It strictly increases with every iteration, so, after a finite number of iterations, we obtain a graph H in which every vertex in P has exactly k neighbours in Q (and every vertex in Q has exactly ℓ neighbours in P).

We construct an automorphism φ of H such that the orbits with respect to the cyclic group $\langle \varphi \rangle$ generated by φ are P and Q. Define φ such that it maps a vertex $v_i \in P$ to $v_{(i+1) \bmod m}$ and a vertex $v'_j \in Q$ to $v'_{(j+1) \bmod n}$. By construction, $\varphi(P) = P$ and $\varphi(Q) = Q$, so it remains to show that φ is a graph isomorphism, i.e. that $\{\varphi(u), \varphi(v)\} \in E_{P,Q}$ if and only if $\{u, v\} \in E_{P,Q}$. Let $v_i \in P$, $v'_j \in Q$. We have $\varphi(v_i) = v_{(i+1) \bmod m}$ and $\varphi(v'_j) = v'_{(j+1) \bmod n}$.

Suppose $\{v_i, v'_j\} \in E_{P,Q}$. Then $j \equiv i + r + s$ modulo n for some $r \in \{0, \dots, m - 1\}$ and some multiple s of m with $s \leq \operatorname{lcm}(m, n)$. Note that, by the above algorithm, whether two vertices $v \in P$ and $v' \in Q$ are adjacent only depends on the difference modulo n of their indices in the respective cyclic orderings of P and Q modulo n. However, $(j + 1) \equiv (i + 1) + r + s$ modulo n and thus, $j - i$ and $(j + 1) - (i + 1)$ equal $r + s$ modulo n. Hence, $\{\varphi(v_i), \varphi(v'_j)\} \in E_{P,Q}$. By a symmetric argument, we get equivalence. □

Figure 6.1b depicts a $(2, 1)$-biregular graph on bipartition classes P and Q of size 5 and 10, respectively, which can be obtained as described in the above proof.

The coarsest equitable partition of a graph is not necessarily its orbit partition, i.e. the partition into the vertex orbits with respect to the automorphism group of the graph. For example, on the disjoint union of two cycles of different sizes, the coarsest equitable partition is the unit partition, whereas the graph actually has two orbits. However, we can use the lemma to see that for each C^2-equivalence class, there is a representative whose coarsest equitable partition is indeed the orbit partition, as stated in the following theorem.

Theorem 6.3. *For every graph G, there is a C^2-equivalent graph H which is multi-circulant with respect to the coarsest equitable partition of G.*

Proof: Let $\{P_1, \dots, P_t\}$ be the coarsest equitable partition of G. For every $i \in [t]$, the graph $G[P_i]$ is regular. By the circulant construction, there is a circulant graph $H(P_i)$ on each P_i, that is C^2-equivalent to $G[P_i]$. For $i \in [t]$, let $\prec_i$ be a cyclic ordering of P_i. By Lemma 6.2, for all i, j with $i \neq j$, we can connect $H(P_i)$ to $H(P_j)$ in such a way that the resulting graph on vertex set $P_i \dot{\cup} P_j$ is C^2-equivalent to $G[P_i \cup P_j]$ and doubly-circulant with respect to P_i and P_j. Furthermore, the construction can be performed such that there is a witnessing automorphism $\varphi_{i,j}$ that maps every vertex in P_i to its successor with respect to $\prec_i$ and similarly, for P_j. Thus, we can consistently combine all the $\varphi_{i,j}$ to an automorphism φ of the whole graph H such that the cycles of φ are the P_i. □

Note that the proof provides a canonical way to construct the graph H. The theorem shows that we can solve the inversion problem for $\mathcal{I}^2_{\mathsf{C}}$ on graphs in linear time.

Corollary 6.4. *$\mathcal{I}^2_{\mathsf{C}}$ admits linear-time inversion on the class of all graphs.*

Given an equivalence relation $\equiv$ on a class $\mathcal{C}$ of structures, the *canonisation problem* for $\equiv$ is the problem of finding a map $c\colon \mathcal{C} \to \mathcal{C}$ such that for every $\mathfrak{A} \in \mathcal{C}$, we have $c(\mathfrak{A}) \equiv \mathfrak{A}$ and for all $\mathfrak{A}, \mathfrak{B} \in \mathcal{C}$ with $\mathfrak{A} \equiv \mathfrak{B}$, we have $c(\mathfrak{A}) \equiv c(\mathfrak{B})$. The map c is called a *canonisation* for $\equiv$ and $c(\mathfrak{A})$ is called the *canon* of $\mathfrak{A}$ (with respect to c). Typically, the goal is to find such a canonisation c that can be evaluated efficiently.

As a by-product of the described constructions, we obtain the following corollary.

Corollary 6.5. *With respect to C^2-equivalence, a canonisation of graphs can be found in time $O((n + m) \log n)$, where n is the number of vertices and m is the number of edges of the input graph.*

Proof: Given a graph G with n vertices and m edges, we can perform Colour Refinement to compute $\mathcal{I}^2_{\mathsf{C}}(G)$. This can be done in time $O((n + m) \log n)$ [32, 126, 139]. By Corollary 6.4, the canon can then be constructed in linear time. □

Theorem 6.3 also immediately yields one of our main structural insights about graphs that are identified by the 1-dimensional Weisfeiler-Leman algorithm.

Corollary 6.6. *If a graph G is identified by C^2, then its coarsest equitable partition is the orbit partition.*

Proof: Suppose G is identified by C^2. That is, there is a formula $\varphi \in \mathsf{C}^2$ that distinguishes G from all non-isomorphic graphs. In particular, there may not be non-isomorphic graphs that are C^2-equivalent to G. By Theorem 6.3, there is a graph H which is C^2-equivalent to G and which is multi-circulant with respect to its coarsest equitable partition. However, since G is identified by C^2, there is a formula $\varphi \in \mathsf{C}^2$ that distinguishes G from all non-isomorphic graphs. In particular, there are no graphs that are C^2-equivalent to G, but not isomorphic to G. Thus, G

and H are isomorphic and hence, the coarsest equitable partition on G is the orbit partition as well, since the coarsest equitable partition and the orbit partition are isomorphism-invariant. □

Note that the statement in Corollary 6.6 cannot be turned into an "if and only if". For example, on the graph consisting of two disjoint 3-cycles, the coarsest equitable partition is the orbit partition, but the graph is not identified by C^2 since it is C^2-equivalent to a 6-cycle.

In Section 6.4, we also show that Corollary 6.6 does not hold for any logic C^k with $k \geq 3$.

For the remainder of this chapter, the concept of graph factors will be useful.

Definition 6.7. For a graph G and $k \in \mathbb{N}$, a *k-factor* of G is a k-regular subgraph of G with vertex set $V(G)$. A *k-factorisation* of G is a collection of k-factors of G whose edge sets form a partition of $E(G)$.

See Figure 6.1c for an example of a 1-factorisation of the complete graph K_6.

Remark 6.8. It is natural to ask whether Corollary 6.6 holds for finite relational structures in general. This is not the case, not even for edge-coloured undirected graphs. For example, let the colours of the edges of the complete graph K_6 correspond to a 1-factorisation, i.e. a partitioning of the edges into perfect matchings (see Figure 6.1c). There is only one 1-factorisation of K_6 up to isomorphism (see [121]). This graph is identified by C^2 (see Theorem 6.26). By a case analysis, it is not hard to see that the graph is rigid, i.e. its orbit partition is discrete. However, its coarsest equitable partition is the unit partition because the described initial colouring is equitable, i.e. not refined by C^2.

6.2 Characterisation of the Graphs Identified by 1-WL

Here we examine the graphs that are identified by the 1-dimensional Weisfeiler-Leman algorithm, or equivalently, by the logic C^2, and give a complete characterisation of them. We derive various results from the characterisation.

For a graph F and a set $E \subseteq \big\{\{u,v\} \,\big|\, u,v \in V(F)\big\}$, we denote by $F+E$ the graph with vertex set $V(F)$ and edge set $E(F) \cup E$.

Definition 6.9. Let T be a vertex-coloured tree with a designated vertex v. For $i \in [5]$, let (T_i, v_i) be an isomorphic copy of (T, v). Let F be the disjoint union of the five trees (T_i, v_i) and let E be the edge set of a 5-cycle on vertex set $\{v_1, \dots, v_5\}$. Then we call the graph $F+E$ a *bouquet*.

A *bouquet forest* is a disjoint union of vertex-coloured trees and non-isomorphic bouquets.

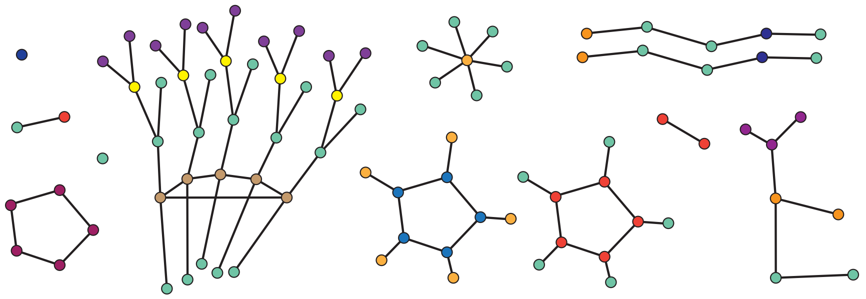


Figure 6.2: A bouquet forest.

Figure 6.2 depicts a bouquet forest with four bouquets.

Notation 6.10. *For sets P, Q, we define* $[P, Q] \coloneqq \big\{\{v, w\} \mid v \neq w,\ v \in P,\ w \in Q\big\}$.

Definition 6.11. Let G be a graph and let $\chi \coloneqq \chi_{G,1}$ be the stable colouring of $V(G)$ with respect to Colour Refinement, i.e. C^2. Let $\pi \coloneqq \pi_{G,1}$. We define the *flip of* G to be the vertex-coloured graph (F, χ) with $V(F) = V(G)$ and $E(F) = E(G) \,\Delta\, ([P_1, Q_1] \cup \ldots \cup [P_t, Q_t])$, where the (P_i, Q_i) are all pairs of (not necessarily distinct) C^2-partition classes of G which satisfy

$$|[P_i, Q_i] \cap E(G)| > |[P_i, Q_i] \setminus E(G)|\,.$$

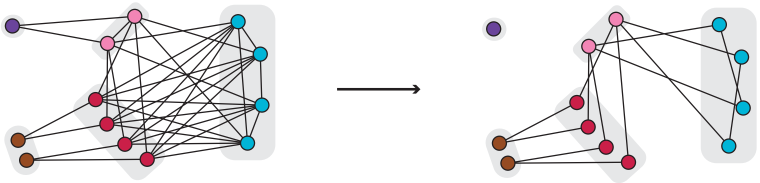


Figure 6.3: A graph with its C^2-partition and its flip.

The symbol Δ in the definition denotes, as usual, the symmetric difference. To obtain the flip of G, we first calculate the C^2-partition of G, i.e. the partition of $V(G)$ into C^2-types (which corresponds to the stable partition $\pi_{G,1}$ computed by Colour Refinement on input G). We then check, for each class P, whether G has more edges in P than non-edges. If so, we take the complement of the edge relation there. We do the same for all pairs P, Q of distinct C^2-partition classes, considering the edges in $[P, Q]$ there. See Figure 6.3 for an example of a graph and its flip. If the obtained graph is again the coloured graph (G, χ), i.e. if G has in each class and between every pairs of classes at most as many edges as non-edges, we also call (G, χ) a *flipped graph.*

While we may consider arbitrary vertex colourings in general, we want to stress the fact that the initial colouring χ' of the flip (F, χ') of a graph G is always the colouring $\chi_{G,1}$ of G. Note that χ' induces an equitable partition of F, although it might not be the coarsest one. For example, if the brown class in Figure 6.3 consisted just of a single vertex, then in the flip, the vertices in the brown and the purple colour class would both be isolated and would therefore be contained in the same C^2-partition class of the uncoloured version of the flip. The aim of this section is to prove the following classification: a graph is identified by C^2 if and only if its flip is a bouquet forest. We first argue that we can restrict ourselves to flipped graphs.

Lemma 6.12. *A graph G is identified by C^2 if and only if its flip is identified by C^2 (as a coloured graph).*

Proof: Let G be a graph and let χ be its C^2-colouring. Let (F, χ) be the flip of G. Note that $V(F) = V(G)$ by definition. Let G' be a graph on $V(G)$ with the same C^2-colouring χ that is non-isomorphic to G. Let (F', χ) be its flip. Note that, in particular, the C^2-partition classes in G' have the same sizes as in G and the edge sets running between the corresponding pairs of such classes have equal cardinalities in G and G'. Thus, for every pair of C^2-partition classes P and Q, it holds that $[P, Q] \cap E(G) = E(F)$ if and only if $[P, Q] \cap E(G') = E(F')$. Hence, every isomorphism from (F, χ) to (F', χ) would also be an isomorphism from G to G'. Thus, (F, χ) and (F', χ) are C^2-equivalent, but not isomorphic.

Similarly, for the converse direction, let (F', χ) be a graph that is C^2-equivalent but not isomorphic to (F, χ). We may assume that χ is the C^2-colouring of both (F, χ) and (F', χ). Again, both graphs have equal colour class sizes and colour degrees within and between them with respect to χ. By inverting in (F', χ) the edge sets between all pairs of partition classes P and Q for which $|[P, Q] \cap E(G)| > |[P, Q] \setminus E(G)|$, we obtain a graph G' which is C^2-equivalent but not isomorphic to G. □

For a graph G, we know that G and $(G, \chi_{G,1})$ have the same automorphisms. If, for a C^2-partition class P of G, the induced subgraph $G[P]$ is not identified by C^2, then G is not identified by C^2. Indeed, replacing $G[P]$ with a C^2-equivalent graph does not change the $\mathcal{I}^2_{\mathsf{C}}$-value of the graph and since $\mathcal{I}^2_{\mathsf{C}}$ is a complete invariant for C^2, the entire resulting graph is C^2-equivalent to G. Similarly, if a subgraph consisting of edges $[P, Q] \cap E(G)$ for distinct classes P and Q is not identified, then G is not identified either. We are going to exploit this simple fact in order to prove the classification. The following two lemmas provide strong restrictions to the edge relation of the regular and biregular graphs identified by C^2.

Lemma 6.13. *A flipped regular graph is identified by C^2 if and only if it is a graph with no edges, a matching, or a 5-cycle.*

Proof: The "if"-part is easy to check. For the "only if"-part, let G be a flipped k-regular graph on n vertices. Assume that G has at least one edge and it is neither a matching nor a 5-cycle. Then we have $n > 2$ and $k \geq 2$. In order to show that G is not identified by C^2, we construct two non-isomorphic graphs H and H', both C^2-equivalent to G. Note that the graphs which are C^2-equivalent to G are exactly the k-regular graphs on n vertices.

Since $k \geq 2$, we can apply the circulant construction from Subsection 6.1 in a way so that the resulting graph is connected (namely, we let $1 \in S$, where S is the set of distances of indices between adjacent vertices). We choose H to be such a connected circulant graph.

For the construction of H', we do a case analysis:

(Case 1: k is odd) Then n is even.

(Case 1.1: $\frac{n}{2}$ is even) Since G is flipped, we know that $k \leq n-1-k$, which implies $k < \frac{n}{2}$. Let H' be the disjoint union of two k-regular graphs on $\frac{n}{2}$ vertices.

(Case 1.2: $\frac{n}{2}$ is odd) Since k is odd and the graph is flipped, we have $2 < k < \frac{n}{2}-1$. Let H' be the disjoint union of two k-regular graphs on $\frac{n}{2}+1$ and $\frac{n}{2}-1$ vertices, for example, circulant ones.

In both cases, we have $H \ncong H'$, since H' is disconnected.

(Case 2: k is even) We distinguish cases according to the parity of n.

(Case 2.1: n is even) Again, we have $2 \leq k < \frac{n}{2}$. Since k is even, independently of the parity of $\frac{n}{2}$, there is a k-regular graph on $\frac{n}{2}$ vertices. We let H' be the disjoint union of two copies of such a graph. Again, we have $H \ncong H'$ since H' is disconnected.

(Case 2.2: n is odd) Since G is flipped, we have $n \geq 2k+1$.

If $n > 2k+1$, from the parities of k and n, we have $k < \frac{n}{2} - 1$ and we can choose H' to be the disjoint union of a k-regular graph on $\lfloor \frac{n}{2} \rfloor$ vertices and a k-regular graph on $\lceil \frac{n}{2} \rceil$ vertices.

In case $n = 2k+1$, note that there is no disconnected k-regular graph on $2k+1$ vertices. Indeed, by the k-regularity, every connected component would have to contain at least $k+1$ vertices and thus, the graph would have at least $2(k+1)$ vertices. Thus, in order to construct H', take two disjoint k-cliques. Connect half of the first clique via a matching to half of the second clique. Lastly, connect all vertices in the unmatched halves to a new vertex v_0.

Since G is not a 5-cycle by assumption, we cannot have $k = 2$. For $k > 2$, the vertex v_0 is the only vertex which does not belong to a k-clique. Hence, H' is not vertex-transitive, therefore not circulant. This implies $H \ncong H'$. □

Lemma 6.14. *The logic C^2 identifies a flipped (k,ℓ)-biregular graph on bipartition (P,Q) with $k \leq \frac{|Q|}{2}$ and $\ell \leq \frac{|P|}{2}$ if and only if $k \leq 1$ or $\ell \leq 1$. Furthermore, if the graph is not identified, there is a non-isomorphic graph that is (k,ℓ)-biregular on the same bipartition (even if $k = \ell$).*

Proof: For the "if"-part, note that if $k \leq 1$ or $\ell \leq 1$, then G is a disjoint union of stars. Therefore, C^2 identifies G.

For the "only if"-part, let G be a flipped (k, ℓ)-biregular graph on bipartition (P, Q). Without loss of generality, assume that $k \geq \ell$. Suppose $k \geq 2$ and $\ell \geq 2$. In order to show that C^2 does not identify G, we construct non-isomorphic (k, ℓ)-biregular graphs H and H', both on bipartition (P, Q), which are C^2-equivalent to G. We will define H and H' such that H is connected and H' is disconnected. Let $|P| = m$ and $|Q| = n$. Note that our assumption $k \geq \ell$ implies $n \geq m$.

Let H be a connected (k, ℓ)-biregular graph on bipartition (P, Q). Such a graph exists due to the following argument. Starting with G, it suffices to explain how to repeatedly decrease the number of connected components while maintaining (k, ℓ)-biregularity. Since $k > 1$ and $\ell > 1$, each connected component has a cycle. Let C_1 and C_2 be two distinct connected components and let $\{p_1, q_1\}$ and $\{p_2, q_2\}$ be edges in these connected components, respectively, that lie on a cycle, where $p_i \in P$ and $q_i \in Q$. We replace the edges $\{p_1, q_1\}$ and $\{p_2, q_2\}$ with the edges $\{p_1, q_2\}$ and $\{p_2, q_1\}$. This decreases the number of connected components by 1 and maintains vertex degrees. We conclude that there is a connected (k, ℓ)-biregular graph on bipartition (P, Q).

We now define the graph H'. Suppose $P = \{v_0, \ldots, v_{m-1}\}$ and $Q = \{w_0, \ldots, w_{n-1}\}$. Let the complete bipartite graph on bipartition $(\{v_0, \ldots, v_{\ell-1}\}, \{w_0, \ldots, w_{k-1}\})$ be one of the connected components of H'. Recall that $k \leq \frac{n}{2}$ and $\ell \leq \frac{m}{2}$. It follows that $k \leq n - k$ and $\ell \leq m - \ell$. Since $m \cdot k = n \cdot \ell$, we have $k \cdot (m - \ell) = \ell \cdot (n - k)$. By Lemma 6.2, we know that there is a (k, ℓ)-biregular graph on bipartition $(\{v_\ell, \ldots, v_{m-1}\}, \{w_k, \ldots, w_{n-1}\})$. We take such a graph as the rest of H'. Obviously, H' is disconnected. This completes the proof. □

Notation 6.15. *Let $k, \ell \in \mathbb{N}$. For a (k, ℓ)-biregular graph on bipartition (P, Q) we introduce the following notations.*

$$\begin{aligned} P \,\square\, Q &:\iff k = \ell = 0 \\ P \doteq Q &:\iff k = \ell = 1 \\ P \ll Q &:\iff k \geq 2 \text{ and } \ell = 1 \end{aligned}$$

We also write $Q \gg P$ instead of $P \ll Q$.

With this notation, Lemma 6.14 says that a flipped (k, ℓ)-biregular graph on bipartition (P, Q) with $k \geq \ell$ where P and Q are colour classes is identified by C^2 if and only if we have $P \,\square\, Q$, $P \doteq Q$, or $P \ll Q$..

For a graph G with C^2-partition $\pi := \pi_{G,1}$, we define the *skeleton* S_G as the graph with $V(S_G) = \pi$ and $E(S_G) = \big\{\{P, Q\} \,\big|\, P \doteq Q \text{ or } P \ll Q \text{ in the flip of } G\big\}$.

Lemma 6.16. *Let G be a flipped graph which is identified by C^2. Then the following hold.*

1. *There is no path $P_0, P_1, \dots, P_t$ in S_G with $P_0 \ll P_1$ and $P_{t-1} \gg P_t$.*
2. *There is no path $P_0, P_1, \dots, P_t$ in S_G where $P_0 \ll P_1$ and P_t induces a 5-cycle or a matching.*
3. *In every connected component of S_G, there is at most one vertex P that induces a matching or a 5-cycle in G.*

Proof: In the following, we just write G for (G, χ). For each condition, assuming that it does not hold, we will define non-isomorphic graphs H and H', both C^2-equivalent to a subgraph of G that is induced by some classes of the C^2-partition of G. To this end, we consider the pairs of C^2-partition classes of G. By the definition of $\mathcal{I}^2_{\mathsf{C}}$, as long as we maintain the biregularity conditions between these classes, the resulting graph is still C^2-equivalent to G. Therefore, for every pair of C^2-partition classes P and Q of G with $i_{PQ} := |N(v) \cap Q|$ and $i_{QP} := |N(w) \cap P|$ for $v \in P, w \in Q$, we intend to construct edge sets between vertices in P and Q for which the induced graphs on bipartition (P, Q) are still (i_{PQ}, i_{QP})-regular. It suffices to show that, via this approach, two non-isomorphic graphs H and H' can be obtained.

Notation 6.17. *For two C^2-partition classes P and Q, let i_{PQ} denote the number of neighbours that a vertex in P has in Q.*

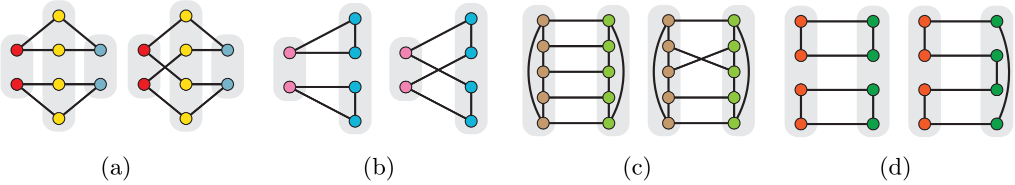


Figure 6.4: Pairs of non-isomorphic graphs illustrating the necessity of the conditions in Lemma 6.16.

We first show Part 1. Assume that $t = 2$ and $G[P_1]$ has no edges. Suppose that, in G, we have $P_0 \ll P_1 \gg P_2$. We may assume $|P_0| \leq |P_2|$. Let $p := i_{P_0 P_1}$ and $q := i_{P_2 P_1}$. Then $p \geq q$.

In the following, Π, Θ and their indexed variants will always denote partitions of P_1 into classes of size p and q, respectively. Furthermore, we denote by $H(\Pi, \Theta)$ the graph on vertex set $P_0 \cup P_1 \cup P_2$ with the following properties.

- For every $i \in \{0, 1, 2\}$ the subgraph induced by P_i has no edges.
- We have $P_0 \ll P_1 \gg P_2$.
- The classes of Π form the neighbourhoods in P_1 of vertices in P_0.
- The classes of Θ form the neighbourhoods in P_1 of vertices in P_2.

First we define H. Because $p \geq q$, there are partitions Π_0 and Θ_0 of P_1 such that there are at most $|P_0| - 1$ classes of Θ_0 that intersect two or more elements of Π_0. Indeed, one way to construct them is to number the vertices in P_1 and then to define the i-th element of Π_0 (and Θ_0) as the set of the i-th p (and q) vertices. We let $H \coloneqq H(\Pi_0, \Theta_0)$.

We now define H'. We know that $|P_0| \geq 2$ since G is flipped. Hence, since $|P_1| = |P_0| \cdot p = |P_2| \cdot q \geq 2 \cdot p$, there is a partition Π_1 of P_1 such that Π_1 has at least two partition classes (all of size p). Moreover, there is a partition Θ_1 of P_1 such that all classes of Θ_1 intersect two or more elements of Π_1. For example, we can define the i-th partition class of Θ_1 to consist of the set of i-th elements of all partition classes of Π_1. We let $H' \coloneqq H(\Pi_1, \Theta_1)$. Since $|P_2| > |P_0| - 1$, this implies $H \ncong H'$ (see Figure 6.4a). This construction can easily be generalised for the case $t > 2$.

For Part 2, choose $t \geq 1$ minimal such that there is a path $P_0, P_1, \dots, P_t$ in S_G satisfying the assumptions. By the minimality of t and Lemma 6.13, all induced graphs $G[P_i]$ for $i \in \{1, \dots, t\}$ are empty and $P_1 \doteq \cdots \doteq P_t$. In particular, it holds that $|P_1| = \cdots = |P_t|$.

First suppose $G[P_1]$ is a 5-cycle. Then also $|P_1| = 5$. Since $P_0 \lll P_1$ and 5 is prime, the set P_0 is a singleton. This contradicts the assumption that G is flipped.

Now suppose $G[P_1]$ be a matching. We let H be the graph $G[P_1] + B$ on vertex set $P_0 \cup \cdots \cup P_1$, where B is a subset of $[P_0, P_1]$ such that we have $P_0 \lll P_1$ in H and, for each $v \in P_0$, the number of edges in $G[P_t]$ that v is connected to via B is as small as possible. (If $i_{P_0 P_1}$ is even, this number is $\frac{i_{P_0 P_1}}{2}$, otherwise it is $\frac{i_{P_0 P_1}+1}{2}$.)

We let H' be the graph $G[P_t] + B'$, where B' is a subset of $[P_0, P_1]$ such that we have $P_0 \lll P_1$ in H' and there is a vertex in P_0 which is connected to $i_{P_0 P_1}$ edges in $G[P_t]$ via B'. Such a B' exists because P_0 cannot be a singleton. Figure 6.4b shows H and H' for $t = 1$ and $|P| = 4$.

We now show Part 3. Choose $t \geq 1$ minimal such that there is a path $P_0, P_1, \dots, P_t$ in S_G for which both P_0 and P_t induce a matching or a 5-cycle in G. By Part 2, we have $P_0 \doteq P_1 \doteq \ldots \doteq P_t$ in G. By Lemma 6.13 and the choice of t, for all $i \in [t-1]$, the edge set of $G[P_i]$ is empty. First, let $G[P_0]$ be a 5-cycle. Then P_t also consists of 5 elements and cannot be a matching. Assume that $G[P_t]$ is a 5-cycle. Let $\widetilde{H}$ be a graph on vertex set $P_0 \dot\cup \ldots \dot\cup P_t$, such that, for every $i \in [t]$, we have $P_{i-1} \doteq P_i$ in $\widetilde{H}$ and the graph $\widetilde{H}[P_i]$ has no edge. Let C be a 5-cycle on P_0. Then there is a unique cycle C' on P_t such that whenever $\{v, v'\}$ is an edge in C and $w, w' \in P_t$ are the two unique vertices connected via paths in $\widetilde{H}$ to v and v', respectively, then $\{w, w'\} \in C'$. Let C'' be a 5-cycle on P_t that is different from C'. We define $H \coloneqq \widetilde{H} + C + C'$ and $H' \coloneqq \widetilde{H} + C + C''$. Then H and H' are non-isomorphic (see Figure 6.4c for $t = 1$).

Assume now that $G[P_0]$ is a matching. Note that $|P_0|$ must be even. Obviously, $G[P_t]$ cannot be a 5-cycle, so assume it is a matching. Let M be a matching on P_0 and let $\widetilde{H}$ be defined as above. Then there is a unique matching M' on P_t such that, whenever $\{v, v'\}$ is an edge in M and $w, w' \in P_t$ are the two unique vertices connected via edges in $\widetilde{H}$ to v and v', respectively, then $\{w, w'\} \in M'$. Let M'' be a matching on P_t that is different from M'. Because G is flipped, $|P_t| = 2$ is impossible. Hence, such a matching M'' exists. We define $H := \widetilde{H} + M + M'$ and $H' := \widetilde{H} + M + M''$ (see Figure 6.4d for $t = 1$ and $|P_0| = 4$). □

Now we have all the ingredients to prove the main theorem of this section.

Theorem 6.18. *A graph is identified by C^2 if and only if its flip is a bouquet forest.*

Proof: We first show the "if"-part. By Lemma 6.12, it suffices to show that every bouquet forest is identified by C^2. This can be done by induction over the maximal size of a bouquet in the forest. The base case is that disjoint unions of arbitrary forests and non-isomorphic vertex-coloured 5-cycles are identified by C^2. Note that, in the definition of a bouquet forest, we do not allow isomorphic vertex-coloured bouquets. (The reason for this is that C^2 cannot distinguish two 5-cycles from a 10-cycle.) For the inductive step, it suffices to observe that a graph is identified by C^2 if all vertex-coloured versions of the graph obtained by removing all leaves of the graph are identified.

For the "only if"-part, let G be a graph identified by C^2 and let χ be the C^2-colouring of G. Let (F, χ) be the flip of G. By Lemma 6.12, we know that (F, χ) is identified by C^2.

By Lemma 6.13, for every colour class P of (F, χ), the graph $F[P]$ either has no edges or is a matching or a 5-cycle.

By Lemma 6.14, for every two distinct colour classes P and Q of (F, χ) we have that $P \mathbin{\square} Q$, $P \doteq Q$ or $P \ll Q$.

We now show that the skeleton S_G is a forest.

First let $P_0, P_1, \ldots, P_t$ with $t \geq 2$ be colour classes of (F, χ) with $P_0 \doteq P_1 \doteq \ldots \doteq P_t$. We claim that $P_0 \doteq P_t$ is impossible. Assume $P_0 \doteq P_t$ for the sake of a contradiction. Note that $|P_0| \neq 1$ because F is flipped. So we have $|P_0| \geq 2$. Then we can arrange the edges in the given matchings as $|P_0|$ disjoint cycles each of size $t + 1$, but also as a single cycle of size $(t + 1) \cdot |P_0|$. This implies that (F, χ) is not identified by C^2. By Lemma 6.12, we can conclude that G is not identified by C^2, a contradiction.

Let $P_0, \ldots, P_t$ be a sequence of colour classes such that, for every $i \in [t]$, we have $P_{i-1} \ll P_i$ or $P_{i-1} \doteq P_i$. Assume that $P_{i-1} \ll P_i$ for at least one $i \in [t]$. Then, by cardinality reasons, it cannot be the case that $P_t \doteq P_0$ or $P_t \ll P_0$.

Also note that if $P_0 \ll \ldots \ll P_t$ with $t \geq 2$, then Part 1 of Lemma 6.16 implies that $P_0 \ll P_t$ is impossible.

We conclude that S_G is a forest.

Furthermore, by Part 3 of Lemma 6.16, in every connected component T of S_G, there can be at most one class $P \in V(T)$ such that $F[P]$ is a matching or a 5-cycle. To see that each connected component that contains a 5-cycle is a bouquet, note that, by Corollary 6.6, all vertices of the 5-cycle must be in the same orbit and thus, the five trees obtained by deleting the 5-cycle must be isomorphic. With Part 2 of Lemma 6.16, it follows that the subgraph of F induced by the vertices of T is either a forest or a bouquet. Moreover, no C^2-partition class of F contains more than one 5-cycle, since this would contradict (F, χ) being identified. Hence, different 5-cycles must have different colours. This implies that there are no isomorphic vertex-coloured bouquets in (F, χ). □

As a nice consequence of the theorem, we obtain an efficient procedure to check whether a graph is identified by the 1-dimensional Weisfeiler-Leman algorithm.

Corollary 6.19. *Given a graph with n vertices and m edges, we can decide whether it is identified by C^2 in time $O((m+n)\log n)$.*

Proof: Colour Refinement can be performed in time $O((m+n)\log n)$ (see [32, 126]). Flipping a graph and checking whether the flip is a bouquet forest can be done in linear time. □

A second corollary of Theorem 6.18 is concerned with vertex colourings of graphs that are identified by C^2.

Corollary 6.20. *Let (G, λ) be a vertex-coloured graph which is identified by C^2 and let λ' be a vertex colouring of G which induces a finer partition on $V(G)$ than λ does. Then (G, λ') is also identified by C^2.*

Proof: Note that the statement is true if G is a bouquet forest. For the general case, we may assume that λ and λ' are equitable. Let (F, λ) be the flip of (G, λ) and let (F', λ') be the flip of (G, λ'). Then the flip of (F, λ') is equal to (F', λ'). To see this, first observe that in F, for every pair of λ-colour classes P, Q, either $[P, Q] \cap E(F)$ or $[P, Q] \setminus E(F)$ equals $[P, Q] \cap E(G)$. The analogous statement holds if P and Q are λ'-colour classes since λ' induces a finer partition of $V(G) = V(F)$ than λ. Thus, flipping (F, λ') results in the same graph as flipping (G, λ'), i.e. in (F', λ').

Since (G, λ) is identified by C^2, by Theorem 6.18, the graph F is a bouquet forest. Hence, (F, λ') is identified and thus, by applying Lemma 6.12, we have that the flip of (F, λ') and therefore the graph (F', λ') is identified. Another application of Lemma 6.12 yields that the vertex-coloured graph (G, λ') is identified. □

In the corollary, the vertex colourings λ and λ' of the graph G can be arbitrary, in particular, λ can be monochromatic. We show in Section 6.4 that, even when λ is required to be monochromatic, this result cannot be generalised to the logics C^k for $k > 2$.

Since 5-cycles and matchings can appear only once per connected component of the skeleton, in this chapter, we call a C^2-partition class that is a 5-cycle or a matching an *exception*. Our classification of finite relational structures that are identified by C^2, which we give in the next section, depends on the structural properties proven about identified graphs in this section. We summarise them for convenience.

Corollary 6.21. *A flipped graph G is identified by C^2 if and only if the following hold:*

1. *Each C^2-partition class induces a graph identified by C^2 (i.e. the induced graph has no edges or it is a matching or a 5-cycle),*
2. *for all C^2-partition classes P and Q, we have $P \mathbin{\square} Q$, $P \doteq Q$, $P \ll Q$, or $Q \ll P$,*
3. *the skeleton S_G is a forest,*
4. *there is no path $P_0, P_1, \ldots, P_t$ in S_G with $P_0 \ll P_1$ and $P_{t-1} \gg P_t$ in G,*
5. *there is no path $P_0, P_1, \ldots, P_t$ in S_G where $P_0 \ll P_1$ and $G[P_t]$ is a 5-cycle or a matching, and*
6. *in every connected component of S_G, there is at most one exception (i.e. a class P that induces a matching or a 5-cycle).*

6.3 Generalisation to Finite Relational Structures

Generalising our observations about graphs, we now classify which finite relational structures are identified by C^2. For every finite relational structure $\mathfrak{A} = (A, R_1, \ldots, R_\ell)$, we define $\mathfrak{A}|_2$, the restriction of $\mathfrak{A}$ to arity 2, to be the relational structure $(A, R'_1, \ldots, R'_\ell)$ with

$$R'_i \coloneqq \{(v_1, \ldots, v_{r_i}) \in R_i \mid \{v_1, \ldots, v_{r_i}\} \text{ has at most 2 elements}\}$$

where r_i is the arity of R_i. Recall the definition of ec-POGs from Subsection 2.5. Since every C^2-formula has at most two free variables, two relational structures $\mathfrak{A}$ and $\mathfrak{B}$ are C^2-equivalent if and only if $\mathfrak{A}|_2$ and $\mathfrak{B}|_2$ are C^2-equivalent. Furthermore, $\mathfrak{A}|_2$ and $\mathfrak{B}|_2$ are C^2-equivalent if and only if the edge-coloured partially oriented graphs ec-POG($\mathfrak{A}|_2$) and ec-POG($\mathfrak{B}|_2$) are C^2-equivalent.

Note that for an equitable partition of an ec-POG, the graph induced by one C^2-partition class is always colour-regular and the graph induced between two C^2-partition classes is colour-biregular.

Outline of the classification. To achieve our classification, we proceed as follows. Since it suffices to analyse which edge-coloured partially oriented graphs (i.e. ec-POGs) are identified, we would essentially like to follow the outline of the proof of the classification for graphs presented in Section 6.2. That is, we first intend to characterise which colour-regular ec-POGs are identified, and then which colour-biregular ec-POGs are identified by C^2. Note that colour-regular ec-POGs are exactly the ec-POGs which have only one C^2-partition class. As we argue later, for an ec-POG that is identified, the underlying undirected graph is also identified. Exploiting this observation, we will first consider only undirected graphs. In our proof for the colour-regular case, we need the undirected colour-biregular case, so we treat it first. We then generalise the results to colour-regular ec-POGs and finally assemble these observations to classify all ec-POGs that are identified by C^2.

Throughout this section, we consider only complete ec-POGs. We can do this without loss of generality, since we can interpret non-edges as edges of a particular colour. The proofs of this section depend on the results from Section 6.2. We employ these results by interpreting uncoloured graphs with edges and non-edges as complete graph with two edge colours.

In agreement with this outline, we now start by analysing colour-biregular ec-POGs that do not have directed edges, i.e, undirected edge-coloured colour-biregular graphs. Recall that $K_{3,3}$ is the complete bipartite graph with two partition classes of size 3.

The following tool is a classical result from set theory.

Theorem 6.22 (Kőnig's Theorem [112])**.** *Every regular bipartite graph contains a perfect matching.*

Lemma 6.23. *Let G be an undirected complete bipartite edge-coloured colour-biregular graph on bipartition (P, Q). Suppose G has at least three edge colours. Then there is a bipartite graph G' on bipartition (P, Q) which is non-isomorphic and C^2-equivalent to G if and only if G is not isomorphic to an edge-coloured $K_{3,3}$.*

Proof: It is straightforward to check that for every regular edge colouring of $K_{3,3}$ there is no C^2-equivalent bipartite non-isomorphic graph. We thus focus on the converse.

Let G be undirected complete edge-coloured colour-biregular on bipartition (P, Q). Suppose $|P| = p \cdot k$ and $|Q| = q \cdot k$ with co-prime p and q (possibly equal to 1).

Let $1 \leq d_1^P \leq d_2^P \leq \cdots \leq d_t^P$ and $1 \leq d_1^Q \leq d_2^Q \leq \cdots \leq d_t^Q$ be the colour degrees of colours $1, \ldots, t$ for vertices in P and Q, respectively. Since $t \geq 3$, it holds that $|P| \geq 3$ and $|Q| \geq 3$. By double-counting, we have that q divides d_i^P and p divides d_i^Q for all $i \in [t]$.

(Case 1: $|P| = |Q|$.) We first consider the case $|P| = |Q|$. We make the following observation. If we construct a colour-regular graph on (P, Q) for some subset of the colours with the correct colour degrees, then by Theorem 6.22, it is always possible to complete the graph such that it has the correct colour degrees for all edge colours. To see this, note that, since $|P| = |Q|$, the graph is in fact colour-regular. It is also bipartite. Now by iteratively applying Kőnig's Theorem, we can use one matching after the other to satisfy the colour degrees.

Assume $d_1^P = d_2^P = 1$ (which implies $d_1^Q = d_2^Q = 1$). If $|P| = 3$, the graph is an edge-coloured $K_{3,3}$. However, if $|P| > 3$, there are two non-isomorphic (not necessarily complete) bipartite graphs with edge colours 1 and 2 and colour degrees d_1^P and d_2^P: we can form a colour-alternating Hamiltonian cycle or two shorter colour-alternating cycles. By Theorem 6.22, these graphs can be extended to satisfy all colour degree conditions and we conclude that, for every bipartite graph with $|P| = |Q| > 3$ and $d_1^P = d_2^P = 1$, there is a C^2-equivalent non-isomorphic graph on the same bipartition. Similarly, by Lemmas 6.12 and 6.14, if $1 < d_2^P < |Q| - 1$, then there are two non-isomorphic (d_2^P, d_2^Q)-biregular graphs on bipartition (P, Q) which can both be extended. Note that, since there are at least three colours, $d_2^P < |Q| - 1$, so this resolves the case $|P| = |Q|$.

(Case 2: $|P| \neq |Q|$.) Suppose now that $|P| \neq |Q|$. Consider a bipartite graph G' on bipartition (P', Q') with $|P'| = |Q'| = |P|/p = |Q|/q = k$ and with colour degrees $\frac{d_1^P}{q}, \dots, \frac{d_t^P}{q}$ and $\frac{d_1^Q}{p}, \dots, \frac{d_t^Q}{p}$. From G', we obtain a graph with bipartition classes of sizes $|P|$ and $|Q|$ by replacing every vertex v in P' with p copies that each have in every colour the same neighbours as v, and then replacing each vertex w in Q' with q copies that each have in every colour the same neighbours as w.

If we perform the construction twice with non-isomorphic graphs G_1' and G_2', then we obtain two non-isomorphic C^2-equivalent graphs G_1 and G_2. We already know from the first part of the proof that such non-isomorphic graphs G_1' and G_2' exist if $k \neq 3$.

It remains to consider the case $k = 3$. Since we have already treated $K_{3,3}$, we can assume that $p > 1$. As above, we will construct two non-isomorphic graphs G_1' and G_2', where the second bipartition class Q' has size $|Q|/q$, that is, $|Q'| = 3$, and then replace every vertex in that class with q copies with the same neighbours in each colour in P. If G_1' and G_2' are non-isomorphic, then the two final graphs are non-isomorphic. Recall that we assume that there are at least three edge colours. Since there are $pk \cdot qk = 3 \cdot pqk$ edges in the complete bipartite graph and since there must be at least $pqk = \operatorname{lcm}(pk, qk)$ edges in a colour, there are exactly three colours, say red, green, and blue, each with pqk edges. Let $Q' = \{w_0, w_1, w_2\}$. Consider the following two graphs G_1' and G_2'. We divide P into three blocks P_0, P_1, P_2 of equal size. We colour all edges between P_i and w_i red, edges between P_i and $w_{(i+1) \bmod 3}$ green and the other edges blue, as to form a blown-up colour-regular bipartite $K_{3,3}$. This yields the graph G_1'. To obtain G_2', we choose, for all $i \in \{0, 1, 2\}$, a vertex v_i

from P_i and recolour its incident red edge green and its incident green edge red. That is, the edges $\{v_i, w_i\}$ are now green, while the edges $\{v_i, w_{(i+1) \bmod 3}\}$ are now red. The newly obtained graph G_2' is non-isomorphic to G_1', since, in G_2', there are vertices that agree on some edge colours towards vertices in Q', but disagree on others. (For example, the edge $\{v_0, w_0\}$ is green just like the edges between vertices in $P_2 \setminus \{v_2\}$ and w_0, but $\{v_0, w_1\}$ is red, whereas P_2 only has blue edges to w_1. Note that $P_2 \setminus \{v_2\}$ is non-empty since $p > 1$. In G_1', for every vertex $v \in P$, the colour of each of its edges to Q' uniquely determines to which block of P it belongs and thus also determines the colours of the other edges incident with v. □

Note that, in the lemma, the bipartition classes are not necessarily C^2-partition classes, i.e. they may not be distinguishable by C^2. However, the equivalence stated in the lemma also holds if P and Q form initial colour classes.

Corollary 6.24. *Let G be an undirected complete bipartite vertex/edge-coloured colour-biregular graph on bipartition (P, Q). Suppose G has at least three edge colours, the vertices in P are coloured red and the vertices in Q are coloured blue. Then there is a bipartite graph G' on bipartition (P, Q) which is non-isomorphic and C^2-equivalent to G if and only if G is not isomorphic to a vertex/edge-coloured $K_{3,3}$.*

Proof: The proof is analogous to the proof of Lemma 6.23. Indeed, in all cases not isomorphic to an edge-coloured $K_{3,3}$, the proof constructs two graphs which are non-isomorphic, but C^2-equivalent to the original graph. In these graphs, we colour the vertices in P red and those in Q blue and we obtain non-isomorphic graphs that are C^2-equivalent to G for the case in which the vertices of G are also coloured in this fashion. □

In the following, we will need both Lemma 6.23 and Corollary 6.24 depending on whether we are in a situation where the bipartition classes P and Q can be interchanged. We first use Lemma 6.23 to characterise colour-regular graphs that are identified by C^2.

Lemma 6.25. *Let G be an undirected colour-regular complete graph with at least three edge colours. Then C^2 identifies G if and only G it is*

1. *a graph on 4 vertices with 3 edge colours that each induce a perfect matching,*
2. *a graph on 6 vertices with 5 edge colours that each induce a perfect matching, or*
3. *a graph on 6 vertices with 3 edge colours, one of which induces the complement of a 6-cycle, and the other two edge colours each induce a perfect matching.*

Proof: Let $1 \leq d_1 \leq d_2 \leq \ldots \leq d_t$ be the colour degrees of G and let $n := |V(G)|$. Since we interpret non-edges as edges of a particular colour, without loss of generality, we can assume that $\sum_{i=1}^{t} d_i = n - 1$.

We distinguish two cases according to the parity of the number n of vertices.

(Case 1: n is odd.)

Suppose first that n is odd. Then all colour degrees must be even. Consequently, the circulant construction from Subsection 6.1 can be adapted to perform the inversion for coloured edges. In more detail, for every map $\psi\colon [n-1] \to [t]$ such that, for all i, we have $\psi(n-i) = \psi(i)$ and, for all j we have $|\psi^{-1}(j)| = d_j$, we obtain a circulant graph $\mathrm{Circ}(\psi)$ by colouring the edge between vertices i and j satisfying $j \leq i$ with the colour $\psi(i-j)$. This can also be interpreted as a 2-factorisation of the graph into $\frac{n-1}{2}$ 2-factors. The map ψ then dictates that the edges of the ℓ-th 2-factor are to be coloured with colour $\psi(\ell)$.

Since there are at least three colours, we have $1 < d_1 \leq \frac{n-1}{3}$. This also implies $n \geq 7$.

There are suitable maps ψ, ψ' such that $\psi^{-1}(1) = \{-\frac{d_1}{2}, \dots, -1, 1, \dots, \frac{d_1}{2}\}$ and $(\psi')^{-1}(1) = \{-\frac{d_1}{2} - 1, \dots, -2, 2, \dots, \frac{d_1}{2} + 1\}$ (where we take numbers modulo n). We claim that $\mathrm{Circ}(\psi)$ and $\mathrm{Circ}(\psi')$ are not isomorphic. Indeed, consider vertex 0. In the subgraph of $\mathrm{Circ}(\psi)$ induced by the edges of colour 1, this vertex has two neighbours that both have $d_1 - 2$ common neighbours with it (namely the neighbours 1 and $n-1$). However, since $d_1 \leq \frac{n-1}{3}$, in $\mathrm{Circ}(\psi')$ the vertex 0 does not have such a neighbour with respect to the edge colour 1. Since the graph is vertex-transitive, it suffices to consider only vertex 0, and we conclude that the graphs are non-isomorphic.

(Case 2: n is even.)

Suppose now that n is even. Note first that all colour-regular graphs on at most four vertices with at least three edge colours have exactly four vertices and are unions of three perfect matchings and isomorphic if the sets of edge colours are equal. We can thus assume in the following that $n \geq 6$.

For every integer i define $\mathrm{even}(d_i)$ to be the largest even integer that is at most d_i. We distinguish two cases, depending on the following inequality, which we call the *sum condition.* We say that *the sum condition is fulfilled* if $\sum_{i=2}^{t} \mathrm{even}(d_i) \geq \frac{n}{2} - 1$. Note that this summation starts at index 2.

(Case 2.1: The sum condition is fulfilled.)

We argue that if the sum condition is fulfilled, then there exists a graph G_H on n vertices indistinguishable from G with the following properties.

- The vertex set V of G_H is the union of two sets V_1 and V_2, each of size $\frac{n}{2}$.
- The two graphs $G_H[V_1]$ and $G_H[V_2]$ are isomorphic, colour-regular graphs.
- The bipartite graph H induced by all edges that run between V_1 and V_2 is colour-regular and contains all edges of colour 1.

To see this, we first apply the circulant construction to obtain two complete isomorphic graphs which are for all $i \in [2, t]$ at most even(d_i)-regular with respect to colour i and 0-regular for colour 1. To satisfy the remaining edge colour restrictions, we can apply Theorem 6.22 repeatedly to the bipartite graph induced by the edges between V_1 and V_2 (treating V_1 and V_2 as independent sets). In particular, note that, since the sum condition is fulfilled, it holds that $d_1 = n - 1 - \sum_{i=2}^{t} d_i \leq n - 1 - (\frac{n}{2} - 1) \leq \frac{n}{2}$.

We will, depending on the colour degrees, construct the graph H as to allow us to also construct a different C^2-equivalent graph H' that can replace H to obtain a non-isomorphic graph.

(Case 2.1.1: $d_1 > 1$.) If $d_1 > 1$, then the graph H can be constructed in such a way that the graph induced by the edges in colour 1 is connected (which can be seen by starting with a Hamiltonian cycle and then applying Theorem 6.22 as in the previous proof). There are at least three colours and d_1 is the smallest colour degree, so we have $d_1 \leq \frac{n-1}{3}$. Since $n \geq 6$, this implies that $d_1 < \frac{n}{2} - 1$ and thus, the graph induced by the edges of colour 1 is neither a matching nor a co-matching. Lemma 6.14 implies that there is a bipartite graph H' that is C^2-equivalent but not isomorphic to H. We can let $G_{H'}$ be the graph obtained from G_H by replacing H with H'.

(Case 2.1.2: $d_1 = d_2 = 1$.) If $d_1 = d_2 = 1$, we can construct G_H such that all edges of colours 1 and 2 are contained in H. We repeat the above argument, this time constructing a bipartite graph H for which the subgraph induced by the edges of colour 1 and the edges of colour 2 is connected. According to Lemma 6.23, there is exactly one size for which there exists no non-isomorphic graph H' that is C^2-equivalent to H, namely the case $n = 6$. Note that in this case, since the sum condition is fulfilled, we have $d_3 = 3$ (which implies $t = 3$), or it holds that $d_3 = 1$ and $d_4 = 2$ (and $t = 4$). In the first case, since two disjoint matchings always form a union of cycles of even length, for every graph with these degrees, the graph induced by the third colour must be the complement of a 6-cycle and thus, the graph G is identified by C^2. In the second case, two non-isomorphic C^2-equivalent graphs are obtained by forming three matchings and two 3-cycles and by forming three matchings and a 6-cycle.

(Case 2.1.3: $d_1 = 1$ and $d_2 > 1$.) Supposing now that $d_1 = 1$ and $d_2 > 1$, consider first the case $n = 6$, which implies $d_2 = d_3 = 2$. A graph with those colour degrees is not identified, since it is possible to let colour d_2 induce two triangles, but also possible to let it induce a 6-cycle. Supposing $n > 6$, one can choose the graphs induced by V_1 and V_2 so as to have at least two edge colours. Indeed, since there are at least three edge colours and $d_3 \geq 2$, we can take a number d'_2 with $1 \leq d'_2 \leq d_2$ and let the induced colour-regular graphs on V_1 and V_2 be at least d'_2-regular in colour 2 and at least 2-regular in colour 3, for example, by using the circulant construction from Subsection 6.1. The graph H contains a perfect matching of colour 1. Note that this does not necessarily mean that the bipartition (V_1, V_2) is

combinatorially determined by the graph. Since the induced graphs on V_1 and V_2 are isomorphic, the matching can be chosen such that, whenever (v_1, v_2) and (v_1', v_2') are distinct edges from V_1 to V_2 of colour 1, then the colour of (v_1, v_1') is equal to the colour of (v_2, v_2'). However, since $G_H[V_2]$ has at least two edge colours, by suitably changing the matching, we can destroy this property (call it the *X-property*), maintaining the C^2-equivalence class. Note that we can even maintain the isomorphism type of H. Indeed, pick a vertex $v_1 \in V_1$ such that changing the neighbour of v_1 with respect to colour 1 from a vertex v_2 to a vertex v_2' destroys the X-property. Then we can restore the isomorphism type of H by swapping the roles of v_2 and v_2' in all edges of H (i.e. by replacing v_2 with v_2' and vice versa).

We claim that this alteration produces a non-isomorphic graph. We argue this by showing that the number of 4-tuples (v_1, v_2, v_1', v_2') of distinct vertices such that $\{v_1, v_2\}$ and $\{v_1', v_2'\}$ are matching edges and $\{v_1, v_1'\}$ has the same colour as $\{v_2, v_2'\}$ changes. Note that the statement is true for 4-tuples for which v_1 and v_1' can be assumed to come from V_1 (or, by symmetry, if both come from V_2). However, since the bipartition into V_1 and V_2 might not be isomorphism-invariant, in addition to the edges considered above, we also need to consider the 4-tuples for which v_1 and v_1' come “from different sides”, i.e. one is contained in V_1 and the other in V_2. However, for such 4-tuples, the edges $\{v_1, v_1'\}$ and $\{v_2, v_2'\}$, whose colours we need to compare, are contained in H. Thus, since we have rearranged the matching in colour 1 while maintaining the isomorphism type of H, this number does not change. of H. Altogether, we have thus created two C^2-equivalent graphs which differ in the number of pairs of matching edges of colour 1 whose starting vertices are connected by the same colour as the end vertices. Thus, the two graphs cannot be isomorphic.

(Case 2.2: The sum condition is not fulfilled.)

Suppose now that the sum condition is not fulfilled. This implies that $d_1 = d_2 = 1$. Indeed, suppose otherwise. If $d_3 = 2$, then $\text{even}(d_2) + \text{even}(d_3) = 4 > \frac{2+2+2+1}{2} - 1 \geq \frac{d_1+d_2+d_3+1}{2} - 1$. If $d_3 \geq 3$, then also $\text{even}(d_2) + \text{even}(d_3) \geq (d_2 - 1) + (d_3 - 1) = \frac{2d_2+2d_3-2}{2} - 1 \geq \frac{d_1+d_2+d_3+1}{2} - 1$. In both cases, it additionally holds that $\text{even}(d_i) \geq \frac{d_i}{2}$ for all $i \geq 3$. Summing up all the $\text{even}(d_i)$ for $i \geq 2$, the sum condition is fulfilled. Also, if there are at most four colours, then $\sum_{i=2}^{t} \text{even}(d_i) \geq n - 4$. This is easy to see when $t = 3$. If $t = 4$, since $n - 1 - d_1 - d_2 = n - 3$ is odd, exactly one of d_3 and d_4 must be even. Thus, $\text{even}(d_3) + \text{even}(d_4) = d_3 + d_4 - 1 = n - 4$. Altogether, since we have assumed $n \geq 6$, the sum condition is fulfilled if $t \leq 4$.

We can thus assume that there are at least 5 colours and that $d_1 = d_2 = 1$. Via the 1-factorisation construction due to Walecki described in detail in [105, Section 3.2], every map $\psi\colon [n-1] \to [t]$ leads to an edge-coloured colour-regular graph $\text{Match}(\psi)$ with the property that two matchings of the obtained 1-factorisation always yield a Hamiltonian cycle. Also, if ψ_1 and ψ_2 are two distinct maps, then the graphs $\text{Match}(\psi_1)$ and $\text{Match}(\psi_2)$ are not identical (but may be isomorphic).

We call ψ *valid* if $|\psi^{-1}(i)| = d_i$ holds for every $i \in \{1, \dots, t\}$. We will in the following consider only valid maps ψ that satisfy $\psi(1) = 1$ and $\psi(2) = 2$, additionally. There are $(n-1)-2 = n-3$ possible elements in $\psi^{-1}(3)$ and $n-3-d_3$ possible elements in $\psi^{-1}(4)$. Also, since $|\psi^{-1}(5)| \geq 1$, it holds that $n-3-d_3 > d_4$. Thus, there are at least $\binom{n-3}{d_3}\binom{n-3-d_3}{d_4} \geq (n-3) \cdot 2$ distinct valid choices for ψ.

Since there are only n possible automorphisms of a colour-alternating Hamiltonian cycle on n vertices, there are also only n possible automorphisms of the graph induced by the colour-alternating Hamiltonian cycle formed by the first and second matching (recall that they form a Hamiltonian cycle). Hence, if all graphs resulting from the choices for ψ were isomorphic, then we would have $(n-3) \cdot 2 \leq n$, which implies $n \leq 6$. We already considered $n \leq 4$, so suppose $n = 6$. In this case, since there must be 5 colours, $d_1 = d_2 = d_3 = d_4 = d_5 = 1$. We thus obtain a 1-factorisation of K_6, of which there is only one up to isomorphism [121], so the graph is identified by C^2. □

So far we have only dealt with undirected graphs. We now extend the previous lemma to colour-regular complete ec-POGs (see Figure 6.5). Recall that ec-POGs have the property that there is no edge colour with both directed and undirected edges.

Theorem 6.26. *Let G be a colour-regular complete* ec-POG*. Then C^2 identifies G if and only if G is*

1. *an undirected complete graph with only one edge colour,*
2. *undirected and has two edge colours, one of which induces a perfect matching,*
3. *a directed 3-cycle,*
4. *an undirected graph on four vertices, which has three edge colours that each induce a perfect matching,*
5. *a graph on four vertices and has two edge colours, one of which induces a 4-cycle that may or may not be directed, and the other edge colour induces a perfect matching,*
6. *the unique regular tournament on five vertices,*
7. *a graph on five vertices and has two edge colours which both induce a* 5*-cycle of which at most one is directed,*
8. *an undirected graph on six vertices that has five edge colours which each induce a perfect matching, or*
9. *an undirected graph on six vertices, which has three edge colours, one of which induces the complement of a* 6*-cycle, and the other two edge colours each induce a perfect matching.*

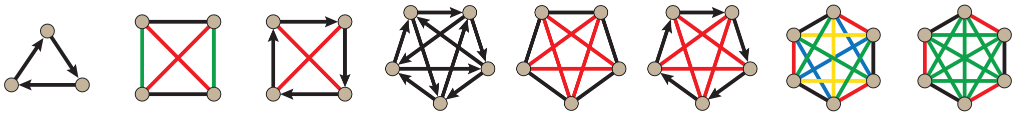


Figure 6.5: The special cases (Parts 3–9) that occur in the classification in Theorem 6.26. Together with the family of graphs from Part 2, they form the *exceptions*.

Proof: Let G be an edge-coloured colour-regular complete ec-POG.

Let $\overline{G}$ be the graph that is obtained from G by replacing every directed edge with an undirected edge of the same colour. We claim that if G is identified by C^2, then so is $\overline{G}$. Indeed, assume there is a non-isomorphic graph $\overline{H}$ not distinguished from $\overline{G}$ by C^2. From $\overline{H}$, we construct an ec-POG H that is C^2-equivalent to G, but not isomorphic to G. To this end, in $\overline{H}$, we leave all edges unchanged whose colour corresponds to an edge colour class consisting of undirected edges in G. Consider now an edge colour class C which, in G, consists of directed edges. The induced graph in G is a union of edge-disjoint directed cycles, i.e. the vertices have equal in- and out-degrees. Thus, in $\overline{G}$, the induced graph is regular of even degree or, equivalently, a union of edge-disjoint 2-factors. By the C^2-equivalence of $\overline{G}$ and $\overline{H}$, the same must hold in $\overline{H}$. By also decomposing the induced graph into edge-disjoint cycles, directing each of these, and proceeding like this for all remaining edge colour classes, we obtain a colour-regular ec-POG H that is C^2-equivalent to G but not isomorphic to it.

To prove the theorem, it thus remains to consider graphs whose undirected underlying graph is identified by C^2. We distinguish two cases according to the number of edge colours.

(Case 1: At most 2 edge colours.) Starting with graphs that have at most two edge colours, we consider all undirected graphs that appear in Lemma 6.13 (with non-edges interpreted as edges of a distinct colour). For graphs with only one edge colour, the question is for which n there exists, up to isomorphism, exactly one regular tournament on n vertices. This is the case for $n \in \{1, 3, 5\}$. Indeed, to see that for $n > 5$ there is no unique regular tournament, it suffices to consider the circulant directed graph in which there is an edge from i to j if $(i - j) \bmod n < \frac{n}{2}$. Reversing all edges of the form $(i, i+1)$ yields a non-isomorphic regular tournament.

Next we consider graphs that consist of a matching and the complement of a matching. Suppose the matching is of colour red and the complement is blue. Since a matching induces a graph of odd degree, it cannot be directed while maintaining regularity. If $n = 4$, then, by directing the complement of the matching, we obtain the graph described in Part 5. Note that n is even, so suppose $n > 5$ from now on. To see that G is not identified, we form two non-isomorphic graphs $\widehat{G}$ and $\widehat{G'}$ on $\{v_1, \dots, v_{\frac{n}{2}}\} \cup \{w_1, \dots, w_{\frac{n}{2}}\}$ that are C^2-equivalent to G as follows. We start with an arbitrary directed complete graph H on vertex set $\{1, \dots, \frac{n}{2}\}$ that contains the edges $(1,2)$, $(2,3)$, and $(3,1)$. In $\widehat{G}$, each pair $\{v_i, w_i\}$ forms an edge of the red matching. For every directed edge (i,j) in H, we insert the blue directed edges (v_i, v_j), (v_j, w_i), (w_i, w_j), and (w_j, v_i). This yields a graph $\widehat{G}$ which consists of a red matching and a blue directed complement of a matching. Note that, in $\widehat{G}$, every 4-tuple of vertices that induce two red matching edges also induces a directed 4-cycle in blue. Also, note that $\widehat{G}$ contains a blue directed 3-cycle C_3 with vertices v_1, v_2, v_3. We form a new graph $\widehat{G'}$ by reversing all edges in C_3. The graph $\widehat{G'}$ is C^2-equivalent to $\widehat{G}$, but it is not isomorphic to $\widehat{G}$, since the 4-tuple of vertices (v_1, v_2, w_1, w_2) induces two red matching edges but no blue directed 4-cycle in $\widehat{G'}$.

Consider now a 5-cycle. Interpreted as an edge-coloured complete graph, it has two edge colours, both of which induce a 5-cycle. If we direct one of the 5-cycles, we obtain a graph described in Part 7. It is easy to see that these graphs are identified by C^2, since reversing all edges of the 5-cycle yields an isomorphic graph. However, directing both 5-cycles, i.e. directing edges of both edge colours, we obtain a graph that is not identified by C^2, since it is not isomorphic to the graph obtained when reversing the directions in one of the 5-cycles: only one of the graphs has a directed 3-cycle with two edges in the first colour.

(Case 2: At least three edge colours.) Concerning graphs with three or more edge colours, we need to consider all graphs mentioned in Lemma 6.25. Since matchings cannot be oriented while maintaining regularity, it suffices to consider graphs on 6 vertices that are the disjoint union of the complement of a 6-cycle and two perfect matchings. However, for the edge colour class that is the complement of a 6-cycle, the degree is odd, so it cannot be oriented in a regular way either. We thus obtain only the undirected versions of the graphs mentioned in Lemma 6.25. □

We are going to determine the structure of ec-POGs that are identified by C^2. So far, we have classified the ec-POGs that have exactly one C^2-partition class. Similarly to the case of undirected graphs treated in Section 6.2, we will now describe how to combine such building blocks to form a larger graph that is identified. As before, by interpreting non-edges as edges of a special colour, we only need to consider complete graphs. Since vertices of different C^2-partition classes are distinguished, we can assume that all edges between two C^2-partition classes are undirected. (In other words, the direction of a directed edge starting in a colour class P and ending in another colour class Q is implied by the colours of the start and end vertices. Edges of the same colour in the other direction can thus always be marked with a different colour.) Recall that Corollary 6.24 says that the only regular graph with a fixed bipartition and at least three edge colours that is colour-biregular is $K_{3,3}$ with an edge colouring that induces three disjoint perfect matchings. This shows that it is possible that two C^2-partition classes of size 3 in an identified graph can be connected via three matchings of different colours. In the light of this, in comparison to the case of two colours covered in Notation 6.15, we need an additional relation.

Definition 6.27. Let G be a vertex-coloured ec-POG and let P and Q be two disjoint subsets of $V(G)$. We introduce the relation $P \equiv_3^3 Q$ to denote the fact that the graph induced by the edges running between P and Q is $K_{3,3}$ with three edge colours, each of which induces a perfect matching between P and Q.

Since we treat non-edges as edges of a particular colour, we also need to adapt our notation which we used for graphs (see Notation 6.15) so that it becomes applicable to graphs with two edge colours. Suppose that, between two distinct C^2-partition classes P and Q, there are edges of two colours, say red and blue. To define the relations $P \,\Box\, Q$, $P \doteq Q$, $P \ll Q$, we always consider the graph induced by the edge colour class that contains fewer edges and ignore orientations. It is not difficult to see that if the number of red edges is equal to the number of blue edges then choosing either induced graph yields the same results. Note that, with this convention, the relations $P \,\Box\, Q$, $P \doteq Q$, $P \ll Q$ in particular imply that there are only at most two colours among the edges running between P and Q.

From Corollary 6.24, together with the discussion following Lemma 6.14 in Section 6.2, we obtain the following.

Corollary 6.28. *Let G be a vertex/edge-coloured undirected graph that is identified by C^2. If P and Q are distinct C^2-partition classes of G, then $P \,\Box\, Q$, $P \doteq Q$, $P \ll Q$, $Q \ll P$, or $P \equiv_3^3 Q$.*

Theorem 6.26 implies that each C^2-partition class of an ec-POG that is identified must be one of the 9 listed types. In an identified ec-POG, we call every C^2-partition class which does not induce an undirected complete graph with only one edge colour an *exception* (i.e. any class that does not fall under Part 1 of Theorem 6.26). Similarly, we call every pair of C^2-partition classes P and Q for which $P \equiv_3^3 Q$ holds an *exception.*

Note that, with this terminology, Lemma 6.16 says that for uncoloured, undirected graphs, each connected component of the skeleton can induce at most one exception in the original graph. Here, in accordance with Lemma 6.13, the possible exceptions are Part 2 and the undirected option in Part 7 from Theorem 6.26. As we will see, a similar observation about exceptions only occurring once per connected component of the skeleton is true for edge-coloured graphs.

Since in an edge-coloured graph it is not clear what a flip is, we need to adjust the approach from Section 6.2 slightly. Let G be a vertex/edge-coloured graph. As before, we define the vertices of the *skeleton* S_G to be the C^2-partition classes of G. Two distinct vertices P, Q in S_G are adjacent in S_G if the corresponding classes in G do not satisfy $P \,\Box\, Q$, i.e. whenever P and Q are not monochromatically connected.

Concerning the properties of identified structures, we obtain a theorem similar to Corollary 6.21 for graphs.

Theorem 6.29. *Let G be a vertex-coloured* ec-POG *(i.e. a vertex/edge-coloured partially oriented graph). Then G is identified by C^2 if and only if the following hold.*

1. *Each C^2-partition class induces a graph identified by C^2 (i.e. the induced graph is one of the graphs mentioned in Theorem 6.26).*
2. *For all C^2-partition classes P and Q, we have $P \square Q$, $P \doteq Q$, $P \ll Q$, $Q \ll P$, or $P \equiv^3_3 Q$.*
3. *The skeleton S_G is a forest.*
4. *There is no path $P_0, P_1, \dots, P_t$ in S_G with $P_0 \ll P_1$ and $P_{t-1} \gg P_t$ in G.*
5. *There is no path $P_0, P_1, \dots, P_t$ in S_G where $P_0 \ll P_1$ and P_t is an exception.*
6. *In every connected component of S_G, there is at most one exception (i.e. either two C^2-partition classes P and Q satisfying $P \equiv^3_3 Q$ or a C^2-partition class among Parts 2–9 described in Theorem 6.26).*

Proof: For the "if"-part, let G first be a vertex-coloured ec-POG that satisfies the given conditions. It suffices to show that for each connected component in the skeleton S_G, the graph induced by its collection of C^2-partition classes is identified by C^2. For such an induced graph, by Part 1, we can assume that it has at least two C^2-partition classes. Also, by Part 3, the skeleton is a tree, i.e. it contains no cycles. The skeleton has at least two leaves and if the entire connected component is not an exception of type $P \equiv^3_3 Q$, then by Part 6, at most one of the leaves is an exception. Therefore, the connected component contains a non-exceptional C^2-partition class P (a leaf of the skeleton) such that for exactly one class $Q \neq P$, we have $P \ll Q$ or $P \doteq Q$, and for all other C^2-partition classes $Q' \neq Q$, it holds that $P \square Q'$. Also, since P itself is not an exception, it is of Type 1 in Theorem 6.26 and thus, also for $Q' = P$, it holds that $Q' \square P$. Moreover, if $P \ll Q$, then by Part 5, $G[Q]$ is a clique or empty. It is easy to see that with all these properties, the connected component is identified if and only if the connected component obtained by removing P is identified. It thus suffices to show that all exceptions are identified. This is shown in Corollary 6.24 and Theorem 6.26.

For the "only if"-part, let G be a vertex-coloured ec-POG identified by C^2. Note that the vertex colouring of G can only induce a finer C^2-partition than a monochromatic vertex colouring. Corollary 6.21 and Theorem 6.26 imply Part 1. Corollary 6.21 and Corollary 6.24 imply Part 2.

Note the following for two C^2-partition classes P and Q of an identified graph that satisfy $P \equiv^3_3 Q$. If we merge two of the three edge colour classes into a single one to obtain a graph G', then the entire graph must still be identified by C^2. The reason is that if two classes P and Q of size 3 satisfy $P \doteq Q$, then the larger edge colour class can always be split so that $P \equiv_3^3 Q$. Similarly, if we merge different edge colour classes within a C^2-partition class P or remove directions from some colour classes of edges within P so that the graph induced by P is still identified, then the new entire graph is still identified by C^2.

In particular, we can replace every C^2-partition class with an undirected complete graph in the corresponding C^2-colour while maintaining that the graph is identified.

With this in mind, Parts 3–5 follow immediately from Corollary 6.21.

To prove Part 6, we first ignore the $\equiv_3^3$ exceptions. Note that by Parts 4 and 5, exceptions of the other types that occur in the same connected component of S_G must have the same size. Moreover, these exceptions must be connected by a path $P_0, P_1, \dots, P_t$ such that we have $P_0 \doteq P_1 \doteq \dots \doteq P_t$ in G. Suppose that, in such a situation, P_0 and P_t each induce an exception. Note that for all such exceptions apart from the tournaments on 3 and 5 vertices, it is possible to remove directions and merge colours in order to obtain a matching or a 5-cycle. It follows from Corollary 6.24 that such exceptions cannot occur in the same connected component of the skeleton. If P_0 and P_t are directed 3-cycles, then we obtain a non-isomorphic graph by reversing the directions in P_t. Similarly, if P_0 is the regular tournament on 5 vertices and P_t is an arbitrary exception on 5 vertices with some directed edges, then we obtain a non-isomorphic graph by reversing all edges in P_0. If P_t is an exception on 5 vertices with no directed edges, then P_t must be the exception that has two edge colours which each induce an undirected 5-cycle. Since P_0 contains 5-cycles of the underlying undirected graph for which the edges form a directed cycle and 5-cycles for which the edges do not form a directed 5-cycles, by permuting the vertices in P_t, we can construct a non-isomorphic graph (by either choosing to match a 5-cycle of P_t with a directed 5-cycle of P_0 or with a 5-cycle whose edges do not form a consistent cyclic orientation).

Suppose finally that an exception of the type $\equiv_3^3$ occurs in a connected component of S_G. Then by merging colours and by Parts 4 and 5, the only types of exceptions that can occur in the connected component are other exceptions of type $\equiv_3^3$ and directed 3-cycles. Observe that an exception of type $P \equiv_3^3 Q$ induces a cyclic order on both P and Q: two of the coloured matchings form a Hamiltonian cycle on $P \cup Q$ and the cyclic order of the vertices in the Hamiltonian cycle is automorphism-invariant. Thus, by swapping two edge colours of the $\equiv_3^3$-exception, we obtain a non-isomorphic graph whenever there is a second exception. □

Corollary 6.30. *Given a vertex-coloured* ec-POG *on n vertices, we can decide whether it is identified by C^2 in time $O(n^2 \log n)$.*

Proof: For this, we compute the coarsest equitable partition in time $O(n^2 \log n)$ using Colour Refinement. We then compute the skeleton of the obtained graph. For each C^2-partition class, we check whether the induced graph appears among the exceptions from Theorem 6.26. Since the non-trivial exceptions have at most 6 vertices, this can be performed in linear time. We also check whether each pair of distinct C^2-partition classes satisfies one of the relations required by Corollary 6.28. We mark each vertex or edge of the skeleton that corresponds to an exception. Finally, we check whether the skeleton is a tree, whether each connected component has at most one exception, and whether the sizes of the C^2-partition classes in each connected component increase monotonically when starting from a smallest class (for example, using depth first search). We also check whether every exception is a smallest class of its connected component. □

Since we can reduce the problem of deciding whether a finite relational structure is identified by C^2 to the problem of deciding whether an ec-POG is identified by C^2, we obtain the following corollary.

Corollary 6.31. *Given a finite relational structure $\mathfrak{A}$ with a universe of size n over a fixed signature, it can be decided in time $O(n^2 \log n)$ whether it is identified by C^2.*

Proof: Since the logic C^2 can only use two variables, if $n \geq 3$ and $\mathfrak{A}$ has a relation of arity at least 3, then it is not identified. For a fixed signature and a given structure $\mathfrak{A}$ over this signature, the graph ec-POG($\mathfrak{A}$) can be constructed in time $O(n^2)$. It suffices now to observe that a relational structure $\mathfrak{A}$ with relations of arity at most 2 is identified if and only if ec-POG($\mathfrak{A}$) is identified. □

Using the classification, we also obtain an extension of Corollary 6.20 to ec-POGs and, more generally, to finite relational structures.

Corollary 6.32. *Let G be an* ec-POG *and let λ be a vertex colouring of G such that (G, λ) is identified by C^2 and let λ' be a vertex colouring of G which induces a finer partition on $V(G)$ than λ does. Then (G, λ') is also identified by C^2.*

Proof: By induction, it suffices to consider a vertex colouring that refines exactly one C^2-partition class of G. For our analysis, we use the classification of C^2-partition classes of identified graphs given in Theorem 6.26. Consider first a partition class of Type 1 in the theorem. A vertex-coloured version of an undirected complete graph cannot have an exception since each colour class is still a complete graph. A partition class of Type 2 is a perfect matching. If we consider the C^2-partition classes obtained after colouring the vertices in the matching arbitrarily, then no new exceptions arise. Specifically, in the graph with the refined colouring, all exceptional C^2-partition classes are matchings. However, if by the recolouring, matching edges from C end up in different partition classes, then those cannot be in the same connected component of the skeleton. Thus, each connected component of the skeleton of the new graph induces at most one exception (which must be a matching).

Next we argue that every vertex-coloured version of a graph described in Parts 3–9 of Theorem 6.26 is identified and has at most one exception. This is easy to see for all described graphs up to 5 vertices since otherwise the graph would have to decompose into 2 or more smaller vertex-disjoint exceptions. However, this is not possible since exceptions have at least 3 vertices. For the mentioned graphs on 6 vertices, consider first the one that consists of 5 disjoint matchings. It is not difficult to see that this graph only has two equitable partitions, the discrete partition and the trivial partition. Consider now the graph on 6 vertices that consists of two perfect matchings and a 3-regular graph. Every equitable partition with a singleton for this graph is discrete. There are thus two types of equitable partitions to consider, namely equitable partitions with three classes of size 2 and equitable partitions with two classes of size 3. In the latter case, the graph induces an edge-coloured version of $K_{3,3}$, and in the former case, the graph has at least two classes P_1 and P_2 such that $P_1 \,\square\, P_2$ (since each of the two matchings can only appear in one biregular connection between two classes). Thus, the new C^2-partition classes of C induce a forest in the skeleton and therefore, the skeleton of G with the refined colouring is still a forest. In any case, the graph is identified and has at most one exception. This proves that Properties 1 and 6 of Theorem 6.29 are preserved.

To show globally that a colour-refined version of an identified ec-POG is also identified, note that also all other properties required by Theorem 6.29 are preserved, since (by Corollaries 6.20 and 6.21) these properties are preserved if all C^2-partition classes are replaced with undirected complete graphs. □

Corollary 6.33. *If a finite relational structure $\mathfrak{A}$ is identified by C^2, then every finite relational structure obtained from $\mathfrak{A}$ by adding unary relations is also identified by C^2.*

Proof: Since $\mathfrak{A}$ is identified, it has at most 2 elements in its universe or there is no relation of arity at least 3. Recall that under this circumstance, $\mathfrak{A}$ is identified if and only if ec-POG($\mathfrak{A}$) is identified. The addition of unary relations corresponds to a refinement of the vertex colouring of ec-POG($\mathfrak{A}$), so the corollary follows from the previous one. □

6.4 Higher Dimensions

It is very natural to ask whether our results can be extended to the logics C^k for $k > 2$, i.e. to higher dimensions of the Weisfeiler-Leman algorithm than dimension 1. Chang [34] and Hoffman [79] showed that triangular graphs on sufficiently many vertices are identified by C^3. They form an infinite family of strongly regular graphs, which shows that for $k > 2$, any classification result for graphs identified by C^k must include an infinite number of non-trivial graphs. Also, since every uncoloured strongly regular graph is stable with respect to the 2-dimensional Weisfeiler-Leman algorithm, a classification for graphs identified by C^3 would have to solve the open question which strongly regular graphs are uniquely determined by their parameters.

This indicates that the situation for $k = 2$ is special. We first show that statements analogous to Corollaries 6.6 and 6.20 do not hold for higher dimensions. We use the construction from [30, Section 6], which provides us with the following fact.

Fact 6.34 (Cai, Fürer, Immerman (cf. [30])**.** *For every $k > 2$, there are non-isomorphic 3-regular graphs G and G' of size $O(k)$ on the same vertex set V, which contains vertices a, b, a', b' with the following properties: the graphs are not distinguishable by C^k and the edge sets satisfy $E(G) \setminus E(G') = \{\{a, a'\}, \{b, b'\}\}$ and $E(G') \setminus E(G) = \{\{a, b'\}, \{b, a'\}\}$. Moreover, we can guarantee that every graph that is C^3-equivalent to G and G' is isomorphic to G or G'.*

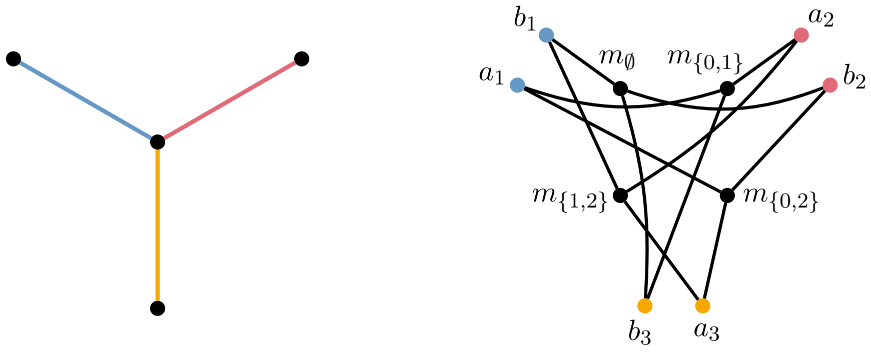


Figure 6.6: A vertex with three incident edges (left) and the corresponding gadget X_3 (right). The colours of a pair $\{a, b\}$ indicate which original edge it is associated with.

The graphs G and G' referred to in the fact are obtained by replacing every vertex v of a suitable connected graph H with a copy of a gadget that only depends on the degree of v (see Figure 6.6). To be more precise, the gadget $X_d := (V_d, E_d)$ for a vertex of degree d has the vertex set $V_d := A_d \cup B_d \cup M_d$, where $A_d = \{a_i \mid i \in [d]\}$ and $B_d = \{b_i \mid i \in [d]\}$ are two disjoint sets of so-called *outer vertices*, and $M_d = \{m_S \mid S \subseteq [d], |S| = 0 \mod 2\}$ is the set of *middle vertices*. The edge set is $E_d = \{(m_S, a_i) \mid i \in S\} \cup \{(m_S, b_i) \mid i \notin S\}$.

In a bijection, we assign to each original edge $\{v, v'\} \in E(H)$ one of the pairs $\{a_i, b_i\}$ from the gadget for v, call it $\{a_{vv'}, b_{vv'}\}$. We proceed similarly with one of the pairs $\{a_j, b_j\}$ from the gadget for v', call it $\{a_{v'v}, b_{v'v}\}$.

To obtain G, we insert for every original edge $\{v, v'\} \in E(H)$ the edges $\{a_{vv'}, a_{v'v}\}$ and $\{b_{vv'}, b_{v'v}\}$, the *parallel* pairs of edges. To obtain $\widetilde{G}$, we select a single arbitrary original edge $\{v, v'\} \in E(H)$ and *twist* it in G. That is, we remove the parallel edges associated with $\{v, v'\}$ and insert the edges $\{a_{vv'}, b_{v'v}\}$ and $\{b_{vv'}, a_{v'v}\}$ instead. Accordingly, we call these pairs of edges *twisted*.

Cai, Fürer, and Immerman show the following. When choosing each pair of edges between gadgets of adjacent vertices to be arbitrarily either parallel or twisted, we can only obtain one of two graphs up to isomorphism. More precisely, the isomorphism type of the resulting graph only depends on the parity of the number of twists in it.

In [30], first coloured graphs are constructed. By a standard transformation, which reduces isomorphism of coloured graphs to isomorphism of uncoloured graphs, one can remove the colours in the graph. There are suitable transformations that do not add more than $O(k)$ extra vertices. Thus, we can assume the graphs G and G' in Fact 6.34 to be uncoloured, while still preserving the property that for a fixed input graph, all possible resulting graphs with the same parity of the number of twists are isomorphic. We will continue to work with coloured graphs, knowing that the same reduction techniques can be applied.

Every gadget of the construction is, when considered by itself, identified by C^3. This can easily be verified by using the fact that the pairs of outer vertices have distinct colours and every two of them form a simple 8-cycle together with the middle vertices. It is straightforward to conclude that a graph that is indistinguishable from G or G' may only differ from them by the number of twists it has. Depending on that parity, the graph is either isomorphic to G or to G'.

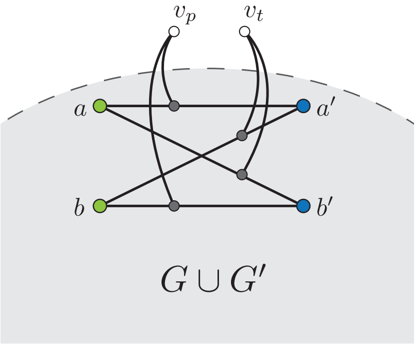


Figure 6.7: Our construction to show that higher-dimensional analogues of Corollaries 6.6 and 6.20 do not hold.

Theorem 6.35. *For every $k > 2$, there is a graph H of size $O(k)$ identified by C^3 for which the C^k-partition is strictly coarser than the orbit partition. Moreover, not all vertex-coloured versions of H are identified by C^k.*

Proof: Let G and G' be the graphs described in Fact 6.34. We form $G \cup G'$, which is the graph on $V(G) = V(G')$ with edge set $E(G) \cup E(G')$. In particular, this graph has the parallel edges $\{a, a'\}, \{b, b'\}$ and the twisted edges $\{a, b'\}, \{b, a'\}$. Subdivide these four edges and connect the two middle vertices of the subdivided parallel edges to a new vertex v_p and the two middle vertices of the subdivided twisted edges to a new vertex v_t (see Figure 6.7). Let H be the resulting graph.

To argue that H is identified by C^3, we will use the fact that in the construction described by Cai, Fürer, and Immerman, we can achieve that vertices in different gadgets have different colours. (Without this, it can still easily be proven that C^4 identifies the graph.) Since every gadget is identified by itself, the only critical edges are the ones connecting gadgets. They can either be twisted or parallel. However, since $G \cup G'$ has both versions present between the vertices a, a', b, b', for every inserted twist, there is an isomorphism to $G \cup G'$ that resolves the twist. Thus, the graph is identified because it is isomorphic to its twisted version. Consequently, the graph H is identified by C^3 since the newly added vertices v_p and v_t are the only vertices of degree 2 and therefore distinguished by C^3 from all other vertices.

It holds that v_p and v_t are indistinguishable by C^{k-4}. (This can easily be seen by considering pebble games. Spoiler could win the game on G and G' by simulating the game on H, placing 4 additional pebbles on a, b, a', and b', which would imply that G and G' can be distinguished by C^k.)

We claim that v_p and v_t are not in the same vertex orbit. Assuming otherwise, there is an automorphism that maps the neighbours of v_p to the neighbours of v_t and vice versa. This implies that there is an automorphism that maps the parallel edges to the twisted edges and vice versa. However, this means that G and G' are isomorphic, yielding a contradiction.

Note that, since v_p and v_t are indistinguishable by C^{k-4}, the graph obtained by colouring v_p with a colour different from the colour of every other vertex is not identified by C^{k-4}. □

Even if a graph is identified and the orbits of its automorphism group are correctly determined by C^k, it may still be the case that this does not hold for all coloured versions of the graph.

Theorem 6.36. *For every $k > 2$, there is a graph H of size $O(k)$ which is identified by C^3 such that the C^k-partition classes are the vertex orbits of H but there are vertex-coloured versions of H that are not identified by C^k and for which the C^k-partition classes are not the vertex orbits of H.*

Proof: We slightly modify the construction outlined in the previous proof (see Figure 6.8). We first subdivide the parallel edges $\{a, a'\}$ and $\{b, b'\}$ to obtain undirected paths $(a = a_1, a_2, a_3 = a')$ and $(b = b_1, b_2, b_3 = b')$, respectively, and also insert all twisted edges $\{a_i, b_{i+1}\}$ and $\{b_i, a_{i+1}\}$. As before, we subdivide all new edges and add, for $i \in \{1, 2\}$, vertices $v_{i,p}$ adjacent to the midpoint of $\{a_i, a_{i+1}\}$ and to the midpoint of $\{b_i, b_{i+1}\}$. We also add vertices $v_{i,t}$ adjacent to the midpoint of $\{a_i, b_{i+1}\}$ and to the midpoint of $\{b_i, a_{i+1}\}$. We obtain a graph H that is, by the same reasoning as above, identified by C^3. Furthermore, the logic C^3 determines the vertex orbits on H since, for $i \in \{1, 2\}$, the vertex $v_{i,t}$ can be mapped to $v_{i,p}$ by an automorphism. Colouring $v_{1,p}$ yields a graph whose orbits are not the C^k-classes, since $v_{2,p}$ and $v_{2,t}$ are not in the same orbit, but not distinguished by C^k, i.e. they are in the same C^k-partition class. □

By repeating the subdivision step, the construction described in the proof can easily be generalised to show that there exist vertex-coloured graphs of order $O(k)$ in which orbits are correctly determined by C^3 even if k vertices are individualised, but in which there are $k+1$ vertices such that individualisation of all of them produces a graph whose C^k-partition classes are not the vertex orbits of the automorphism group.

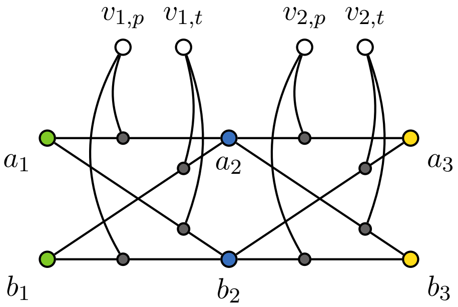


Figure 6.8: The construction used to show that if a graph and its vertex orbits are identified, this does not have to be the case for vertex-coloured versions of the graph.

6.5 Discussion

The characterisations given in Theorems 6.18 and 6.29 enable a full understanding of the expressive power of the 1-dimensional Weisfeiler-Leman algorithm. Moreover, as stated in Corollaries 6.19 and 6.31, they yield efficient criteria to decide whether a finite relational structure (or, more precisely, its associated edge-coloured partially oriented graph) is identified by Colour Refinement. Independently of our work, Arvind, Köbler, Rattan, and Verbitsky have investigated the structure of undirected graphs identified by C^2 and obtained results similar to the ones we provide in Section 6.2 (see [4, 6]).

In the previous section, we explained why we cannot expect to obtain nice complete characterisations of the graphs that higher-dimensional versions of the Weisfeiler-Leman algorithm are able to identify. Theorem 6.35 shows that basically none of the nice properties we could state for the graphs that are identified by the 1-dimensional Weisfeiler-Leman algorithm translates to higher dimensions.

Already for $k = 2$, the situation is quite intricate. It is known that all graphs whose colour class sizes are bounded by 3 are identified by the 2-dimensional Weisfeiler-Leman algorithm [90]. Still, by Theorem 3.13, for every $k \in \mathbb{N}$, there are pairs of graphs with colour class sizes bounded by 4 which are not identified by the k-dimensional Weisfeiler-Leman algorithm. By a careful combinatorial analysis of the coherent configurations associated with the input graphs, Fuhlbrück, Köbler, and Verbitsky found an efficient procedure to determine whether a graph with colour class sizes bounded by 4 is identified by the 2-dimensional Weisfeiler-Leman algorithm [46]. Still, we are far from a polynomial-time algorithm that decides whether arbitrary input graphs are identified by the 2-dimensional Weisfeiler-Leman algorithm.

7 The Power of the WL Algorithm to Decompose Graphs

We have investigated the expressive power of the 1-dimensional Weisfeiler-Leman algorithm and presented a complete characterisation of the graphs which it identifies. We also gave an intuition about major obstacles for a complete characterisation of the graphs which higher dimensions of the algorithm identify. Still, instead of trying to find a description of the entire identified graphs, towards a better understanding of the power of the k-dimensional Weisfeiler-Leman algorithm for $k \geq 2$, we can look for parts of a graph which the algorithm distinguishes from the remainder, i.e. which it "identifies" within the graph.

Once we know that the algorithm is able to detect certain subgraphs within a graph and distinguish them from others, we can consider the graph as decomposed with respect to those parts. Such graph decompositions are very powerful tools to analyse the structure of a graph that might be too complex when considered as a whole (see, for example, [63]).

Many of the combinatorial graph isomorphism algorithms use recursive graph decompositions, among those are Hopcroft's and Tarjan's isomorphism test for planar graphs [81, 82, 83] and Grohe's and Mariño's isomorphism test for graphs parameterised by their treewidth [67]. To be more precise, the latter publication proves that the $(k+2)$-dimensional Weisfeiler-Leman algorithm identifies every graph of treewidth at most k. Since it is able to make use of the bound on the treewidth, the Weisfeiler-Leman algorithm must implicitly compute at least some aspects of a corresponding tree decomposition.

Still, we are lacking a precise understanding of what those aspects are, what other graph decompositions the Weisfeiler-Leman algorithm is capable of computing, and what this actually means for the expressive power of the algorithm. In this chapter, we analyse the situation in detail. We present a proof that already the 2-dimensional Weisfeiler-Leman algorithm detects 2-separators in the input graph. We deduce that it implicitly computes the decomposition of a graph into its 3-connected components and formalise this statement. In the next chapter, we connect the techniques from this chapter to the expressive power of the algorithm by applying them to planar graphs.

This chapter combines contents that were published in [99], [100], and [101].

7.1 Detecting Separators with the WL Algorithm

In this section, we present some insights about the expressive power of the 2-dimensional Weisfeiler-Leman algorithm, which we will use for a reduction to 3-connected graphs in the following sections.

Observation 7.1. *For $k \in \mathbb{N}_{\geq 2}$, the k-dimensional Weisfeiler-Leman algorithm distinguishes every pair of graphs G and H which have different numbers of connected components from each other.*

Proof: It suffices to show the observation for $k = 2$ and for $|G| = |H|$. The statement then readily follows from the fact that for every vertex $v \in V(G)$, the colour $\chi_{G,2}(v, v)$ encodes the number of vertices contained in the same connected component as v. □

Note that the observation fails for $k = 1$. The proof of the observation exploits the fact that the 2-dimensional Weisfeiler-Leman algorithm distinguishes pairs of vertices between which there is a path in the given graph from pairs of vertices which are not connected. With a similar but more refined argument, we can show that the 2-dimensional Weisfeiler-Leman algorithm detects cut vertices in graphs. For this, we first prove that it distinguishes pairs of vertices from different 2-connected components from pairs of vertices that share a 2-connected component.

Theorem 7.2. *Assume $k \in \mathbb{N}_{\geq 2}$ and let G and H be two graphs. Let u and v be vertices from the same 2-connected component of G and let u' and v' be vertices that are not contained in a common 2-connected component of H. Then $\chi_{G,k}(u, v) \neq \chi_{H,k}(u', v')$.*

Proof: To improve readability, in this proof, we omit the subscript k, i.e. we write χ_G and χ_H instead of $\chi_{G,k}$ and $\chi_{H,k}$, respectively.

For an integer i and vertices x, y, denote by $W_i(x, y)$ the number of walks of length exactly i from x to y. (It will be clear from the context in which graph we count the number of walks.) By induction on i, it is easy to see that for $k \geq 2$, the inequality $W_i(x, y) \neq W_i(x', y')$ implies that $\chi_G(x, y) \neq \chi_H(x', y')$ (see also [161, page 18]). Thus, it suffices to show that for some i, we have $W_i(u, v) \neq W_i(u', v')$. Since u' and v' are not contained in the same 2-connected component, there is a cut vertex w' such that every walk from u' to v' passes through w'. Suppose that there does not exist a vertex w such that for all i, the following hold:

1. $W_i(u, w) = W_i(u', w')$
2. $W_i(w, w) = W_i(w', w')$
3. $W_i(w, v) = W_i(w', v')$.

Then for every w, we have $\chi_G(u,w) \neq \chi_H(u',w')$ or $\chi_G(w,w) \neq \chi_H(w',w')$ or $\chi_G(w,v) \neq \chi_H(w',v')$. If $\chi_G(w) \neq \chi_H(w')$, then $\chi_G(u,w) \neq \chi_H(u',w')$ and thus, for every vertex w it holds that $\chi_G(u,w) \neq \chi_H(u',w')$ or $\chi_G(w,v) \neq \chi_H(w',v')$. In other words, there is no vertex w such that $\big(\chi_G(w,v), \chi_G(u,w)\big) = \big(\chi_H(w',v'), \chi_H(u',w')\big)$. By the definition of the Weisfeiler-Leman algorithm, this implies that $\chi_G(u,v) \neq \chi_H(u',v')$.

Now suppose that there is a vertex w such that for all i, Conditions 1, 2, and 3 hold. Then for every i, the number of walks of length i from u to v which pass w equals the number of walks from u' to v' which pass w'. However, there must be a walk from u to v which avoids w. Let d be its length. We have $W_d(u,v) > W_d(u',v')$ and thus $\chi_G(u,v) \neq \chi_H(u',v')$. □

Next we argue that for $k \geq 2$, the k-dimensional Weisfeiler-Leman algorithm distinguishes cut vertices from other vertices.

Corollary 7.3. *Let $k \in \mathbb{N}_{\geq 2}$ and assume G and H are connected graphs. Let $w \in V(G)$ and $w' \in V(H)$ be vertices such that $G - \{w\}$ is connected and $H - \{w'\}$ is disconnected. Then $\chi_{G,k}(w) \neq \chi_{H,k}(w')$.*

Proof: Let u' and v' be two neighbours of w' not sharing a common 2-connected component in H. Note that such vertices do not exist for w in G.

It suffices to show that for all $u \in V(G)$, it holds that $\chi_{G,k}(u,w) \neq \chi_{H,k}(u',w')$. By Theorem 7.2, the colour $\chi_{H,k}(u',w')$ encodes the existence of v', which is a neighbour of w' and not contained in the same 2-connected component as u'. However, such a vertex does not exist in G for any choice of u and the fixed vertex w. □

As the example in Figure 7.1 shows, dimension $k = 1$ does not suffice to distinguish vertices contained in 1- or 2-separators from others. Thus, the bound $k \geq 2$ in the corollary is tight.

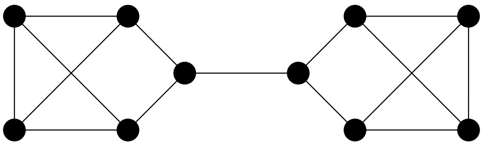


Figure 7.1: A 3-regular graph. The two central vertices are cut vertices, while all other vertices are not.

From the corollary, using the characterisation of the k-dimensional Weisfeiler-Leman algorithm via pebble games (see Theorem 3.19), we can directly deduce that the $(k+1)$-dimensional algorithm distinguishes k-separators from other k-tuples of vertices.

Recall that for ℓ vertices $u_1, \dots, u_\ell$ with $\ell < k$, we have set

$$\chi_{G,k}(u_1, \dots, u_\ell) \coloneqq \chi_{G,k}(u_1, \dots, u_\ell, \underbrace{u_\ell, \dots, u_\ell}_{k-\ell \text{ times}}).$$

Corollary 7.4. *Suppose $k \in \mathbb{N}$ and that G and H are connected graphs. Let $\{w_1, \dots, w_k\}$ be a k-separator in G. Let $\{v_1, \dots, v_k\} \subseteq V(H)$ and suppose that $\chi_{G,k+1}(w_1, \dots, w_k) = \chi_{H,k+1}(v_1, \dots, v_k)$. Then $\{v_1, \dots, v_k\}$ forms a k-separator in H.*

Proof: If $k = 1$, the statement follows by contraposition from Corollary 7.3. Now suppose $k \geq 2$. Let G and H be two graphs and suppose $\{w_1, \dots, w_k\} \subseteq V(G)$ is a k-separator in G. Also let $\{v_1, \dots, v_k\} \subseteq V(H)$ and furthermore, suppose that $\chi_{G,k+1}(w_1, \dots, w_k) = \chi_{H,k+1}(v_1, \dots, v_k)$. The claim follows from the observation that we can assume $G - \{w_1, \dots, w_{k-1}\}$ and $H - \{v_1, \dots, v_{k-1}\}$ to be connected, that $\{w_k\}$ is a cut vertex in the graph $G - \{w_1, \dots, w_{k-1}\}$, and that it suffices to show the analogous statement for H. □

In the remainder of this section, we show that indeed, already the 2-dimensional Weisfeiler-Leman algorithm distinguishes 2-separators from other pairs of vertices and the k-dimensional Weisfeiler-Leman algorithm distinguishes k-separators from other vertex k-tuples for $k \geq 2$. This is far from being obvious and in fact, it does not hold for $k = 1$ (cf. Figure 7.1). Also, for example, a statement as in Theorem 7.2 saying that a vertex pair (u, v) obtains a distinct colour whenever u and v are separated by a certain k-separator does not follow straight away as a generalisation of the theorem. Indeed, Spoiler would have to place $k - 1$ pebbles on $k - 1$ vertices of the k-separator and then use 3 more pebbles to count the walks from u to v just as in the theorem. However, the k-dimensional Weisfeiler-Leman algorithm corresponds to the pebble game with just $k + 1$ pebble pairs.

We essentially perform a structural analysis. We reduce the general case to the case of at most two vertex colours with respect to χ_2 in Section 7.1.1. Then, in Section 7.1.2, we examine the graphs in which all vertices obtain the same colour with respect to χ_2. In Section 7.1.3, we then consider the graphs in which there are exactly two vertex colours with respect to χ_2.

Exploiting the correspondence from Theorem 3.19, our results imply that for every $n \in \mathbb{N}$, there is a formula $\varphi_n(x_1, x_2) \in \mathsf{C}^3$ such that for every n-vertex graph G, it holds that $G \models \varphi_n(v, w)$ if and only if $\{v, w\}$ is a 2-separator in G. The formulas which our proof yields are essentially a disjunction over all n-vertex graphs. While this makes the proof rather involved, it also stresses the power of the 2-dimensional Weisfeiler-Leman algorithm and the expressive power of the logic C^3.

7.1.1 Reduction to Two Colours

In this subsection, we show that with a suitable structural statement about graphs in which χ_2 only produces at most two vertex colours, for $k \geq 2$, already the k-dimensional Weisfeiler-Leman algorithm detects k-separators, i.e. we strengthen the statement from Corollary 7.4.

Recall again that, as Figure 7.1 shows, the condition $k \geq 2$ is necessary here, since the 1-dimensional Weisfeiler-Leman algorithm does not distinguish cut vertices from vertices that do not separate the given graph.

Let S be a set of colours. We say a path $u_0, \dots, u_\ell$ *avoids* S if $\chi_G(u_i, u_i) \notin S$ for every $i \in [\ell - 1]$. Note that we impose no restriction on the colours of the endpoints of the path.

Lemma 7.5. *Let G be a graph and let $X \coloneqq \{\chi_{G,2}(v,v) \mid v \in V(G)\}$. Furthermore, let $S \subseteq X$ and define $G[[S]] = (V, E)$, where $V = \{v \in V(G) \mid \chi_{G,2}(v,v) \in S\}$ and*

$$E = \Big\{\{u,v\} \,\Big|\, \text{there is a path from } u \text{ to } v \text{ in } G \text{ that avoids } S\Big\}.$$

Then $\chi_{G[[S]],2}(u,u) = \chi_{G[[S]],2}(v,v)$ for all $u, v \in V$ with $\chi_{G,2}(u,u) = \chi_{G,2}(v,v)$.

Proof: It is easy to see that, given two vertices $u, v \in V(G)$, the 2-dimensional Weisfeiler-Leman algorithm detects whether there is a path from u to v that avoids S. Thus, there is a set of colours T such that $\{u,v\} \in E$ if and only if $\chi_{G,2}(u,v) \in T$. Hence, any refinement performed by the 2-dimensional Weisfeiler-Leman algorithm in $G[[S]]$ can also be done in G. □

For our reduction of the general case to two vertex colours, we presuppose the following structural insight about graphs with at most two χ_2-vertex colours. We prove the theorem in Section 7.1.3.

Theorem 7.6. *Let G be a 2-connected graph with the following properties.*

1. *G has a 2-separator $\{w_1, w_2\}$, and*
2. *for every $v \in V(G)$, there is an $i \in \{1,2\}$ such that $\chi_{G,2}(v,v) = \chi_{G,2}(w_i,w_i)$.*

Then G is a cycle.

Before we prove the theorem, we apply it to perform our reduction, i.e. to deduce that the 2-dimensional Weisfeiler-Leman algorithm detects 2-separators in graphs.

Theorem 7.7. *Let G and H be 2-connected graphs and let $w_1, w_2 \in V(G)$ such that $\{w_1, w_2\}$ forms a 2-separator in G. Also let $v_1, v_2 \in V(H)$ and suppose $\chi_{G,2}(w_1,w_2) = \chi_{H,2}(v_1,v_2)$. Then $\{v_1, v_2\}$ forms a 2-separator in H.*

Proof: Let $S := \{\chi_{G,2}(w_1, w_1), \chi_{G,2}(w_2, w_2)\}$ and let $G' := G[[S]]$. Clearly, the graph G' is connected.

First suppose that $|V(G')| = 2$. Let $A := \{v \in V(H) \mid \chi_{H,2}(v, v) \in S\}$. Then $|A| = 2$ and thus $A = \{v_1, v_2\}$. Moreover, $H - A$ is disconnected, since the 2-dimensional Weisfeiler-Leman algorithm detects that $G - \{w_1, w_2\}$ is disconnected. Hence, $\{v_1, v_2\}$ forms a 2-separator in H.

Now assume $|V(G')| \geq 3$. We argue that G' is 2-connected. Suppose, towards a contradiction, that there is a separating vertex w in G'. Let C and C' be the vertex sets of two connected components of $G' - \{w\}$. Let $v \in C$ and $v' \in C'$. We show that w separates v from v' in G. Towards a contradiction, suppose there is a path P from v to v' in G that does not pass through w. Then there is a corresponding path P' in G', which simply skips all inner vertices of P not contained in S. In particular, P' connects v and v', but avoids w. This contradicts w being a cut vertex in G'. Hence, G' is 2-connected.

Suppose there is a vertex set C of a connected component of $G - \{w_1, w_2\}$ such that $V(G') \subseteq C \cup \{w_1, w_2\}$. Let C' be the vertex set of a second connected component of $G - \{w_1, w_2\}$ and let $v \in C'$. Then w_1 and w_2 are the only vertices with colour in S that can be reached from v via a path that avoids S. Hence, using the expressive power of the 2-dimensional Weisfeiler-Leman algorithm, it is not hard to see that there must also be a vertex $u \in V(H)$ such that v_1 and v_2 are the only vertices with colour in S that can be reached from u via a path that avoids S. Since $|V(G')| \geq 3$, there is a $u' \in V(H)$ such that $v_1 \neq u' \neq v_2$ and $\chi_H(u', u') \in S$, because in order not to be distinguished, unions of colour classes with colour in S must have the same cardinality in both graphs. But then $\{v_1, v_2\}$ separates u from u' in H and thus, $\{v_1, v_2\}$ forms a 2-separator in H.

In the other case, $\{w_1, w_2\}$ forms a 2-separator in G'. Hence, G' is a cycle by Lemma 7.5 and Theorem 7.6. Note that $|V(G')| \geq 4$ and $\{w_1, w_2\} \notin E(G')$. It follows that $H[[S]]$ is also a cycle, since otherwise, the 2-dimensional Weisfeiler-Leman algorithm would distinguish the graphs. Also, $|V(H[[S]])| \geq 4$ and $\{v_1, v_2\} \notin E(H[[S]])$. So $\{v_1, v_2\}$ forms a 2-separator in $H[[S]]$ and thus, it also forms a 2-separator in H. □

The theorem directly translates to higher dimensions.

Corollary 7.8. *Suppose $k \in \mathbb{N}_{\geq 2}$. Let G and H be connected graphs. Assume that $\{w_1, \ldots, w_k\} \subseteq V(G)$ is a k-separator in G. Let $\{v_1, \ldots, v_k\} \subseteq V(H)$ and suppose $\chi_{G,k}(w_1, \ldots, w_k) = \chi_{H,k}(v_1, \ldots, v_k)$. Then $\{v_1, \ldots, v_k\}$ forms a k-separator in H.*

Proof: First suppose $k = 2$. If G and H are 2-connected, the statement is exactly Theorem 7.7. If either G or H is not 2-connected, then that graph contains a cut vertex, while the other graph does not. By Corollary 7.3, the presence of the cut vertex is encoded in every vertex colour and thus, the multisets $\{\{\chi_{G,2}(u, v) \mid u, v \in V(G)\}\}$ and $\{\{\chi_{H,2}(u, v) \mid u, v \in V(H)\}\}$ are disjoint. Therefore, the statement trivially holds.

Suppose both G and H are not 2-connected. The statement is obviously true if w_1 or w_2 is a cut vertex in G. If this is not the case, then w_1 and w_2 must lie in a common 2-connected component of G, otherwise they form no 2-separator. By Theorem 7.2, the same must hold for v_1 and v_2 in H. Furthermore, again by Theorem 7.2, the 2-dimensional Weisfeiler-Leman algorithm distinguishes arcs from w_1, w_2 and from v_1, v_2 to vertices in the same 2-connected component from arcs to other vertices. Thus, the algorithm distinguishes the 2-connected component C_G containing w_1 and w_2 and the 2-connected component C_H containing v_1 and v_2, respectively, from the remainder of G and H, respectively. Hence, the computed colours in G and H induce a refined partition of the one induced by the colours computed in C_G and C_H, respectively. Therefore, (v_1, v_2) and (w_1, w_2) have equal colours in C_G and C_H and thus, applying Theorem 7.7, we can deduce that $\{v_1, v_2\}$ forms a 2-separator in C_H and therefore also in H.

Finally, consider the case $k > 2$. Let G and H be graphs and suppose that $\{w_1, \dots, w_{k+1}\} \subseteq V(G)$ is a $(k+1)$-separator in G. Also let $\{v_1, \dots, v_{k+1}\} \subseteq V(H)$ and furthermore, suppose that $\chi_{G,k+1}(w_1, \dots, w_{k+1}) = \chi_{H,k+1}(v_1, \dots, v_{k+1})$. The claim follows from the observation that we can assume $G - \{w_1, \dots, w_{k-2}\}$ and $H - \{v_1, \dots, v_{k-2}\}$ to be connected, that $\{w_{k-1}, w_k\}$ forms a 2-separator in the graph $G - \{w_1, \dots, w_{k-2}\}$, and that it suffices to show the analogous statement for H. □

It remains to prove Theorem 7.6. The proof will be carried out in the following two subsections.

7.1.2 One Colour

Our goal is to prove that the 2-dimensional Weisfeiler-Leman algorithm distinguishes vertex pairs that are separators in a graph from other pairs of vertices. By the reduction from the previous subsection, we can restrict ourselves to graphs with at most two χ_2-vertex colours. In this subsection, we start with an analysis of the graphs G in which all vertices obtain the same $\chi_{G,2}$-colour.

A main tool for the analysis are distance patterns of vertices. For a graph G and a vertex $v \in V(G)$, let $D(v) := \{\{\mathrm{dist}(v, w) \mid w \in V(G)\}\}$. Note that for vertices $u, v \in V(G)$, it holds that $\chi_{G,2}(u, u) \neq \chi_{G,2}(v, v)$ whenever $D(u) \neq D(v)$, since the 2-dimensional Weisfeiler-Leman algorithm detects distances between vertex pairs.

Lemma 7.9. *Let G be a graph and $uv \in E(G)$. Suppose that $D(u) = D(v)$. Then*

$$\begin{aligned}&\{\{\mathrm{dist}(u, w) \mid w \in V(G)\colon\ \mathrm{dist}(u, w) < \mathrm{dist}(v, w)\}\}\\ =\ &\{\{\mathrm{dist}(v, w) \mid w \in V(G)\colon\ \mathrm{dist}(v, w) < \mathrm{dist}(u, w)\}\}.\end{aligned}$$

Proof: We have $|\mathrm{dist}(v, w) - \mathrm{dist}(u, w)| \leq 1$ for all $w \in V(G)$ since $\{u, v\} \in E(G)$. Suppose the statement is false and let $d \in \mathbb{N}$ be the maximal number such that d

has distinct multiplicities in the two multisets. Let m_1 be the multiplicity of d in the first multiset and m_2 be the multiplicity of d in the second. Without loss of generality, assume that $m_1 > m_2$. Then

$$m_1 = |\{w \in V(G) \mid \text{dist}(u,w) = d \wedge \text{dist}(v,w) = d+1\}|$$

and

$$m_2 = |\{w \in V(G) \mid \text{dist}(v,w) = d \wedge \text{dist}(u,w) = d+1\}|.$$

But

$$\begin{aligned} &|\{w \in V(G) \mid \text{dist}(u,w) = d+1 \wedge \text{dist}(v,w) = d+1\}| \\ = &|\{w \in V(G) \mid \text{dist}(v,w) = d+1 \wedge \text{dist}(u,w) = d+1\}| \end{aligned}$$

and

$$\begin{aligned} &|\{w \in V(G) \mid \text{dist}(u,w) = d+2 \wedge \text{dist}(v,w) = d+1\}| \\ = &|\{w \in V(G) \mid \text{dist}(v,w) = d+2 \wedge \text{dist}(u,w) = d+1\}|, \end{aligned}$$

where the first equality is trivial and the second equality follows from the maximality of d. However, then $D(u)$ and $D(v)$ contain the number $d+1$ in distinct multiplicities, a contradiction. □

Throughout the remainder of this subsection, if not explicitly stated otherwise, we make the following assumption.

Assumption 7.10. *G is a connected graph on n vertices with the following properties:*

1. *G has a 2-separator $\{w_1, w_2\}$, and*
2. *$\chi_{G,2}(u,u) = \chi_{G,2}(v,v)$ for all $u, v \in V(G)$.*

In the rest of this subsection, we analyse the structure of G and ultimately prove that G is a cycle. Note that Assumption 7.10 implies that G is regular, i.e. $\deg(u) = \deg(v)$ for all $u, v \in V(G)$. We collect some further structural observations.

Lemma 7.11. *G is 2-connected.*

This is a consequence of Condition 2 in Assumption 7.10, since the 2-dimensional Weisfeiler-Leman algorithm distinguishes cut vertices from other vertices (see Corollary 7.3) and it is easy to see that it is not possible that every vertex in G is a cut vertex. Note that the lemma implies that each of the separator vertices w_1 and w_2 has at least one neighbour in each of the connected components of $G - \{w_1, w_2\}$.

Lemma 7.12. *Let C be the vertex set of a connected component of $G - \{w_1, w_2\}$ such that $|C| < \frac{n}{2}$ and let $v \in C$. Then there is no vertex $u \in N(v)$ such that* $\text{dist}(u, w_1) < \text{dist}(v, w_1)$ *and* $\text{dist}(u, w_2) < \text{dist}(v, w_2)$.

Proof: Suppose towards a contradiction that such a vertex $u \in N(v)$ exists. For all $w \in V(G)$, we have $|\operatorname{dist}(v,w) - \operatorname{dist}(u,w)| \leq 1$, since $\{u,v\} \in E(G)$. Furthermore, we have $\sum_{w \in V(G)} \big(\operatorname{dist}(v,w) - \operatorname{dist}(u,w)\big) = 0$ because $D(v) = D(u)$ due to Condition 2 in Assumption 7.10. But $\operatorname{dist}(v,w) > \operatorname{dist}(u,w)$ for all $w \in V(G) \setminus C$, and $|V(G) \setminus C| > \frac{n}{2}$. This is a contradiction. □

Lemma 7.13. *Let* $d := \operatorname{dist}(w_1, w_2)$ *and let* C *be the vertex set of a connected component of* $G - \{w_1, w_2\}$ *such that* $|C| \leq \frac{n-2}{2}$. *Then for all* $v \in C \cup \{w_1, w_2\}$ *and all* $i \in \{1,2\}$, *it holds that* $\operatorname{dist}(v, w_i) \leq d$.

Proof: By symmetry, it suffices to prove $\operatorname{dist}(v, w_2) \leq d$. The statement is proved by induction on $\ell := \operatorname{dist}(v, w_1)$. For $\ell = 0$, it holds that $v = w_1$ and $\operatorname{dist}(w_1, w_2) = d$. So suppose the statement holds for all $u \in C \cup \{w_1, w_2\}$ with $\operatorname{dist}(u, w_1) \leq \ell$. Obviously, the statement is true if $v = w_1$ or $v = w_2$. So pick $v \in C$ with $\operatorname{dist}(v, w_1) = \ell + 1$. Let $u \in N(v)$ such that $\operatorname{dist}(u, w_1) \leq \ell$. Then $\operatorname{dist}(v, w_2) \leq \operatorname{dist}(u, w_2) \leq d$ by Lemma 7.12 and the induction hypothesis. □

Lemma 7.14. $\{w_1, w_2\} \notin E(G)$.

Proof: Suppose towards a contradiction that $\{w_1, w_2\} \in E(G)$. Let C be the vertex set of a connected component of $G - \{w_1, w_2\}$ such that $|C| \leq \frac{n-2}{2}$. By Lemma 7.13, we conclude that $C \subseteq N(w_1) \cap N(w_2)$. Let $v \in C$. Since G is 2-connected, the vertex w_1 must have at least one neighbour in $V(G) \setminus C$, in addition to being adjacent to C and to w_2. Thus, $\deg(w_1) \geq |C| + 2 > |C| - 1 + |\{w_1, w_2\}| \geq \deg(v)$, which contradicts G being a regular graph. □

Lemma 7.15. *Suppose that* $N(w_1) \cap N(w_2) \neq \emptyset$. *Then* G *is a cycle.*

Proof: By Lemma 7.14, it holds that $\{w_1, w_2\} \notin E(G)$. Furthermore, by the assumption of the lemma, we have $\operatorname{dist}(w_1, w_2) = 2$. Let C be the vertex set of a connected component of $G - \{w_1, w_2\}$ such that $|C| \leq \frac{n-2}{2}$. Also let $C' := V(G) \setminus (C \cup \{w_1, w_2\})$. For $i, j \geq 1$ let

$$C_{i,j} := \{v \in C \mid \operatorname{dist}(v, w_1) = i \text{ and } \operatorname{dist}(v, w_2) = j\}.$$

By Lemma 7.13, we conclude $C_{i,j} = \emptyset$ unless $(i,j) \in \{(1,1),(1,2),(2,1),(2,2)\}$.

Suppose there exists $v \in C_{1,2}$. We have $D(v) = D(w_1)$ and, by Lemma 7.11, also $N(w_1) \cap C' \neq \emptyset$. Thus, there is a vertex $u' \neq w_1$ such that $\operatorname{dist}(w_1, u') < \operatorname{dist}(v, u')$ and therefore, by Lemma 7.9, there is also a vertex $u \neq v$ such that $\operatorname{dist}(v, u) < \operatorname{dist}(w_1, u)$. For every such vertex u, it holds that $\operatorname{dist}(w_1, u) \leq 2$ and thus, $u \in N(v)$. Therefore, by Lemma 7.9, for every vertex $v' \neq w_1$ with $\operatorname{dist}(w_1, v') < \operatorname{dist}(v, v')$, it holds that $v' \in N(w_1)$. This implies that there is no $v' \in C'$ such that $\operatorname{dist}(w_1, v') = 2$ since such a vertex would satisfy $3 = \operatorname{dist}(v, v') > \operatorname{dist}(w_1, v')$. Because w_2 is not a cut vertex (cf. Lemma 7.11), from every $v' \in C'$, there is a path to w_1 that does not contain w_2. However, this is only possible if there is no vertex $v' \in C'$ such that $\operatorname{dist}(w_1, v') > 1$, in other words: $C' \subseteq N(w_1)$. Since G is regular and $|N(w_1) \setminus C'| \geq 1$, it follows that $\deg(v) \geq |C'| + 1$. But

$N(v) \subseteq (C \cup \{w_1\}) \setminus \{v\}$, which implies $\deg(v) \leq |C|$. The combination of both inequalities yields $|C'|+1 \leq |C|$, which implies $n = 2+|C|+|C'| \leq 1+2\cdot|C| \leq n-1$, a contradiction. So $C_{1,2} = \emptyset$ and, by symmetry, it also holds that $C_{2,1} = \emptyset$. But then $C_{2,2} = \emptyset$ by Lemma 7.12.

So $C = C_{1,1}$, which means that $C \subseteq N(w_1) \cap N(w_2)$. In particular, $\deg(w_1) \geq |C|+1$ since $|N(w_1) \cap C'| \geq 1$. Since G is regular, this implies that $\deg(v) \geq |C|+1$ for every $v \in C$, which is only possible if $N[v] = C \cup \{w_1, w_2\}$ (where $N[v] := N(v) \cup \{v\}$). Because $C \neq \emptyset$, this means that there is a vertex $v \in V(G)$ such that $G[N[v]]$ contains only one non-edge. Now by Condition 2 in Assumption 7.10, this also has to hold for w_1, and hence, since no vertex in $C \cap N(w_1)$ is adjacent to any vertex in $C' \cap N(w_1)$, it must hold that $\deg(w_1) = 2$. Therefore, by regularity, all vertices in G have degree 2 and thus, being connected, G is a cycle. □

With the collected tools at hand, we can deduce the structure of G.

Lemma 7.16. *G is a cycle.*

Proof: It suffices to prove the lemma for the case that G is a graph with a maximum edge set that satisfies Assumption 7.10. Indeed, if G is a cycle, then G has n edges, and the lemma trivially holds for every graph with less edges, since every connected regular graph has at least n edges.

Let $d := \mathrm{dist}(w_1, w_2)$. By Lemmas 7.14 and 7.15, we can assume that $d \geq 3$. Let C be the vertex set of a connected component of $G - \{w_1, w_2\}$ of size $|C| \leq \frac{n-2}{2}$. Also let $C' := V(G) \setminus (C \cup \{w_1, w_2\})$. For $i, j \in \mathbb{N}_0$, let

$$C_{i,j} := \{v \in C \mid \mathrm{dist}(v, w_1) = i \text{ and } \mathrm{dist}(v, w_2) = j\}.$$

By Lemma 7.13, we conclude that $C_{i,j} = \emptyset$ unless $i, j \leq d$. Furthermore, by the definition of d, we have that $C_{i,j} = \emptyset$ unless $i + j \geq d$. This situation is also visualised in Figure 7.2.

Claim 1. $C_{d,d} = C_{d-1,d} = C_{d,d-1} = \emptyset$.

Proof of the claim: We argue that $C_{d,d} = C_{d-1,d} = \emptyset$. By symmetry, we also obtain that $C_{d,d-1} = \emptyset$.

Suppose towards a contradiction that $C_{d,d} \cup C_{d-1,d} \neq \emptyset$ and pick $v \in C_{\ell,d}$ for some $\ell \in \{d, d-1\}$. Let $v_0, \dots, v_\ell$ be a shortest path from $w_1 = v_0$ to $v = v_\ell$. By an easy inductive argument and Lemmas 7.12 and 7.13, it follows that $v_i \in C_{i,d}$ for every $i \in [\ell]$. In particular, $C_{1,d} \neq \emptyset$.

Now let $v' \in C_{1,d}$ and consider an arbitrary vertex $u \neq v'$ such that $\mathrm{dist}(v', u) < \mathrm{dist}(w_1, u)$ (a possible choice is $u = v_2$, since $d \geq 3$). Then $u \in C$ and $\mathrm{dist}(v', u) \leq d - 1$ by Lemma 7.13. Since $D(v') = D(w_1)$, we conclude that for every vertex $u \neq w_1$ with $\mathrm{dist}(v', u) > \mathrm{dist}(w_1, u)$, it holds that $\mathrm{dist}(w_1, u) \leq d - 1$ (cf. Lemma 7.9). In particular, $\mathrm{dist}(w_1, u) \leq d - 1$ for all $u \in C'$.

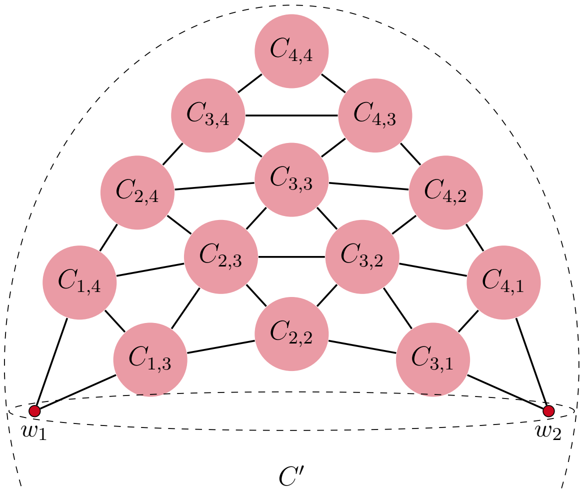


Figure 7.2: Visualisation of the sets $C_{i,j}$ for $d = 4$ in the proof of Lemma 7.16. Each arc between two sets indicates that there may be edges connecting vertices from the two sets.

Overall, this means that, on the one hand, $\text{dist}(w_1, u) \leq d$ for all $u \in V(G)$. On the other hand, there is a $u \in C'$ such that $\text{dist}(w_1, u) \geq 2$ because $d \geq 3$. But then $\text{dist}(v, u) \geq d + 1$ for $v \in C_{d,d} \cup C_{d-1,d}$. So $D(w_1) \neq D(v)$, which is a contradiction. ⌟

Claim 2. Let $u, v \in V(G)$ such that $\text{dist}(u, v) < d$. Then there is a unique shortest path from u to v.

Proof of the claim: Suppose the statement does not hold and let $\ell < d$ be the minimal distance for which the claim is violated, i.e. there are vertices $u, v \in V(G)$ such that there are two paths of length $\ell = \text{dist}(u, v)$ from u to v. Also let $E' := \big\{\{u', v'\} \,\big|\, \chi_{G,2}(u, v) = \chi_{G,2}(u', v')\big\}$ and consider the graph $G' := (V(G), E(G) \cup E')$, i.e. the graph obtained from G by inserting all (undirected) edges contained in the set E'. We argue that G' still satisfies Assumption 7.10, which contradicts the edge maximality of G.

First, the colouring $\chi_{G,2}$ is also a stable colouring for G', which implies that $\chi_{G,2}$ refines the colouring $\chi_{G',2}$. In particular, Condition 2 of Assumption 7.10 is satisfied for the graph G'.

Now let $\{u', v'\} \in E'$. Then $\text{dist}(u', v') = \ell$ and there are at least two different walks of length ℓ from u' to v' because the same statement holds for u and v. Due to the minimality of ℓ, the two walks are internally vertex-disjoint paths. If u' and v' lie in different connected components of $G - \{w_1, w_2\}$, then one of the two paths must pass through w_1 and one through w_2, forming a cycle of length $2\ell < 2d$. This implies

$\text{dist}(w_1, w_2) < d$, a contradiction. Thus, we conclude that there is a connected component with vertex set C of $G - \{w_1, w_2\}$ such that $u', v' \in C \cup \{w_1, w_2\}$. But this means Condition 1 of Assumption 7.10 is satisfied for the graph G'. ⌟

Claim 3. Let $u, v \in V(G)$ such that $\ell \coloneqq \text{dist}(u, v) < d$. Furthermore, suppose there is a walk $u = u_0, \dots, u_{\ell+1} = v$ of length $\ell + 1$ from u to v. Then there is an $i \in [\ell]$ such that $\{u_{i-1}, u_{i+1}\} \in E(G)$.

Proof of the claim: Suppose the statement does not hold and let $\ell < d$ be the minimal number for which the claim is violated. Let $u, v \in V(G)$ such that $\ell = \text{dist}(u, v)$ and there is a walk $u = u_0, \dots, u_{\ell+1} = v$ of length $\ell + 1$ from u to v such that for all $i \in [\ell]$, it holds that $\{u_{i-1}, u_{i+1}\} \notin E(G)$. Also, let $E' \coloneqq \big\{\{u', v'\} \mid \chi_{G,2}(u, v) = \chi_{G,2}(u', v')\big\}$ and consider the graph $G' = (V(G), E(G) \cup E')$. Similarly to the previous claim, we argue that G' still satisfies Assumption 7.10, which contradicts the edge maximality of G.

Indeed, by the same argument as in the previous claim, Condition 2 of Assumption 7.10 is satisfied for the graph G'.

Now let $\{u', v'\} \in E'$. Then $\text{dist}(u', v') = \ell$ and there is a walk of length $\ell + 1$ from u' to v'. Suppose that for every such walk $u' = u'_0, \dots, u'_{\ell+1} = v'$, there is an $i \in [\ell]$ such that $\{u'_{i-1}, u'_{i+1}\} \in E(G)$. On such a walk, the shortcut from u_{i-1} to u_{i+1} yields a walk of length ℓ from u' to v', i.e. a shortest one. However, by Claim 2, the shortest walk is unique and also, the colour of $\{u'_{i-1}, u'_{i+1}\}$ encodes the existence of the triangle formed by the vertices u'_{i-1}, u'_i, u'_{i+1}. However, this implies by the assumptions for u and v that the shortest path from u to v does not contain any edge which obtains the colour $\chi_{G,2}(u'_{i-1}, u'_{i+1})$ with respect to the 2-dimensional Weisfeiler-Leman algorithm. Therefore, $\{u', v'\} \notin E'$, a contradiction. Thus, there is a walk $u' = u'_0, \dots, u'_{\ell+1} = v'$ of length $\ell + 1$ from u' to v' such that for all $i \in [\ell]$, it holds that $\{u'_{i-1}, u'_{i+1}\} \in E(G)$.

Due the minimality of ℓ, the unique shortest path from u' to v' and $u'_0, \dots, u'_{\ell+1}$ are internally vertex-disjoint. Since $\text{dist}(w_1, w_2) = d$, this implies that there is a connected component of $G - \{w_1, w_2\}$ with vertex set C such that $u', v' \in C \cup \{w_1, w_2\}$ (using the same arguments as before). But this again means Condition 1 of Assumption 7.10 is also satisfied for the graph G'. ⌟

These claims drastically restrict the structure of the graph G and will allow us to prove that G is a cycle. Intuitively speaking, the claims imply that, when looking towards the connected component $G[C]$ from any of the w_i, the graph has a tree-like structure, i.e. the initial segments of paths up to length $d - 1$ starting in w_i form a tree rooted in w_i.

For $k' \in [d, 2d]$, let

$$\mathcal{C}_{k'} \coloneqq \bigcup_{i,j\,:\, i+j=k'} C_{i,j}.$$

Let $k \in [d, 2d]$ be the maximal number such that $\mathcal{C}_k \neq \emptyset$. Note that $k \leq 2d-2$ by Claim 1. First suppose that $k \geq d+1$.

Claim 4. Let $v \in C_{i,k-i}$ for $i \in [k-d+1, d-1]$. Then $|N(v) \cap (C_{i-1,k-i+1} \cup C_{i-1,k-i})| = 1$ and also $|N(v) \cap (C_{i,k-i-1} \cup C_{i+1,k-i-1})| = 1$.

Proof of the claim: This follows directly from Claim 2. ⌟

Claim 5. Suppose $C_{i,k-i} \neq \emptyset$ for some $k-d+1 \leq i \leq d-1$. Then G is a cycle.

Proof of the claim: Pick i with $i \in [k-d+1, d-1]$ such that $C_{i,k-i} \neq \emptyset$ and let $v \in C_{i,k-i}$. By Claim 4 and the maximality of k, we get that $|N(v) \setminus C_{i,k-i}| = 2$. If $|N(v)| = 2$, then we are done (recall that G is regular and connected). So suppose there is a $u \in N(v) \cap C_{i,k-i}$. Then, using Claim 3, it is not hard to see that $N(u) \setminus C_{i,k-i} = N(v) \setminus C_{i,k-i}$ (observe that $i < d$ and $k-i < d$). Now let A be the vertex set of the connected component containing v in the graph $G[C_{i,k-i}]$. Then $G[A]$ is a clique. Indeed, for every pair $u, u' \in A$, there are at least two paths of length 2 from u to u', since $N(u) \setminus C_{i,k-i} = N(u') \setminus C_{i,k-i}$. Thus, $\{u, u'\} \in E(G)$ by Claim 2. But now $G[N[v]]$ contains at most one non-edge, namely between the vertices in $N(v) \setminus C_{i,k-i}$. So the same has to be true for w_1, which implies that $\deg(w_1) = 2$. ⌟

Hence, we can assume that $C_{i,k-i} = \emptyset$ for all $i \in [k-d+1, d-1]$. Since $\mathcal{C}_k \neq \emptyset$, this means that $C_{d,k-d} \neq \emptyset$ or $C_{k-d,d} \neq \emptyset$. Without loss of generality, assume that $C_{k-d,d} \neq \emptyset$. Note that $k-d < d$.

Let $v \in C_{k-d,d}$. Let $w \in N(v)$ such that $\operatorname{dist}(w_1, w) = k-d-1$. Note that $w \in C_{k-d-1,d}$ (when $k = d+1$, then $w = w_1$). Also, observe that $|N(v) \cap C_{k-d-1,d}| = 1$ by Claim 2. Let A be the vertex set of the connected component in $G[C_{k-d,d}]$ containing v. Let $u, u' \in A$ such that $\{u, u'\} \in E(G)$. Then $N(u) \cap C_{k-d-1,d} = N(u') \cap C_{k-d-1,d}$ by Claim 3. So $N(u) \cap C_{k-d-1,d} = \{w\}$ for every $u \in A$. Also $G[A]$ forms a clique, since there is a unique shortest path between pairs of vertices at distance 2 by Claim 2. So $G[A \cup \{w\}]$ forms a clique. Now let $u \in N(v) \setminus (A \cup \{w\})$. Then $u \in C_{k-d,d-1}$ (recall that $C_{k-d+1,d-1} = \emptyset$). First, $\{u, w\} \in E(G)$ by Claim 3. Also, $A \subseteq N(u)$, since there is a unique shortest path between pairs of vertices at distance 2 by Claim 2. Finally, let $u' \in N(v) \setminus (A \cup \{w\})$ such that $u \neq u'$. Then $\{v, w\} \subseteq N(u) \cap N(u')$ and hence, $\{u, u'\} \in E(G)$, again using Claim 2. So overall, the closed neighbourhood $N[v]$ forms a clique and thus, the same must hold for $N[w_1]$, which is a contradiction, since w_1 belongs to a separator.

In the other case, $k = d$. Observe that $C_{i,d-i} \neq \emptyset$ for all $i \in [d-1]$. Just like in Chapter 4, for two sets $U, W \subseteq V(G)$, we let $G[U, W]$ be the subgraph of G with vertex set $U \cup W$ and edge set $E(G) \cap \big\{\{u, w\} \,\big|\, u \in U, w \in W\big\}$, i.e. the bipartite graph induced between U and W.

Claim 6. $|C_{i,d-i}| = |C_{j,d-j}|$ for all $i, j \in [d-1]$. Also, $G[C_{i,d-i}, C_{i+1,d-i-1}]$ is a perfect matching (i.e. every vertex in the graph has degree 1) for $i \in [d-1]$.

Proof of the claim: This follows directly from Claim 2. ⌟

Since G is regular and $|N(w_1)| \geq |C_{1,d-1}| + 1$, we further conclude that $G[C_{i,d-i}]$ is a complete graph for every $i \in [d-1]$. Now suppose that $|C_{i,d-i}| \geq 2$ for some (and therefore every) $i \in [d-1]$. Then G contains an induced subgraph isomorphic to C_4, the cycle of length 4, which contradicts Claim 2 (recall that $d \geq 3$). So $|C_{i,d-i}| = 1$ for all $i \in [d-1]$ and hence, G is a cycle. □

Reformulating the previous lemma, we obtain the following theorem.

Theorem 7.17. *Let G be a graph such that $\chi_{G,2}(u,u) = \chi_{G,2}(v,v)$ for all $u, v \in V(G)$. Then (exactly) one of the following holds:*

1. *G is not connected, or*
2. *G is 3-connected, or*
3. *G is a cycle of length $\ell \geq 3$.*

This concludes our analysis of the graphs in which all vertices have the same colour with respect to the 2-dimensional Weisfeiler-Leman algorithm. In the next subsection, we treat the graphs with two χ_2-vertex colours.

7.1.3 Two Colours

Recall that our overall goal is to prove that the 2-dimensional Weisfeiler-Leman algorithm assigns special colours to 2-separators in a graph assuming there are at most two χ_2-vertex colours. We will use Lemma 7.16 to prove this in case the tuples (u,u) and (v,v) of a 2-separator $\{u,v\}$ obtain the same χ_2-colour. To treat the much more difficult case that u and v obtain distinct colours, we intend to generalise the results of the previous subsection to two vertex colours. Maybe somewhat surprisingly, we obtain a similar statement to Lemma 7.16. However, now we require the input graphs to be 2-connected instead of only being connected. This is a necessary condition, since, for example, the star graphs $K_{1,n}$ for $n \geq 2$ are neither 3-connected nor cycles, but still have only two vertex colours with respect to χ_2.

The route to proving the statement is similar to the one described in Subsection 7.1.2. Still, two colours allowing for more complexity in the graph structure, the statements and proofs become more involved and additional cases need to be considered. We start by adapting several of the auxiliary lemmas given in the previous subsection to the setting of two vertex colours.

Lemma 7.18. *Let G be a connected graph with n vertices and suppose $\{w_1, w_2\}$ is a 2-separator of G. Let C be the vertex set of a connected component of $G - \{w_1, w_2\}$ such that $|C| < \frac{n}{2}$ and let $v \in C$. Then there is no $u \in V(G)$ such that*

1. $\mathrm{dist}(u,v) \leq 2$,

2. $\chi_{G,2}(u,u) = \chi_{G,2}(v,v)$, *and*

3. $\operatorname{dist}(u,w_i) \leq \operatorname{dist}(v,w_i) - 2$ *for both* $i \in \{1,2\}$*.*

Proof: We use a similar argument as in the proof of Lemma 7.12. Suppose towards a contradiction that such a vertex $u \in V(G)$ exists. For all $w \in V(G)$, we have $|\operatorname{dist}(v,w) - \operatorname{dist}(u,w)| \leq 2$ by Part 1. Furthermore, we have $\sum_{w \in V(G)} \big(\operatorname{dist}(v,w) - \operatorname{dist}(u,w)\big) = 0$ because $D(v) = D(u)$ due to Part 2. But $\operatorname{dist}(v,w) \geq \operatorname{dist}(u,w) + 2$ for all $w \in V(G) \setminus C$, and $|V(G) \setminus C| > \frac{n}{2}$. This is a contradiction. □

Lemma 7.19. *Let $G = (U,V,E)$ be a 2-connected bipartite graph with n vertices and the following properties.*

1. G has a 2-separator w_1w_2 with $w_1 \in U$ and $w_2 \in V$,

2. $\chi_{G,2}(u,u) = \chi_{G,2}(u',u')$ *for all* $u,u' \in U$*, and*

3. $\chi_{G,2}(v,v) = \chi_{G,2}(v',v')$ *for all* $v,v' \in V$*.*

Let $d := \operatorname{dist}(w_1,w_2)$ and let C be the vertex set of a connected component of $G - \{w_1,w_2\}$ of size $|C| \leq \frac{n-2}{2}$. Then $\operatorname{dist}(v,w_i) \leq d+1$ for all $v \in C$ and $i \in \{1,2\}$.

Proof: Since G is bipartite, d is odd. By symmetry, it suffices to show the statement for $i = 2$. For $u \in U \cap C$, we prove by induction on $\ell := \operatorname{dist}(u,w_1)$ that $\operatorname{dist}(u,w_2) \leq d$. Then the statement follows, because every $v \in V \cap C$ is connected to some $u \in U \cap C$.

For $\ell = 0$, it holds that $u = w_1$ and $\operatorname{dist}(w_1,w_2) = d$. So suppose the statement holds for all $u \in U \cap C$ such that $\operatorname{dist}(u,w_1) \leq \ell$, and pick $u' \in U \cap C$ with $\operatorname{dist}(u',w_1) = \ell + 2$. Let $u \in U \cap C$ such that $\operatorname{dist}(u,w_1) \leq \ell$ and $\operatorname{dist}(u,u') \leq 2$. Then $\operatorname{dist}(u',w_2) \leq \operatorname{dist}(u,w_2) + 1 \leq d+1$ by Lemma 7.18 and the induction hypothesis. But since G is bipartite and we know that $u' \in U$ and $w_2 \in V$, we have that $\operatorname{dist}(u',w_2)$ is odd and thus, $\operatorname{dist}(u',w_2) \leq d$. □

Lemma 7.20. *Let $G = (U,V,E)$ be a 2-connected bipartite graph with n vertices and the following properties.*

1. G has a 2-separator w_1w_2,

2. $\chi_{G,2}(u,u) = \chi_{G,2}(u',u')$ *for all* $u,u' \in U$*, and*

3. $\chi_{G,2}(v,v) = \chi_{G,2}(v',v')$ *for all* $v,v' \in V$*.*

Then $\{w_1,w_2\} \notin E(G)$*.*

Proof: Since G is bipartite and by symmetry, we only need to consider the case that $w_1 \in U$ and $w_2 \in V$. Suppose towards a contradiction that $\{w_1, w_2\} \in E(G)$. Let C be the vertex set of a connected component of $G - \{w_1, w_2\}$ such that $|C| \leq \frac{n-2}{2}$. For $i, j \in \mathbb{N}_0$, let

$$C_{i,j} := \{v \in C \mid \text{dist}(v, w_1) = i \text{ and } \text{dist}(v, w_2) = j\}.$$

By Lemma 7.19, we deduce $C_{i,j} = \emptyset$ unless $(i,j) \in \{(1,1),(1,2),(2,1),(2,2)\}$. Furthermore, $C_{1,1} = C_{2,2} = \emptyset$, since G is bipartite.

Note that $C_{1,2} \cup \{w_2\} \subseteq N(w_1)$. Moreover, since G is 2-connected, w_1 must have a neighbour in $V(G) \setminus (C \cup \{w_1, w_2\})$. Thus $\deg(w_1) \geq |C_{1,2}| + 2$. Let $v \in C_{2,1} \subseteq U$. Then all neighbours of v are contained in $C_{1,2} \cup \{w_2\}$, since $C_{1,1} = C_{2,2} = \emptyset$. Hence, $\deg(v) \leq |C_{1,2}| + 1 < \deg(w_1)$, which contradicts Condition 2. □

Lemma 7.21. *Let G be a graph and suppose that there are $c_1 \neq c_2$ such that $\{\chi_{G,2}(v,v) \mid v \in V(G)\} = \{c_1, c_2\}$. Let $U := \{u \in V(G) \mid \chi_{G,2}(u,u) = c_1\}$ and $V := \{v \in V(G) \mid \chi_{G,2}(v,v) = c_2\}$. Also let $U_1, \dots, U_k$ be the vertex sets of the connected components of $G[U]$ and let $V_1, \dots, V_\ell$ be the vertex sets of the connected components of $G[V]$. Let G' be the graph with $V(G') = \{U_1, \dots, U_k, V_1, \dots, V_\ell\}$ and $\{U_i, V_j\} \in E(G')$ if and only if there are $u \in U_i$, $v \in V_j$ such that $\{u, v\} \in E(G)$.*

Then it holds that $\chi_{G',2}(U_i, U_i) = \chi_{G',2}(U_j, U_j)$ for all $i, j \in [k]$, and $\chi_{G',2}(V_i, V_i) = \chi_{G',2}(V_j, V_j)$ for all $i, j \in [\ell]$.

Proof: We prove the following statement for all $W_1, \dots, W_4 \in V(G')$ by induction on the number of iterations r of the 2-dimensional Weisfeiler-Leman algorithm. In the remainder of this proof, we drop the subscript 2 for readability. If $\chi^r_{G'}(W_1, W_2) \neq \chi^r_{G'}(W_3, W_4)$, then

$$\big\{\chi_G(w_1, w_2) \,\big|\, w_1 \in W_1, w_2 \in W_2\big\} \cap \big\{\chi_G(w_3, w_4) \,\big|\, w_3 \in W_3, w_4 \in W_4\big\} = \emptyset.$$

First observe that this implies the statement of the lemma when setting $W_1 = W_2 = U_i$ and $W_3 = W_4 = U_j$, and similarly for $W_1 = W_2 = V_i$ and $W_3 = W_4 = V_j$.

For $r = 0$, the statement is simple. Indeed, $\chi^r_{G'}(W_1, W_2) \neq \chi^r_{G'}(W_3, W_4)$ if and only if $\{W_1, W_2\} \in E(G) \not\Leftrightarrow \{W_3, W_4\} \notin E(G)$ or $W_1 = W_2 \not\Leftrightarrow W_3 = W_4$. In both cases, the statement follows from the fact that the 2-dimensional Weisfeiler-Leman algorithm is aware of the connected components $U_1, \dots, U_k, V_1, \dots, V_\ell$.

For the inductive step, suppose $r \geq 0$ and pick four sets $W_1, \dots, W_4 \in V(G')$ such that $\chi^{r+1}_{G'}(W_1, W_2) \neq \chi^{r+1}_{G'}(W_3, W_4)$. If $\chi^r_{G'}(W_1, W_2) \neq \chi^r_{G'}(W_3, W_4)$, the statement immediately follows from the induction hypothesis. Hence, there are colours c_1 and c_2 such that $|M| \neq |M'|$, where

$$M := \big\{W' \,\big|\, \chi^r_{G'}(W', W_2) = c_1,\, \chi^r_{G'}(W_1, W') = c_2\big\}$$

and

$$M' := \{W' \mid \chi^r_{G'}(W', W_4) = c_1,\, \chi^r_{G'}(W_3, W') = c_2\}.$$

Let $w_i \in W_i$ for $i \in [4]$. It suffices to argue that

$$\{\!\!\{(\chi_G(w, w_2), \chi_G(w_1, w)) \mid w \in V(G)\}\!\!\} \neq \{\!\!\{(\chi_G(w, w_4), \chi_G(w_3, w)) \mid w \in V(G)\}\!\!\}.$$

But this follows from the induction hypothesis and that $|M| \neq |M'|$. □

Now, similarly as in Theorem 7.17, we can draw strong conclusions about the global structure of G. To finalise the proof that the k-dimensional Weisfeiler-Leman algorithm detects k-separators, it remains for us to show Theorem 7.6, i.e. that every 2-connected graph with a 2-separator and just two vertex colours with respect to χ_2 is a cycle.

Proof of Theorem 7.6: By Lemma 7.16, without loss of generality, we can assume $\chi_{G,2}(w_1, w_1) \neq \chi_{G,2}(w_2, w_2)$. The statement is proved by induction on the graph order n. For $n \in [4]$, a simple case analysis among the possible graphs G yields the statement.

So let $n \geq 5$. Again, it suffices to prove the statement for the case that G is an n-vertex graph with a maximum edge set that satisfies the requirements of the theorem. Let

$$U := \{u \in V(G) \mid \chi_{G,2}(u, u) = \chi_{G,2}(w_1, w_1)\}$$

and

$$V := \{v \in V(G) \mid \chi_{G,2}(v, v) = \chi_{G,2}(w_2, w_2)\}.$$

Let $U_1, \ldots, U_k$ be the vertex sets of the connected components of $G[U]$ and let $V_1, \ldots, V_\ell$ be the vertex sets of the connected components of $G[V]$. Without loss of generality, assume that $w_1 \in U_1$ and $w_2 \in V_1$. Let C be the vertex set of a connected component of $G - \{w_1, w_2\}$ with size $|C| \leq \frac{n-2}{2}$. Also, let $C' := V(G) \setminus (C \cup \{w_1, w_2\})$.

Claim 1. Suppose that $C \subseteq U_1 \cup V_1$ or $C' \subseteq U_1 \cup V_1$. Then G is a cycle.

Proof of the claim: It is not hard to see that, by reasons of connectivity and since $|U_i| = |U_{i'}|$ and $|V_j| = |V_{j'}|$ for all i, j, it suffices to show that $|U_1| \leq 2$ and $|V_1| \leq 2$.

Suppose $C \subseteq U_1 \cup V_1$. Then $C \cap (U_1 \cup V_1) \neq \emptyset$. By symmetry, we may assume that there exists a vertex $u_1 \in U_1 \cap C$ with $\{u_1, w_1\} \in E(G)$. We first argue that $U_1 \cap C' = \emptyset$. Towards a contradiction, suppose that there exists $u_1' \in U_1 \cap C'$. Then w_1 is a cut vertex in $G[U_1]$. Since the 2-dimensional Weisfeiler-Leman algorithm distinguishes arcs within connected components from arcs between different connected components (see Theorem 7.2), by Corollary 7.3, all vertices in U_1 must be cut vertices, in addition to $G[U_1]$ being connected. It is easy to see that there is no connected graph in which all vertices are cut vertices. Analogously, we cannot have that both $V_1 \cap C \neq \emptyset$ and $V_1 \cap C' \neq \emptyset$ hold.

Thus, $U_1 \cap C' = \emptyset$. First assume that $V_1 \cap C = \emptyset$. In this case, we have $C \cup \{w_1\} = U_1$ and, by regularity, $\{u, w_2\} \in E(G)$ for all $u \in U_1$. In particular, every vertex in U has exactly one neighbour in V. However, for w_1, this neighbour must be distinct from w_2 because it must lie in C'. If $|U_1| \geq 3$, this results in $\chi_{G,2}(w_1, w_1) \neq \chi_{G,2}(u_1, u_1)$, a contradiction.

Now assume that $V_1 \cap C' = \emptyset$. Note that for all $j \neq 1$ and all $v \in V_j$, we have that $\{u_1, v\} \notin E(G)$. Thus, $|\{j \mid \exists v \in V_j \colon \{u_1, v\} \in E(G)\}| = 1$ by the connectivity of G. Since $\chi_{G,2}(u_1, u_1) = \chi_{G,2}(u, u)$ for all $u \in U$, every vertex in U has neighbours in exactly one of the sets V_j. Moreover, for all $u \in U_1 \setminus \{w_1\}$, this set is V_1. However, for $w_1 \in U_1$, it is some $V_j \subseteq C'$. Indeed, since $U_1 \cap C' = \emptyset$, the vertex w_1 is the only candidate in $U \cap (C \cup \{w_1\})$ to be adjacent to a vertex in C'.

Suppose there is another vertex $u_2 \in U_1$ with $w_1 \neq u_2 \neq u_1$. Then for every vertex $v_1 \in V_1$, the colours $\chi_{G,2}(u_1, v_1)$ and $\chi_{G,2}(u_2, v_1)$ encode that there exist vertices $v, v' \in V_1$ (possibly equal) such that $\{u_1, v\}, \{u_2, v'\} \in E(G)$. The existence of such a v_1 is thus encoded in $\chi_{G,2}(u_1, u_2)$. However, for w_1 and any other vertex in U_1, there is no equivalent vertex in V. Thus, w_1 is not incident with any edge of colour $\chi_{G,2}(u_1, u_2)$. Hence, $\chi_{G,2}(u_1, u_1) \neq \chi_{G,2}(w_1, w_1)$, which contradicts the assumptions. Therefore $|U_1| \leq 2$. Analogously, it can be shown that $|V_1| \leq 2$.

The case $C' \subseteq U_1 \cup V_1$ follows by symmetry. Altogether, we can deduce that G has to be a cycle. ⌟

So for the remainder of the proof, we assume that $C \nsubseteq U_1 \cup V_1$ and $C' \nsubseteq U_1 \cup V_1$. Note that, since we have assumed $w_1 \in U_1$ and $w_2 \in V_1$, this implies that $G[U]$ and $G[V]$ consist of at least two connected components each (again using the fact that $G[U]$ and $G[V]$ do not contain any cut vertices).

Let G' be the graph with $V(G') = \{U_1, \dots, U_k, V_1, \dots, V_\ell\}$ and $\{U_i, V_j\} \in E(G')$ if and only if there are $u \in U_i$ and $v \in V_j$ such that $\{u, v\} \in E(G)$. We argue that G' satisfies the requirements of the theorem. First note that $G' - \{U_1, V_1\}$ is not connected and hence, $\{U_1, V_1\}$ forms a 2-separator of G'. Also, $\chi_{G',2}(U_i, U_i) = \chi_{G',2}(U_1, U_1)$ for all $i \in [k]$, and $\chi_{G',2}(V_j, V_j) = \chi_{G',2}(V_1, V_1)$ for all $j \in [\ell]$ by Lemma 7.21.

Clearly, G' is connected because G is. So it remains to argue that G' is 2-connected. If not, then G' contains a cut vertex. Without loss of generality, assume that U_i for a certain $i \in [k]$ is a cut vertex. Then, since the 2-dimensional Weisfeiler-Leman algorithm recognises cut vertices (see Corollary 7.3) and by Lemma 7.21, every vertex U_i with $i \in [k]$ is a cut vertex of G'. Considering the cut tree of G' (i.e. the tree in which every cut vertex and every 2-connected component forms a vertex), it is not hard to see that this implies that every V_j for $j \in [\ell]$ has only one neighbour. But this is only possible if $C \subseteq V_1$ or $C' \subseteq V_1$, and we assumed the contraries of both cases.

Suppose that $|V(G')| < |V(G)|$. Then, by the induction hypothesis, the graph G' is a cycle. Thus, in G', every U_i is adjacent to exactly one or two connected components of $G[V]$.

We first consider the subcase that both $C \cap U_1 = \emptyset$ and $C \cap V_1 = \emptyset$ hold.

The vertex w_1 is adjacent to vertices in a connected component with vertex set $V_j \subseteq C$, whereas all vertices $u \in U_1$ with $u \neq w_1$ must be adjacent to the same connected component with vertex set $V_{j'} \subseteq (C' \cup \{w_2\})$. In particular, we have $j \neq j'$. Assuming $|U_1| > 2$, similarly as in the proof of Claim 1, we reach a contradiction considering $\chi_{G,2}(w_1, w_1)$. By symmetry, we cannot have $|V_1| > 2$ either. The subcase that $C' \cap U_1 = \emptyset$ and $C' \cap V_1 = \emptyset$ can be treated analogously.

Thus, suppose without loss of generality that $C \cap U_1 \neq \emptyset$ and $C' \cap V_1 \neq \emptyset$. (The case that $C' \cap U_1 \neq \emptyset$ and $C \cap V_1 \neq \emptyset$ hold follows by symmetry.)

Since we cannot have that $C = U_1 \setminus \{w_1\}$, there must be a connected component with vertex set $V_j \subset C$ for a certain $j \neq 1$ such that for every $u_1 \in U_1 \cap C$, we have $N(u_1) \cap V \subseteq V_j$. However, all neighbours of w_1 in V are contained in a $V_{j'} \subseteq C' \cup \{w_2\}$. In particular, $j \neq j'$. Again, we reach a contradiction when assuming that $|U_1| > 2$ and, by symmetry, also for the assumption $|V_1| > 2$.

Thus, if $|V(G')| < |G|$, then G is a cycle.

Now assume $|V(G')| = |G|$. This means that G is a bipartite graph with bipartition (U, V) and all U_i and all V_j are singletons. From Lemma 7.20, we know $\{w_1, w_2\} \notin E(G)$. Let $d \coloneqq \operatorname{dist}(w_1, w_2)$. Note that d is odd and thus, $d \geq 3$.

Claim 2. Let $u, v \in V(G)$ such that $\operatorname{dist}(u, v) < d$. Then there is a unique shortest path from u to v.

Proof of the claim: Suppose the statement does not hold and let $\ell < d$ be the minimal distance for which the claim is violated, i.e. there are vertices $u, v \in V(G)$ such that there are two paths of length $\ell = \operatorname{dist}(u, v)$ from u to v. Also, let $E' \coloneqq \big\{\{u', v'\} \,\big|\, \chi_{G,2}(u, v) = \chi_{G,2}(u', v')\big\}$ and consider the graph $G' = \big(V(G), E(G) \cup E'\big)$. We argue that G' still satisfies the assumptions of the theorem, which contradicts the edge maximality of G.

First, the colouring $\chi_{G,2}$ is also a stable colouring for G', which implies that $\chi_{G,2}$ refines the colouring $\chi_{G',2}$. In particular, for every $v \in V(G)$, there is an $i \in \{1, 2\}$ such that $\chi_{G,2}(v, v) = \chi_{G,2}(w_i, w_i)$.

Now let $\{u', v'\} \in E'$. Then $\operatorname{dist}(u', v') = \ell$ and there are at least two different walks of length ℓ from u' to v', because the same statement holds for u and v. Due to the minimality of ℓ, the two walks are vertex-disjoint paths. If u' and v' lie in different connected components of $G - \{w_1, w_2\}$, then one of the two paths must pass through w_1 and one through w_2, forming a cycle of length $2\ell < 2d$. This implies $\operatorname{dist}(w_1, w_2) < d$, a contradiction. Thus, we conclude that there is a connected component of $G - \{w_1, w_2\}$ with vertex set C such that $u', v' \in C \cup \{w_1, w_2\}$. But this means that $\{w_1, w_2\}$ is also a 2-separator of the graph G'. ⌟

For $i, j \in \mathbb{N}_0$, let

$$C_{i,j} \coloneqq \{v \in C \mid \operatorname{dist}(v, w_1) = i \text{ and } \operatorname{dist}(v, w_2) = j\}.$$

By Lemma 7.19, we conclude that $C_{i,j} = \emptyset$ unless $i, j \leq d + 1$. Furthermore, by the definition of d, it holds that $C_{i,j} = \emptyset$ unless $i + j \geq d$. Since G is bipartite, we know that $C_{i,j} = \emptyset$ whenever $i + j$ is even.

Claim 3. Suppose $d \geq 7$. Then G is a cycle.

Proof of the claim: We argue that $\deg(u) = 2$ for every $u \in U$. Analogously, one can prove that $\deg(v) = 2$ for every $v \in V$, which together means that G is a cycle.

First suppose that $C_{4,d-4} \neq \emptyset$ and let $u \in C_{4,d-4}$. Note that $u \in U$, since 4 is even. Then $N(u) \subseteq C_{3,d-3} \cup C_{5,d-5} \cup C_{5,d-3}$. Moreover, $|N(u) \cap (C_{3,d-3} \cup C_{5,d-5})| = 2$, otherwise there would be two shortest paths from u to w_1 or two shortest paths from u to w_2, contradicting Claim 2. Now suppose towards a contradiction that $\deg(u) > 2$. Then there is a $v \in N(u) \cap C_{5,d-3}$. We have that $N(v) \subseteq C_{4,d-4} \cup C_{4,d-2} \cup C_{6,d-4} \cup C_{6,d-2}$. Using Claim 2, we conclude that $|N(v) \cap (C_{4,d-4} \cup C_{4,d-2})| \leq 1$ and $|N(v) \cap (C_{4,d-4} \cup C_{6,d-4})| \leq 1$. Since $\deg(v) \geq 2$ and $u \in N(v) \cap C_{4,d-4} \neq \emptyset$, it follows that $N(v) \cap C_{6,d-2} \neq \emptyset$. But this contradicts Lemma 7.18. So $\deg(u) = 2$.

Now suppose $C_{4,d-4} = \emptyset$ and hence, $C_{i,d-i} = \emptyset$ for all $i \in [d-1]$. Since $N(w_1) \cap C = C_{1,d-1} \cup C_{1,d+1}$, we conclude that $C_{1,d+1} \neq \emptyset$ and hence, $C_{i,d+2-i} \neq \emptyset$ for all $1 \leq i \leq d+1$.

So pick $u \in C_{4,d-2}$. Then $N(u) \subseteq C_{3,d-1} \cup C_{5,d-3} \cup C_{5,d-1}$. Again, $|N(u) \cap (C_{3,d-1} \cup C_{5,d-3})| = 2$, using Claim 2. Suppose towards a contradiction that $\deg(u) > 2$. Then there is a vertex $v \in N(u) \cap C_{5,d-1}$. We have that $N(v) \subseteq C_{4,d-2} \cup C_{4,d} \cup C_{6,d-2} \cup C_{6,d}$. By Claim 2, we conclude that $|N(v) \cap (C_{4,d-2} \cup C_{4,d})| \leq 1$ and $|N(v) \cap (C_{4,d-2} \cup C_{6,d-2})| \leq 1$. Since $\deg(v) \geq 2$, it follows that $N(v) \cap C_{6,d} \neq \emptyset$. As before, this contradicts Lemma 7.18. ⌟

Claim 4. Suppose $d = 5$. Then G is a cycle.

Proof of the claim: Consider Figure 7.3 for a visualisation of the situation. Again, it suffices to show $\deg(u) = 2$ for every $u \in U$, since showing $\deg(v) = 2$ for every $v \in V$ works analogously. If $C_{2,3} \neq \emptyset$, we can argue similarly as in the proof of Claim 3 that there are a vertex $u \in C_{2,3}$ and a vertex $v \in N(u) \cap C_{3,4}$ with $N(v) \cap C_{4,5} \neq \emptyset$, which contradicts Lemma 7.18.

Thus, we can assume that $C_{2,3} = \emptyset$. Hence, we have $C_{i,d-i} = \emptyset$ for all $i \in [4]$ and therefore $C_{1,6} \neq \emptyset$ and $C_{i,7-i} \neq \emptyset$ for all $i \in [6]$. Analogously as in the proof of Claim 3, there are vertices $u \in C_{4,3}$ and $v \in N(u) \cap C_{5,4}$, supposing $\deg(u) > 2$. We have $N(v) \subseteq C_{4,3} \cup C_{4,5} \cup C_{6,3} \cup C_{6,5}$. By Claim 2, we have $|N(v) \cap (C_{4,3} \cup C_{6,3})| \leq 1$. We would like to apply the same argument to $N(v) \cap (C_{4,3} \cup C_{4,5})$. However, since $\operatorname{dist}(v, w_1) = 5 = d$, we cannot apply Claim 2 immediately. Still, note that the

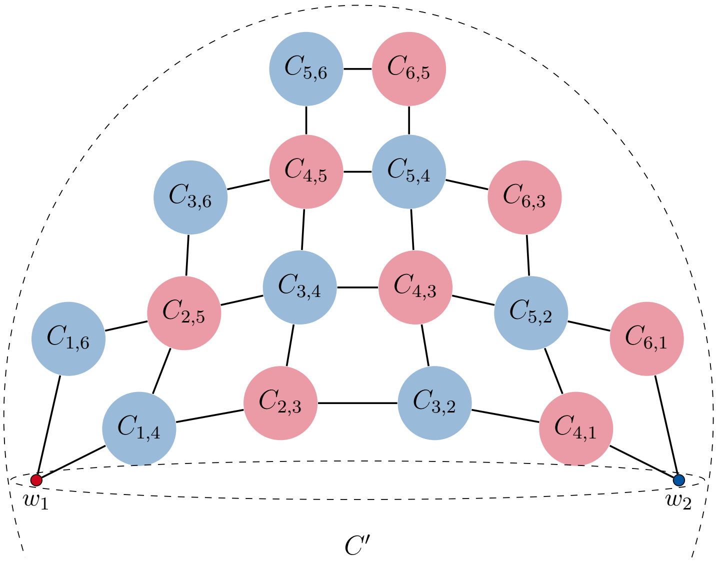


Figure 7.3: Visualisation of the sets $C_{i,j}$ for $d = 5$ in the proof of Theorem 7.6. Each edge between two sets indicates that there may be edges connecting vertices from the two sets.

shortest path from w_1 to w_2 with all internal vertices in C has length 7. Therefore, analogously as in the proof of Claim 2, using the edge maximality of G, we can show that there cannot be cycles of length at most 10 in G and that thus, there must be a unique shortest path from v to w_1. In particular, this implies that $|N(v) \cap (C_{4,3} \cup C_{4,5})| \leq 1$. We can conclude the proof analogously as the one for Claim 3. ⌟

Claim 5. Suppose $d = 3$. Then G is a cycle.

Proof of the claim: For the following arguments, see also Figure 7.4. First observe that all vertices of C must be contained in one of the sets $C_{i,j}$ with $(i, j) \in \{(1, 2), (2, 1), (1, 4), (2, 3), (3, 2), (4, 1), (3, 4), (4, 3)\}$.

If w_2 has only one neighbour v in C', then $\{w_1, v\}$ is a uni-coloured 2-separator. Consider the graph $\widehat{G}$ with vertex set U and an edge between every pair of vertices u and v if there is a path from u to v of length 2 (i.e. via a vertex in V) in G. Since the 2-dimensional Weisfeiler-Leman algorithm implicitly detects such paths, we can apply Theorem 7.17. It is easy to see that $\widehat{G}$ is connected and not 3-connected. Thus, it must be a cycle. In particular, every vertex in $\widehat{G}$ has degree 2. Therefore, the position of each vertex in V is uniquely determined: the graph G is a subdivision of $\widehat{G}$. Thus, G is a cycle.

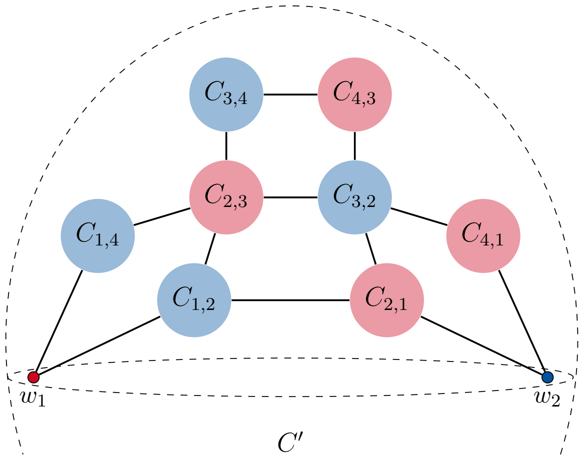


Figure 7.4: Visualisation of the sets $C_{i,j}$ for $d = 3$ in the proof of Theorem 7.6. Each edge between two sets indicates that there may be edges connecting vertices from the two sets.

Now assume

$$\deg(w_2) > |N(w_2) \cap C'| \geq 2. \tag{7.1}$$

First suppose $C_{1,2} \neq \emptyset$ and, equivalently, $C_{2,1} \neq \emptyset$. For every vertex $u \in C_{2,1}$, we have $N(u) \subseteq C_{1,2} \cup C_{3,2} \cup \{w_2\}$. By Claim 2, it holds that $|N(u) \cap C_{1,2}| = 1$. Thus, since $\{u, w_2\} \in E(G)$, we have $|N(u) \cap C_{3,2}| = r_U - 2$, where $r_U = \deg(u) = \deg(w_1)$. Furthermore, again by Claim 2, for $u' \in C_{2,1}$ with $u' \neq u$, we have $N(u) \cap N(u') = \{w_2\}$ and thus, $|N(C_{2,1}) \cap C_{3,2}| = |C_{2,1}| \cdot (r_U - 2)$. Moreover, by a similar reasoning, $|N(C_{4,1}) \cap C_{3,2}| = |C_{4,1}| \cdot (r_U - 1)$. Every vertex in $C_{3,2}$ has distance 2 to w_2 and thus has a neighbour in $C_{2,1} \cup C_{4,1}$. Note that $N(w_2) \cap C = C_{2,1} \cup C_{4,1}$. Now since, by Claim 2, we have $(N(C_{2,1}) \cap N(C_{4,1})) \setminus \{w_2\} = \emptyset$, we obtain that $|C_{3,2}| = |C_{2,1}| \cdot (r_U - 2) + |C_{4,1}| \cdot (r_U - 1) < (|C_{2,1}| + |C_{4,1}|) \cdot (r_U - 1) \leq (r_V - 2) \cdot (r_U - 1)$ by (7.1), where $r_V = \deg(w_2)$.

Let $v \in C_{1,2}$. Similarly as above, $|N(v) \cap C_{2,3}| = r_V - 2$. Furthermore, for every vertex $u \in N(v) \cap C_{2,3}$, by Claim 2 and Lemma 7.18, we have $N(u) \subseteq C_{1,2} \cup C_{3,2}$ and $|N(u) \cap C_{1,2}| = 1$. Again using Claim 2, every two vertices in $N(v) \cap C_{2,3}$ must have disjoint neighbourhoods in $C_{3,2}$. Thus, we obtain that $|C_{3,2}| \geq (r_V - 2)(r_U - 1)$.

Altogether $(r_V - 2)(r_U - 1) \leq |C_{3,2}| < (r_V - 2)(r_U - 1)$, which yields a contradiction. Therefore, $C_{1,2} = C_{2,1} = \emptyset$. Hence, it must hold that $C_{1,4} \neq \emptyset$ and, consequently, also $C_{i,5-i} \neq \emptyset$ for every $i \in [4]$. Using the same arguments as before, we have $|C_{3,2}| \leq (r_V - 2)(r_U - 1)$. Every vertex $v' \in C_{3,2}$ has at least one neighbour in the set $C_{4,1}$ and at least one neighbour in $C_{2,3}$. Since for every $u' \in C_{4,3}$, there is a $v' \in C_{3,2} \cap N(u')$, this implies $|C_{4,3}| \leq (r_V - 2)(r_U - 1)(r_V - 2)$.

Let $v \in C_{1,4}$ (recall that $C_{1,4} \neq \emptyset$). Then $|N(v) \cap C_{2,3}| = r_V - 1$ and for every vertex $u' \in N(v) \cap C_{2,3}$, we have by Claim 2 that $|N(u') \cap (C_{3,4} \cup C_{3,2})| = r_U - 1$. Again by Claim 2, for a second vertex $u'' \in N(v) \cap C_{2,3}$ with $u'' \neq u'$, it must hold that

$$N(u') \cap (C_{3,4} \cup C_{3,2}) \cap N(u'') = \emptyset. \tag{7.2}$$

Let $S := N(N(v)) \cap (C_{3,4} \cup C_{3,2})$. Note that the shortest path from w_1 to w_2 with all internal vertices in C has length 5. Therefore, using the edge maximality of G, analogously as in the proof of Claim 2, there cannot be any cycles of length 6 in G. In particular, this implies that for every pair $v', v'' \in S$ with $v' \neq v''$, we have $N(v') \cap N(v'') \cap C_{4,3} = \emptyset$.

Thus, Equation (7.2) implies that $|C_{4,3}| \geq (r_V - 1)(r_U - 1)(r_V - 2)$, since every vertex in S has at most two neighbours not contained in $C_{4,3}$.

Altogether, we obtain $(r_V - 1)(r_U - 1)(r_V - 2) \leq |C_{4,3}| \leq (r_V - 2)(r_U - 1)(r_V - 2)$, which is a contradiction. This concludes the proof. □

7.2 Decompositions

In the remainder of this chapter, we show that for $k \geq 2$, the k-dimensional Weisfeiler-Leman algorithm implicitly computes the decomposition into the 3-connected components of a given graph. We intend to follow an inductive approach, similarly as in the well-known proof that the 1-dimensional Weisfeiler-Leman algorithm identifies all trees: for $k \in \{2, 3\}$, the decomposition of a graph into its k-connected components has a tree-like structure. We consider these trees and aim to cut off the leaves of the decomposition trees of 2- and 3-connected components, respectively, in an isomorphism-invariant way. By adapting the colourings of the smaller decomposition trees accordingly, we ensure that we never exceed the expressive power of the 2-dimensional Weisfeiler-Leman algorithm, while at the same time maintaining all isomorphism-invariant information about the leaves. By induction, this will finally enable us to show that, in order to determine an upper bound on the Weisfeiler-Leman dimension of a (minor-closed) graph class, it suffices to consider the edge-coloured 3-connected graphs in the class (see Theorem 7.37).

As a preparation for the reductions to 2- and 3-connected graphs, this section introduces the decompositions and the constructed colourings of the decomposition trees and collects several structural observations about them. We employ these contents in Sections 7.3 and 7.4. In Section 7.3, we show that the 2-dimensional Weisfeiler-Leman algorithm computes the decomposition into the 2-connected components and then, in Section 7.4, we deduce the same statement for the decomposition into the 3-connected components.

We start off with a basic observation. Let $\mathcal{G}$ be a hereditary graph class, i.e. a graph class that is closed under taking induced subgraphs. When intending to determine the dimension of the Weisfeiler-Leman algorithm that is needed to identify graphs from $\mathcal{G}$, the corollary below allows us to restrict ourselves to connected graphs from $\mathcal{G}$ (assuming Weisfeiler-Leman dimension at least 2). In other words, already the 2-dimensional Weisfeiler-Leman algorithm implicitly computes the decomposition of a graph into its connected components, as stated in the following corollary.

Corollary 7.22. *Let $\mathcal{G}$ be a hereditary graph class. The Weisfeiler-Leman dimension of $\mathcal{G}$ is at most $\max\{2, d\}$, where d is a bound on the Weisfeiler-Leman dimension of every connected graph in $\mathcal{G}$.*

Proof: This follows from the observation that for two non-isomorphic connected components, the sets of colours which the 2-dimensional Weisfeiler-Leman algorithm computes for their vertices are disjoint assuming that it distinguishes all non-isomorphic connected graphs. □

We collect some further notions and facts about graph decompositions that are useful for our purposes.

For a graph G, define the set $P(G)$ to consist of the pairs (S, K), where S is a separator of G of minimum cardinality and $K \subseteq V(G) - S$ is the vertex set of a connected component of $G - S$.

Note that, if G is a connected graph that is not 2-connected, then $P(G)$ is the set of pairs $(\{s\}, K)$ where s is a cut vertex and $G[K]$ a connected component of $G - \{s\}$. In this case, we also write (s, K) instead of $(\{s\}, K)$. If G is 2-connected but not 3-connected, all separators in $P(G)$ have size 2.

There is a natural partial order $\leq$ on $P(G)$ with respect to inclusion in the second component, i.e. we can define:

$$(S, K) \leq (S', K') \iff K \subseteq K'.$$

We define $P_0(G)$ to be the set of minimal elements of $P(G)$ with respect to $\leq$. For a connected graph G which is not 3-connected, the elements of $P_0(G)$ correspond to the leaves in a suitable decomposition tree (i.e. the decomposition into 2- or 3-connected components).

Remark 7.23. It immediately follows from the definitions that the sets $P(G)$ and $P_0(G)$ (and the corresponding partial orders) are preserved under isomorphisms.

Note that $P_0(G)$ is non-empty whenever G is not a complete graph. Also note that if G is not 2-connected, then for two distinct elements (S, K) and (S', K') in $P_0(G)$, we have $K \cap K' = \emptyset$. Furthermore, in the case that G is connected but not 2-connected, the set $P_0(G)$ contains exactly the pairs (s, K) for which s is a cut vertex and $G[K]$ a connected component of $G - \{s\}$ that does not contain a cut vertex of G. These two observations can be generalised to graphs with a higher connectivity, but for this we need an additional requirement on the minimum degree, as stated in the following lemma.

Lemma 7.24. *Let $k \in \mathbb{N}$ and let G be a graph that is k-connected but not $(k+1)$-connected and has minimum degree at least $\frac{3k-1}{2}$.*

1. *If $(S, K) \in P_0(G)$ and $(S', K') \in P(G)$ are distinct, then $K \subseteq K'$ or $(K \cup S) \cap (K' \cup S') = S \cap S'$.*
2. *If $(S, K), (S', K') \in P_0(G)$ are distinct, then $(K \cup S) \cap (K' \cup S') = S \cap S'$.*
3. *A pair $(S, K) \in P(G)$ is contained in $P_0(G)$ if and only if there is no separator S' of G of minimum cardinality with $S' \cap K \neq \emptyset$.*

Proof: To show Part 1, assume that $(S, K) \in P_0(G)$ and $(S', K') \in P(G)$ are distinct. Note that $S = N(K)$ and $S' = N(K')$, since S and S' are separators of minimal size.

Suppose that $K \not\subseteq K'$ and let $u_0 \in K \setminus K'$. First assume there exists a $v_0 \in (K \cup S) \cap K'$. Since K is connected and $S = N(K)$, there exists a path P from v_0 to u_0 that does not intersect $\overline{K} := V(G) \setminus (K \cup S)$ and whose inner vertices all lie in K. Since $u_0 \notin K'$, but $v_0 \in K'$, the path must contain vertices that lie in $S' \cap K$. Let v be the last vertex on P that lies in K'. Then its successor on P is some vertex $u \in K \cap S'$. Therefore, the graph $G - \{u\}$ is at most $(k-1)$-connected. On the other hand, u lies in K. By [98, Corollary 1], when removing a vertex contained in a minimal component from a k-connected graph with minimum vertex degree at least $\frac{3k-1}{2}$, the graph remains k-connected. Therefore, the graph $G - \{u\}$ is k-connected, yielding a contradiction.

Thus, we now know that $(K \cup S) \cap K' = \emptyset$. It suffices to show that $K \cap S' = \emptyset$. Supposing otherwise, choose an element $s \in K \cap S'$. This element has a neighbour $w \in K'$, since $N(K') = S'$. We cannot have that $w \in \overline{K}$, since K and $\overline{K}$ are separated by S. Thus, $w \in (K \cup S) \cap K'$, contradicting $(K \cup S) \cap K' = \emptyset$.

Part 2 follows by applying Part 1 twice.

We now show Part 3. If $(S, K) \in P(G)$ is not minimal, then there exists $(S', K') \in P(G)$ with $K' \subsetneq K$. Then $S' \subseteq K \cup S$, since $S' = N(K')$ and $S = N(K)$. But $S \neq S'$, which shows that $S' \cap K \neq \emptyset$. Conversely, towards a contradiction, suppose $(S, K) \in P_0(G)$ and that there is a minimum-size separator S' of G with $S' \cap K \neq \emptyset$. Let $K' \subseteq V(G \setminus S')$ be a vertex set such that $G[K']$ is a connected component of $G - S'$. Then (S, K) and (S', K') violate Part 1 of the lemma. □

In the following, we present a method to remove the vertices appearing in the second components of pairs in P_0 from the underlying graphs in such a way that for two graphs, the property of being isomorphic is preserved. This will allow us to devise an inductive isomorphism test. We will then show that the 2-dimensional Weisfeiler-Leman algorithm in a sense implicitly computes the decomposition into 3-connected components and performs this induction.

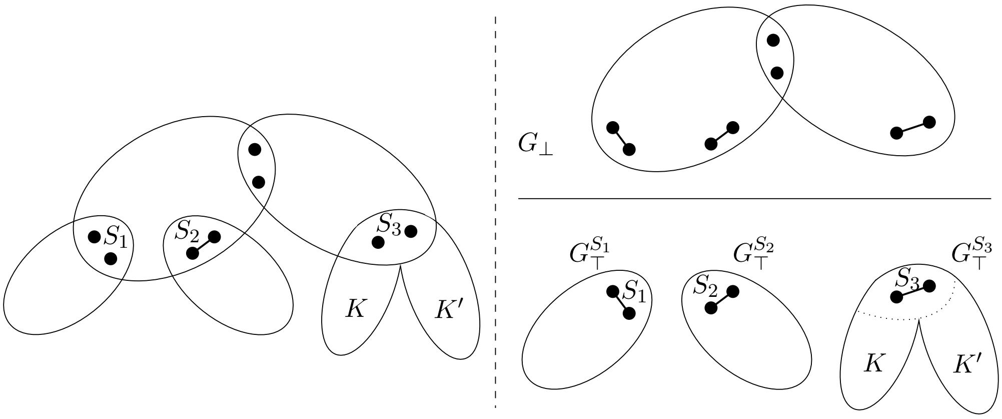


Figure 7.5: The figure illustrates the notions of the graphs $G_\perp$ and $G_\top^S$. In the 2-connected graph on the left, the 2-separators are indicated and the respective decomposed graphs are shown on the right.

For a graph G and a set $S \subseteq V(G)$, we define the graph $G_\top^S$ as consisting of the vertices of S and those appearing together with S in $P_0(G)$. More precisely, $G_\top^S$ is the graph on the vertex set

$$V' = S \cup \bigcup_{(S,K)\in P_0(G)} K$$

with edge set $E' = E(G[V']) \cup \big\{\{s,s'\} \,\big|\, s,s' \in S \text{ and } s \neq s'\big\}$, see Figure 7.5 left and bottom right.[1] Note that if S is not a separator or does not appear as a first component of a tuple in $P_0(G)$, then $V' = S$.

For an edge colouring λ of G, we define an edge colouring $\lambda_\top^S$ for the graph $G_\top^S$ as follows:

$$\lambda_\top^S(v_1,v_2) := \begin{cases} (0,0) & \text{if } \{v_1,v_2\} \subseteq S \text{ and } \{v_1,v_2\} \notin E(G) \\ (\lambda(v_1,v_2),1) & \text{if } \{v_1,v_2\} \subseteq S \text{ and } \{v_1,v_2\} \in E(G) \\ (\lambda(v_1,v_2),2) & \text{otherwise.} \end{cases}$$

If G is a vertex-coloured graph with vertex colouring λ', in order to obtain a colouring for $G_\top^S$, we define an edge colouring λ as $\lambda(v_1,v_2) := \lambda'(v_1)$ and let $\lambda_\top^S$ be as above.

The idea behind $\lambda_\top^S$ is that it maintains all colours present in $G_\top^S$ seen as a subgraph of G, while also encoding the way in which the subgraph is attached to the remainder of G. That is, all λ-colours of vertex pairs in V' are encoded and additionally, $\lambda_\top^S$ is

[1] For the reader familiar with tree decompositions, we remark that this graph corresponds to the torso of the bag $S \cup K$ in a suitable tree decomposition.

aware of the special role of the 2-separator even though in $G^S_\top$, it is not a separator anymore.

For $(S, K) \in P_0(G)$, we also need to be able to speak about the subgraph on vertex set $S \cup K$. For this, we define $G^{(S,K)}_\top := G^S_\top[S \cup K]$, which differs from $G^S_\top$ in that only vertices from S and K are retained. Again, we define a colouring $\lambda^{(S,K)}_\top$, which is simply the restriction of $\lambda^S_\top$ to pairs (v_1, v_2) for which $v_1, v_2 \in S \cup K$.

As a preparation for our inductive proof, we fix the notation for the coloured graphs obtained after cutting off the leaves of the decomposition tree. Given a graph G, we let $G_\perp$ (see Figure 7.5 left and top right) be the graph with vertex set

$$V_\perp := V(G) \setminus \left(\bigcup_{(S,K) \in P_0(G)} K \right)$$

and edge set

$$E_\perp := E(G[V_\perp]) \cup \big\{\{s_1, s_2\} \mid \exists (S,K) \in P_0(G) \text{ s.t. } s_1, s_2 \in S,\ s_1 \neq s_2\big\}.$$

We observe that if G is not 2-connected, then $G_\perp$ is equal to $G[V_\perp]$. In general, if for some k, the graph G is k-connected but not $(k+1)$-connected, and G has minimum degree at least $\frac{3k-1}{2}$, then Lemma 7.24 applies. In particular, the various components whose vertex sets appear in $P_0(G)$ are disjoint. If G is 2-connected but not 3-connected, this implies that $G_\perp$ is a minor of G.

In the following, we restrict our discussions to graphs that are not 3-connected. We explicitly include graphs that are not 2-connected. Given an edge colouring λ of G, we define an edge colouring $\lambda_\perp$ of $G_\perp$ as follows. Assume that $v_1, v_2 \in V(G_\perp)$. Let $S := \{v_1, v_2\}$, possibly with $v_1 = v_2$.

If S is a 2-separator of G, but $S \notin E(G)$, we set

$$\lambda_\perp(v_1, v_2) := \Big(0, \text{ISOTYPE}\Big(\big(G^S_\top, \lambda^S_\top\big)_{(v_1,v_2)}\Big)\Big).$$

Furthermore, if $v_1 = v_2$ or if $\{v_1, v_2\} \in E(G)$, we set

$$\lambda_\perp(v_1, v_2) := \Big(\lambda(v_1, v_2), \text{ISOTYPE}\Big(\big(G^S_\top, \lambda^S_\top\big)_{(v_1,v_2)}\Big)\Big),$$

where the expression $\text{ISOTYPE}((G^S_\top, \lambda^S_\top)_{(v_1,v_2)})$ denotes the isomorphism class of the graph $(G^S_\top, \lambda^S_\top)_{(v_1,v_2)}$ obtained from the edge-coloured graph $(G^S_\top, \lambda^S_\top)$ by individualising the vertices v_1 and v_2 in that order. Thus, $(G^{\{v_1,v_2\}}_\top, \lambda^{\{v_1,v_2\}}_\top)_{(v_1,v_2)}$ and $(G'^{\{v'_1,v'_2\}}_\top, \lambda'^{\{v'_1,v'_2\}}_\top)_{(v'_1,v'_2)}$ have the same isomorphism type if and only if there is an isomorphism from the first graph to the second mapping v_1 to v'_1 and v_2 to v'_2. Note that, by definition, the $\lambda_\perp$-colours of separators of G are distinct from those of other pairs of vertices. Indeed, if $\{v_1, v_2\}$ is not a separator, the graph

$(G_\top^{\{v_1,v_2\}}, \lambda_\top^{\{v_1,v_2\}})_{(v_1,v_2)}$ is a single coloured vertex (if $v_1 = v_2$) or a single coloured edge.

If not stated otherwise, we implicitly assume that for a graph G with initial colouring λ, the corresponding graph $G_\perp$ is a coloured graph with initial colouring $\lambda_\perp$.

The following lemma states that the coloured $\perp$-graphs indeed preserve all information from the original graphs.

Lemma 7.25. *For $k \in \{1,2\}$, if G and G' are k-connected graphs with edge colourings λ and λ', respectively, and G and G' are not $(k+1)$-connected and have minimum degree at least $\frac{3k-1}{2}$, then*

$$(G,\lambda) \cong (G',\lambda') \iff (G_\perp,\lambda_\perp) \cong (G'_\perp,\lambda'_\perp).$$

Proof: ($\Longrightarrow$) Suppose that φ is an isomorphism from (G,λ) to (G',λ'). Since $P_0(G)$ is isomorphism-invariant (Remark 7.23), we know that $\varphi\big(V(G_\perp)\big) = V(G'_\perp)$. To see that φ induces an isomorphism from $(G_\perp,\lambda_\perp)$ to $(G'_\perp,\lambda'_\perp)$, it suffices to observe that the definitions of $G_\perp$ from G and $\lambda_\perp$ from λ are isomorphism-invariant.

($\Longleftarrow$) Conversely, suppose that $\widetilde{\varphi}$ is an isomorphism from $(G_\perp,\lambda_\perp)$ to $(G'_\perp,\lambda'_\perp)$. Define $\{S_1,\dots,S_t\} := \{S \mid \exists K \text{ s.t. } (S,K) \in P_0(G)\}$, i.e. as the set of separators that appear in $P_0(G)$. Since $\widetilde{\varphi}$ respects the colourings $\lambda_\perp$ and $\lambda'_\perp$, we can conclude that

$$\big\{\widetilde{\varphi}(S_1),\dots,\widetilde{\varphi}(S_t)\big\} = \big\{S \bigm| \exists K \text{ s.t. } (S,K) \in P_0(G')\big\}.$$

For each $j \in [t]$, we choose an isomorphism φ_j from $G_\top^{S_j}$ to $G_\top'^{\widetilde{\varphi}(S_j)}$ that maps each $s \in S_j$ to $\widetilde{\varphi}(s) \in \widetilde{\varphi}(S_j)$. We know that such an isomorphism exists because $\widetilde{\varphi}$ respects the colourings $\lambda_\perp$ and $\lambda'_\perp$. Define a map φ from (G,λ) to (G',λ') by setting

$$\varphi(v) := \begin{cases} \widetilde{\varphi}(v) & \text{if } v \in V(G_\perp) \\ \varphi_j(v) & \text{if there is a set } K \subseteq V(G) \text{ with } v \in K \text{ and } (S_j,K) \in P_0(G). \end{cases}$$

This map is well-defined since, by Parts 2 and 3 of Lemma 7.24, the elements in the second components of pairs in $P_0(G)$ are disjoint and not contained in $V(G_\perp)$. Moreover, the map is an isomorphism, since it respects all edges. Finally, by construction, it also respects the colours of vertices and arcs. □

We now have all technical insight into decompositions that we need for the reduction to edge-coloured 3-connected graphs. The reduction proceeds via vertex-coloured 2-connected graphs and the first part of it is treated in the next section.

7.3 Reduction to Vertex-Coloured 2-Connected Graphs

Recall that a *hereditary* graph class is a class of graphs that is closed under taking induced subgraphs. By Corollary 7.22, for a hereditary graph class $\mathcal{G}$ and $k \geq 2$, the k-dimensional Weisfeiler-Leman algorithm distinguishes all non-isomorphic (vertex-coloured) graphs in $\mathcal{G}$ if it distinguishes all non-isomorphic (vertex-coloured) connected graphs in $\mathcal{G}$. (More precisely, by a vertex-coloured graph in $\mathcal{G}$, we mean a coloured graph whose underlying uncoloured graph is contained in $\mathcal{G}$.)

In this section, we show a stronger statement, replacing the assumption on connected graphs with an assumption on 2-connected graphs as follows.

Theorem 7.26. *Let $\mathcal{G}$ be a hereditary graph class. If, for $k \in \mathbb{N}_{\geq 2}$, the k-dimensional Weisfeiler-Leman algorithm distinguishes every two non-isomorphic 2-connected vertex-coloured graphs (G, λ) and (G', λ') with $G, G' \in \mathcal{G}$ from each other, then the k-dimensional Weisfeiler-Leman algorithm distinguishes all non-isomorphic vertex-coloured graphs in $\mathcal{G}$.*

For the rest of this section, let $\mathcal{G}$ be a hereditary graph class. Recall that for a graph $G \in \mathcal{G}$ with an initial vertex colouring or edge colouring λ, the colouring $\chi_{G,k}$ is the stable k-tuple colouring produced by the k-dimensional Weisfeiler-Leman algorithm on (G, λ).

We prove Theorem 7.26 by induction over the sizes of the input graphs. The strategy is to show that on input (G, λ), the Weisfeiler-Leman algorithm implicitly computes the graph $(G_\perp, \lambda_\perp)$, and to then apply Lemma 7.25.

Lemma 7.27. *Let $k \in \mathbb{N}_{\geq 2}$ and assume G and H are connected graphs that are not 2-connected. For vertices $v \in V(G_\perp)$ and $w \in V(H) \setminus V(H_\perp)$, it holds that $\chi_{G,k}(v) \neq \chi_{H,k}(w)$.*

Proof: Note that, for a connected but not 2-connected graph G, a vertex $v \in V(G)$ is in $V(G_\perp)$ if and only if it is a cut vertex or there are at least two cut vertices that lie in the same 2-connected component as v. The equivalent statement holds for H. If v is a cut vertex of G, then the lemma follows immediately from Corollary 7.3.

If v is not a cut vertex, then there are at least two cut vertices u and u' lying in the same 2-connected component as v. Note that there is no such pair of vertices for w. By Corollary 7.3, the vertices u and u' obtain colours distinct from colours of non-cut vertices. Thus, also the colours $\chi_{G,k}(v, u)$ and $\chi_{G,k}(v, u')$ are distinct from all colours of edges from v to non-cut vertices. Moreover, Theorem 7.2 yields that the colours $\chi_{G,k}(v, u)$ and $\chi_{G,k}(v, u')$ encode that v, u, u' all share a common 2-connected component. This information about the existence of such u and u' is contained in the colour $\chi_{G,k}(v)$ and thus, $\chi_{G,k}(v) \neq \chi_{H,k}(w)$. □

Lemma 7.28. *Let $(G, \lambda), (G', \lambda')$ be vertex-coloured graphs with $G, G' \in \mathcal{G}$. Assume that $(s, K) \in P_0(G)$ and $(s', K') \in P_0(G')$. For $k \in \mathbb{N}_{\geq 2}$, suppose the k-dimensional Weisfeiler-Leman algorithm distinguishes all non-isomorphic vertex-coloured 2-connected graphs in $\mathcal{G}$. Assume that there is no isomorphism from $(G_\top^{(s,K)}, \lambda_\top^{(s,K)})$ to $(G_\top'^{(s',K')}, \lambda_\top'^{(s',K')})$ that maps s to s'. Then*

$$\left\{\chi_{G,k}(s, v) \;\middle|\; v \in K\right\} \cap \left\{\chi_{G',k}(s', v') \;\middle|\; v' \in K'\right\} = \emptyset.$$

Proof: If $\chi_{G,k}(s) \neq \chi_{G',k}(s')$, then the conclusion of the lemma is obvious. Thus, we can assume otherwise.

We have already seen with Corollary 7.3 that cut vertices obtain other colours than non-cut vertices. Thus, we can assume that G and G' are already coloured in a way such that s and s' have a colour different from the colours of vertices in $K \cup K'$. Also, by Theorem 7.2, without loss of generality, we may assume that vertex pairs in a common 2-connected component have different colours than vertex pairs that are not contained in a common 2-connected component.

For readability, we drop the superscripts (s, K) and (s', K') and the subscript k in the remainder of this proof. We show by induction on i that for all $u, v \in K \cup \{s\}$ and all $u', v' \in K' \cup \{s'\}$ with $\{u, v\} \not\subseteq \{s\}$ and $\{u', v'\} \not\subseteq \{s'\}$, the following implication is true:

$$\chi^i_{G_\top}(u,v) \neq \chi^i_{G'_\top}(u',v') \;\Rightarrow\; \chi^i_G(u,v) \neq \chi^i_{G'}(u',v'). \tag{7.3}$$

For $i = 0$, the claim follows from the definition of the colourings $\lambda_\top$ and $\lambda'_\top$. For the induction step from i to $i+1$, assume that there exist vertices $u, v \in K \cup \{s\}$ and $u', v' \in K' \cup \{s'\}$ such that $\{u, v\} \not\subseteq \{s\}$, $\{u', v'\} \not\subseteq \{s'\}$ with $\chi^i_{G_\top}(u,v) = \chi^i_{G'_\top}(u',v')$ and $\chi^{i+1}_{G_\top}(u,v) \neq \chi^{i+1}_{G'_\top}(u',v')$ (otherwise, the statement holds trivially).

We show that we can additionally assume that $\chi^i_{G_\top}(u,u) = \chi^i_{G'_\top}(u',u')$ and $\chi^i_{G_\top}(v,v) = \chi^i_{G'_\top}(v',v')$. Indeed, suppose $\chi^i_{G_\top}(u,u) \neq \chi^i_{G'_\top}(u',u')$ and $\chi^i_{G_\top}(u,v) = \chi^i_{G'_\top}(u',v')$. This assumption implies that

$$\{\!\{(\chi^{i-1}_{G_\top}(x,u), \chi^{i-1}_{G_\top}(u,x)) \mid x \in V(G)\}\!\} \neq \{\!\{(\chi^{i-1}_{G'_\top}(x',u'), \chi^{i-1}_{G_\top}(u',x')) \mid x' \in V(G')\}\!\},$$

which, by converse equivalence, yields

$$\{\!\{\chi^{i-1}_{G_\top}(x,u) \mid x \in V(G)\}\!\} \neq \{\!\{\chi^{i-1}_{G'_\top}(x',u') \mid x' \in V(G')\}\!\}.$$

However, this implies $\chi^i_{G_\top}(u,v) \neq \chi^i_{G'_\top}(u',v')$ again, a contradiction. By symmetry, we can also conclude $\chi^i_{G_\top}(v,v) = \chi^i_{G'_\top}(v',v')$.

Thus, there must be a colour tuple (c_1, c_2) such that the sets

$$M := \big\{x \;\big|\; x \in V(G_\top) \setminus \{u,v\}, \big(\chi^i_{G_\top}(x,v), \chi^i_{G_\top}(u,x)\big) = (c_1,c_2)\big\}$$

and

$$M' := \big\{x' \;\big|\; x' \in V(G'_\top) \setminus \{u',v'\}, \Big(\chi^i_{G'_\top}(x',v'), \chi^i_{G'_\top}(u',x')\Big) = (c_1,c_2)\big\}$$

do not have the same cardinality. Let

$$D := \big\{\, \big(\chi^i_G(x,v), \chi^i_G(u,x)\big) \;\big|\; x \in M\big\} \cup \big\{\, \big(\chi^i_{G'}(x',v'), \chi^i_{G'}(u',x')\big) \mid x' \in M'\big\}.$$

By induction and by Theorem 7.2, we have that

$$\bigl\{x \bigm| x \in V(G) \setminus \{u, v\}, \bigl(\chi^i_G(x, v), \chi^i_G(u, x)\bigr) \in D\bigr\} = M$$

and

$$\bigl\{x' \bigm| x' \in V(G') \setminus \{u', v'\}, \bigl(\chi^i_{G'}(x', v'), \chi^i_{G'}(u', x')\bigr) \in D\bigr\} = M',$$

and hence, these sets do not have the same cardinality. Thus, $\chi^{i+1}_G(u, v) \neq \chi^{i+1}_{G'}(u', v')$.

Having shown Implication (7.3), it suffices to show that

$$\bigl\{\chi_{G_\top}(s, v) \bigm| v \in K\bigr\} \cap \bigl\{\chi_{G'_\top}(s', v') \bigm| v' \in K'\bigr\} = \emptyset.$$

However, this follows directly from the assumption that the k-dimensional Weisfeiler-Leman algorithm distinguishes every pair of non-isomorphic vertex-coloured 2-connected graphs in $\mathcal{G}$ and that the graphs $(G^{(s,K)}_\top, \lambda^{(s,K)}_\top)$ and $(G'^{(s',K')}_\top, \lambda'^{(s',K')}_\top)$ are 2-connected. □

With this, we can prove the following.

Lemma 7.29. *Assume $k \geq 2$ and suppose the k-dimensional Weisfeiler-Leman algorithm distinguishes all non-isomorphic vertex-coloured 2-connected graphs in $\mathcal{G}$. For two graphs $G, G' \in \mathcal{G}$ with vertex colourings λ, λ', respectively, suppose $s \in V(G)$ and $s' \in V(G')$. Assume that there is no isomorphism from $(G^s_\top, \lambda^s_\top)$ to $(G'^{s'}_\top, \lambda'^{s'}_\top)$ that maps s to s'. Then $\chi_{G,k}(s) \neq \chi_{G',k}(s')$.*

Proof: Towards a contradiction, assume that $\chi_{G,k}(s) = \chi_{G',k}(s')$. Further let

$$\bigl\{K_1, \ldots, K_t\bigr\} := \bigl\{K \bigm| (s, K) \in P_0(G)\bigr\}$$

and that

$$\bigl\{K'_1, \ldots, K'_{t'}\bigr\} := \bigl\{K' \bigm| (s', K') \in P_0(G)\bigr\}.$$

From $(G^s_\top, \lambda^s_\top)_{(s)} \not\cong (G'^{s'}_\top, \lambda'^{s'}_\top)_{(s')}$, we conclude that there is a vertex-coloured graph (H, λ_H) such that the sets

$$I := \Bigl\{i \Bigm| \Bigl(G^{(s,K_i)}_\top, \lambda^{(s,K_i)}_\top\Bigr) \cong (H, \lambda_H)\Bigr\}$$

and

$$I' := \Biggl\{j \Biggm| \Biggl(G'^{(s',K'_j)}_\top, \lambda'^{(s',K'_j)}_\top\Biggr) \cong (H, \lambda_H)\Biggr\}$$

have different cardinalities. Note that all K_i with $i \in I$ and all K'_j with $j \in I'$ have the same cardinality. By Lemma 7.28, for $v \in K_j$ with $j \in I$ and for $v' \in K_{j'}$ with $j' \notin I'$, we have $\chi_{G,k}(s, v) \neq \chi_{G',k}(s', v')$. Letting $C := \bigl\{\chi_{G,k}(s, v) \bigm| \exists i \in I \colon v \in K_i\bigr\}$, the vertices s and s' do not have the same number of neighbours connected via an arc of colour C. We conclude that $\chi_{G,k}(s) \neq \chi_{G',k}(s')$. □

Corollary 7.30. *Assume $k \in \mathbb{N}_{\geq 2}$ and suppose the k-dimensional Weisfeiler-Leman algorithm distinguishes all non-isomorphic vertex-coloured 2-connected graphs in $\mathcal{G}$. Let (G, λ) and (G', λ') be vertex-coloured graphs with $G, G' \in \mathcal{G}$ that are connected but not 2-connected. If for $v_1, v_2 \in V(G_\perp)$ and $v_1', v_2' \in V(G_\perp')$, it holds that $\chi_{G_\perp,k}(v_1, v_2) \neq \chi_{G_\perp',k}(v_1', v_2')$, then $\chi_{G,k}(v_1, v_2) \neq \chi_{G',k}(v_1', v_2')$.*

Proof: By Lemma 7.27, with respect to the colourings $\chi_{G,k}$ and $\chi_{G',k}$, the vertices in $V(G_\perp)$ and $V(G_\perp')$ have other colours than the vertices in $V(G) \setminus V(G_\perp)$ and $V(G') \setminus V(G_\perp')$. Thus, it suffices to show that the colourings $\chi_{G,k}$ and $\chi_{G',k}$ refine the colourings $\lambda_\perp$ and $\lambda_\perp'$, respectively, on the domains of those. For this, we first observe that by Corollary 7.4, we can assume that either both $\{v_1, v_2\}$ and $\{v_1', v_2'\}$ are 2-separators or neither of them is. Note that $\lambda_\perp$ and $\lambda_\perp'$ are only defined on tuples (u, v) for which either $u = v$ or $\{u, v\}$ forms a 2-separator or an edge. If $\{v_1, v_2\}$ and $\{v_1', v_2'\}$ are 2-separators, they are not separators of minimal size. Thus, $G_\top^{\{v_1,v_2\}}$ and $G_\top'^{\{v_1',v_2'\}}$ have no other vertices than v_1 and v_2, and v_1' and v_2', respectively. Hence, from the definition of $\lambda_\perp$ and $\lambda_\perp'$, we obtain $\lambda_\perp(u, v) = \lambda_\perp'(u', v')$ or $\lambda(u, v) = \lambda'(u', v')$. In both cases, we are done.

Therefore, again by the definition of $\lambda_\perp$ and $\lambda_\perp'$, we only have to consider the following two cases.

1. If $v_1 = v_2$, $v_1' = v_2'$, and it holds that $\text{ISOTYPE}((G_\top^{\{v_1\}}, \lambda_\top^{\{v_1\}})_{(v_1)}) \neq \text{ISOTYPE}((G_\top'^{\{v_1'\}}, \lambda_\top'^{\{v_1'\}})_{(v_1')})$, then $\chi_{G,k}(v_1) \neq \chi_{G',k}(v_1')$.
2. If $\{v_1, v_2\} \in E(G)$, $\{v_1', v_2'\} \in E(G')$, and $\lambda_\perp(v_1, v_2) \neq \lambda_\perp'(v_1', v_2')$, then $\chi_{G,k}(v_1, v_2) \neq \chi_{G',k}(v_1', v_2')$.

To show the first part, from the assumption that

$$\text{ISOTYPE}\left((G_\top^{\{v_1\}}, \lambda_\top^{\{v_1\}})_{(v_1)}\right) \neq \text{ISOTYPE}\left((G_\top'^{\{v_1'\}}, \lambda_\top'^{\{v_1'\}})_{(v_1')}\right),$$

we know that v_1 and v_1' must be cut vertices. Thus, the statement is exactly Lemma 7.29. For the second part, we obtain $\lambda(v_1, v_2) \neq \lambda'(v_1', v_2')$, which implies $\chi_{G,k}(v_1, v_2) \neq \chi_{G',k}(v_1', v_2')$. □

We can now show the reduction from arbitrary to vertex-coloured 2-connected graphs.

Proof of Theorem 7.26: Let $G, G' \in \mathcal{G}$ be non-isomorphic graphs with vertex colourings λ and λ', respectively. It suffices to show the statement in the case that G and G' are connected. We proceed by induction on $|G| + |G'|$. The base case $|G| + |G'| = 2$ is trivial. For the induction step, if both graphs are 2-connected, then the statement follows directly from the assumptions. If exactly one of the graphs is 2-connected, then exactly one of the graphs has a cut vertex and the statement follows from Lemma 7.3. Thus, suppose both graphs are not 2-connected, but connected. Since $(G, \lambda) \ncong (G', \lambda')$, we know by Lemma 7.25 that $(G_\perp, \lambda_\perp) \ncong (G_\perp', \lambda_\perp')$. By induction, the k-dimensional Weisfeiler-Leman algorithm distinguishes $(G_\perp, \lambda_\perp)$ from $(G'_\perp, \lambda'_\perp)$. Furthermore, by Lemma 7.27, the vertices in $V(G_\perp)$ and $V(G'_\perp)$ have other colours than the vertices in $V(G) \setminus V(G_\perp)$ and $V(G') \setminus V(G'_\perp)$. Moreover, by Corollary 7.30, the partition of the vertices and arcs induced by the colouring $\chi_{G,k}$ restricted to vertex tuples with elements from $V(G_\perp)$ is finer than the partition induced by $\chi_{G_\perp,k}$ on the graph $G_\perp$ with initial colouring $\lambda_\perp$. Similarly, the partition induced by $\chi_{G',k}$ restricted to vertex tuples with elements from $V(G'_\perp)$ is finer than the partition induced by $\chi_{G'_\perp,k}$ on $G'_\perp$ with initial colouring $\lambda'_\perp$. Thus, the k-dimensional Weisfeiler-Leman algorithm distinguishes (G, λ) from (G', λ'). □

Therefore, we can assume that the graphs we intend to distinguish with the Weisfeiler-Leman algorithm are 2-connected. In the remainder of this chapter, we show that it actually suffices to be able to distinguish edge-coloured 3-connected graphs.

7.4 Reduction to Edge-Coloured 3-Connected Graphs

In this section, our aim is to relax the assumption from Theorem 7.26, which requires that 2-connected graphs are distinguished, to an assumption of 3-connected graphs being distinguished.

The strategy to prove our reduction follows similar ideas as those used in Section 7.3. It relies on the assumption that the input consists of vertex-coloured 2-connected graphs, which we can make without loss of generality by the reduction from Theorem 7.26. Now we consider the decomposition of vertex-coloured 2-connected graphs into their so-called “3-connected components”.

However, a 3-connected component of a 2-connected graph G is a minor of G and not necessarily a subgraph. Thus, we require that the graph class $\mathcal{G}$ is minor-closed. Furthermore, to enable the inductive approach, we will now have to consider graphs G in which the 2-tuples (u, v) with $\{u, v\} \in E(G)$, i.e. the arcs, are also coloured. However, it turns out that it is not sufficient to require that edge-coloured graphs are distinguished. In fact, we need the stronger ability of the algorithm to determine orbits (see Definition 3.15).

Just as we did in the previous section, we want to apply a recursive strategy that relies on Lemma 7.25. However, to apply that lemma we require a minimum degree of 3. The following lemma states that vertices of degree 2 can be removed.

Lemma 7.31. *Let $\mathcal{G}$ be a minor-closed graph class and assume $k \in \mathbb{N}_{\geq 2}$. Suppose the k-dimensional Weisfeiler-Leman algorithm distinguishes all edge-coloured graphs in $\mathcal{G}$ of minimum degree at least 3. Then the k-dimensional Weisfeiler-Leman algorithm distinguishes all non-isomorphic edge-coloured graphs in $\mathcal{G}$.*

Sketch of the proof: The proof is a basic exercise regarding the Weisfeiler-Leman algorithm. Let (G, λ) be an edge-coloured graph of minimum degree less than 3 with $G \in \mathcal{G}$ and let (G', λ') be a second edge-coloured graph with $G \in \mathcal{G}$ that is not isomorphic to G.

As the *base case*, we first observe that the 2-dimensional Weisfeiler-Leman algorithm decides isomorphism for graphs whose degree sequences differ, as well as for graphs of maximum degree at most 2 (i.e. disjoint unions of cycles, isolated edges and isolated vertices). In particular, we can assume without loss of generality that G' has the same degree sequence as G.

Thus, assume both G and G' contain vertices of degree at least 3. For two vertices u and v, call a path from u to v a *chain* if all its inner vertices have degree 2 in the surrounding graph. Note that, since vertices of different degrees obtain different colours, the colour $\chi_{G,k}(u, v)$ or $\chi_{G',k}(u, v)$ implicitly encodes, for each i, the number of chains of length exactly i from u to v.

We describe an algorithmic reduction to the base case.

Step 1: Repeat the following procedure recursively on its own output while the resulting graphs have the same degree sequences and contain vertices of degree 1 and vertices of degree at least 3. Assign to each vertex $v \in V(G)$ a colour encoding the tuple containing, for each pair C, C' of a vertex colour C and an edge colour C', the number

$$\left|\{w \mid w \in N(v),\, w \text{ has degree 1 and colour } C, \text{ and } (v, w) \text{ has colour } C'\}\right|$$

and remove all vertices of degree 1 from G. Construct a coloured subgraph of G' in an analogous fashion.

Step 2: Let (H, λ_H) and $(H', \lambda_{H'})$ be the output of Step 1. If H and H' have different degree sequences, output NO. If they have maximum degree at most 2, apply the base case to determine the output. If both graphs have minimum degree 2 and contain vertices of degree at least 3, define (J, λ_J) and $(J', \lambda_{J'})$ as the minors of H and H' obtained by only retaining vertices of degree at least 3, connecting two such vertices with an edge if there is a chain between them, and assigning to the edge a colour that encodes the multiset of lengths of (coloured) chains between the vertices. Call Step 1 on input (J, λ_J) and $(J', \lambda_{J'})$.

Clearly, the algorithm terminates and outputs NO if and only if the final computed graphs are non-isomorphic. Now an induction over the number of recursive calls shows that the k-dimensional Weisfeiler-Leman algorithm distinguishes (G, λ) and (G', λ') if and only if the above algorithm outputs NO. □

The lemma allows us to focus on graphs G with minimum degree 3. Doing so, similarly to Theorem 7.2, the following lemma says that arcs between connected components in $P_0(G)$ obtain different colours than arcs within connected components in $P_0(G)$.

Lemma 7.32. *Assume $k \in \mathbb{N}_{\geq 2}$ and let G and G' be 2-connected graphs of minimum degree at least 3 that are not* 3*-connected. Suppose $(S, K) \in P_0(G)$ and $(S', K') \in P_0(G')$ and that $u, v \in K$ and $u' \in K', v' \notin K'$. Then $\chi_{G,k}(u, v) \neq \chi_{G',k}(u', v')$.*

Proof: By Corollary 7.8, we can assume that vertices contained in a 2-separator have different χ_k-colours than vertices which are not contained in any 2-separator. Thus, vertex pairs connected via a path which does not contain any vertices of 2-separators obtain different colours than vertex pairs for which every connecting path contains a vertex of a 2-separator. Since K is a connected component of $G - S$, there is a path in K from u to v. Moreover, by Part 3 of Lemma 7.24, no vertex on the path is contained in a 2-separator of G. On the other hand, in G', all paths from u' to v' must include a vertex of S', since $v' \notin K'$. We conclude that $\chi_{G,k}(u, v) \neq \chi_{G',k}(u', v')$. □

Lemma 7.33. *Assume $k \in \mathbb{N}_{\geq 2}$ and let G and G' be 2-connected graphs of minimum degree at least 3 that are not* 3*-connected. Then for all $v \in V(G_\perp)$ and $v' \in V(G') \setminus V(G'_\perp)$, it holds that $\chi_{G,k}(v) \neq \chi_{G',k}(v')$.*

Proof: Let G, G', v, v' be as described in the prerequisites of the lemma. By Part 3 of Lemma 7.24, the vertex v' is not contained in any 2-separator of G'. If v is contained in a 2-separator of G, by Corollary 7.8, the k-dimensional Weisfeiler-Leman algorithm distinguishes v from v'. Hence, we can assume otherwise.

We show that there are exactly two vertices that each are contained in a 2-separator and can be reached from v' via a path that does not use any other vertices contained in 2-separators, while this is not the case for v.

Since $v' \in V(G') \setminus V(G'_\perp)$, there must be a 2-separator S' and a connected component K' of $G - S'$ with $(S', K') \in P_0(G')$ and $v \in K'$. Furthermore, since G' is connected and S' is a separator of minimal size, there must be a path in $G'[S' \cup K']$ from v' to each of the two vertices contained in S'. Each path can be chosen to avoid the other vertex in S', otherwise there would be a cut vertex in G', contradicting its 2-connectedness. This means that for the two chosen paths, all but the last vertex are contained in K'. In particular, by Part 3 of Lemma 7.24, none of them is contained in a 2-separator. Note that all vertices reachable from v' in $G' - S'$ lie in K' and thus, there are exactly two vertices that each are contained in a 2-separator and can be reached from v' via a path that does not use any other vertices contained in 2-separators.

We now show that there are at least three vertices that are contained in a 2-separator and can be reached from v via a path that does not use any other vertices contained in 2-separators. Choose $(S, K) \in P(G)$ such that $v \in K$. Let $\widetilde{s}_1$ be one of the vertices in S. Then there is a path from v to $\widetilde{s}_1$. Let s_1 be the first vertex on the path that is contained in a 2-separator. Its existence is guaranteed because $\widetilde{s}_1$ is a candidate.

Choose $\widetilde{s}_2$ such that $\{s_1, \widetilde{s}_2\}$ forms a 2-separator of G. Since s_1 is not a cut vertex, there must be a path from v to $\widetilde{s}_2$ that avoids s_1. Again, let s_2 be the first vertex on the path that is contained in a 2-separator. Now, two cases can occur. In the first case, $\{s_1, s_2\}$ is not a 2-separator. Then there is a vertex $\widetilde{s}_3$ such that $\{s_1, \widetilde{s}_3\}$ is a 2-separator and such that there is a path from v to $\widetilde{s}_3$ in $G - \{s_1, s_2\}$. Choose s_3 as the first vertex on the path that is contained in a 2-separator.

In the second case, $\{s_1, s_2\}$ is a 2-separator. Let $\widetilde{K}$ be the connected component of $G - \{s_1, s_2\}$ that contains v. Then $(\{s_1, s_2\}, \widetilde{K}) \notin P_0(G)$, since $v \in V(G_\perp)$. Due to Part 3 of Lemma 7.24, there is a 2-separator $\widetilde{S}$ such that $\widetilde{S} \cap \widetilde{K} \neq \emptyset$. Choose $\widetilde{s}_3 \in \widetilde{S} \cap \widetilde{K}$. Since $\widetilde{K}$ is connected, there is a path in $\widetilde{K}$ from v to $\widetilde{s}_3$. Let s_3 be the first vertex on the path contained in a 2-separator.

In both cases, we have found three vertices $\{s_1, s_2, s_3\}$ that each are contained in a 2-separator and such that every s_i is reachable from v via a path that avoids all vertices contained in 2-separators except for s_i.

This property distinguishes v and v' and is detectable by the k-dimensional Weisfeiler-Leman algorithm. Thus, $\chi_{G,k}(v) \neq \chi_{G',k}(v')$. □

Lemma 7.34. *For $k \in \mathbb{N}_{\geq 2}$, suppose the k-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected graphs in $\mathcal{G}$. Suppose (G, λ) and (G', λ') with $G, G' \in \mathcal{G}$ are edge-coloured 2-connected graph of minimum degree at least 3 that are not 3-connected. Assume that $(\{s_1, s_2\}, K) \in P_0(G)$ and $(\{s'_1, s'_2\}, K') \in P_0(G')$.*

Suppose that no isomorphism from the graph $\big(G_\top^{(\{s_1,s_2\},K)}, \lambda_\top^{(\{s_1,s_2\},K)}\big)$ to the graph $\big(G_\top^{\prime(\{s'_1,s'_2\},K')}, \lambda_\top^{\prime(\{s'_1,s'_2\},K')}\big)$ maps s_1 to s'_1. Then

$$\big\{\chi_{G,k}(s_1, v) \bigm| v \in K\big\} \cap \big\{\chi_{G',k}(s'_1, v) \bigm| v \in K'\big\} = \emptyset.$$

Proof: This proof is an adaption of the proof of Lemma 7.28. For readability and since the dimension of the algorithm is fixed in this proof, we drop the subscript k, i.e. we write χ_G instead of $\chi_{G,k}$.

If $\chi_G(s_1) \neq \chi_{G'}(s'_1)$, then the conclusion of the lemma is obvious. Thus, we can assume otherwise. We have already seen with Corollary 7.8 that 2-separators obtain other colours than other pairs of vertices. Thus, with Part 3 of Lemma 7.24, we can assume that G and G' are already coloured in a way such that the pairs (s_1, s_2), (s_2, s_1), (s'_1, s'_2), (s'_2, s'_1) have colours different from the colours of pairs of vertices (t_1, t_2) with $\{t_1, t_2\} \cap (K \cup K') \neq \emptyset$. Also, we may presuppose without loss of generality that s_1, s_2, s'_1, s'_2 have colours different from colours of vertices that are not contained in any 2-separator of any of the graphs.

Moreover, by Lemma 7.32, we may assume that pairs of vertices that are both contained in K have other colours than pairs of vertices for which just one vertex is contained in K (and similarly for K' in G').

For readability, in the rest of this proof, we drop the superscripts $(\{s_1, s_2\}, K)$ and $(\{s'_1, s'_2\}, K')$. That is, whenever we write $G_\top$, we mean $G_\top^{(\{s_1,s_2\},K)}$, and whenever we write $G'_\top$, we actually mean $G_\top'^{(\{s'_1,s'_2\},K')}$. Similarly, $\lambda_\top$ and $\lambda'_\top$ stand for their versions with the superscripts $(\{s_1, s_2\}, K)$ and $(\{s'_1, s'_2\}, K')$.

Claim 1. For all $i \in \mathbb{N}_0$, all $u, v \in K \cup \{s_1, s_2\}$ and all $u', v' \in K' \cup \{s'_1, s'_2\}$ with $\{u, v\} \not\subseteq \{s_1, s_2\}$ and $\{u', v'\} \not\subseteq \{s'_1, s'_2\}$, the following implication holds:

$$\chi^i_{G_\top}(u, v) \neq \chi^i_{G'_\top}(u', v') \;\Rightarrow\; \chi^i_G(u, v) \neq \chi^i_{G'}(u', v'). \tag{7.4}$$

Proof of the claim: We proceed via induction on i. For the induction base $i = 0$, suppose $\chi^0_{G_\top}(u, v) \neq \chi^0_{G'_\top}(u', v')$. Then either there is no isomorphism from the graph $G_\top[\{u, v\}]$ to $G'_\top[\{u', v'\}]$ mapping u to u' and v to v', or $\lambda_\top(u, v) \neq \lambda'_\top(u', v')$.

In the first case, by definition of the graphs $G_\top$ and $G'_\top$ and their colourings, we immediately obtain $\chi^0_G(u, v) \neq \chi^0_{G'}(u', v')$, since an isomorphism from $G[\{u, v\}]$ to $G'[\{u', v'\}]$ that maps u to u' and v to v' would induce an isomorphism from the graph $G_\top[\{u, v\}]$ to $G'_\top[\{u', v'\}]$ with the same mappings. Thus, consider the second case, i.e. that $\lambda_\top(u, v) \neq \lambda'_\top(u', v')$. Since $\{u, v\} \not\subseteq \{s_1, s_2\}$ and $\{u', v'\} \not\subseteq \{s'_1, s'_2\}$, it holds that $(\lambda(u, v), 2) = \lambda_\top(u, v) \neq \lambda_\top(u', v') = (\lambda'(u', v'), 2)$. Thus, $\chi^0_G(u, v) = \lambda(u, v) \neq \lambda'(u', v') = \chi^0_{G'}(u', v')$.

For the induction step from i to $i+1$, assume there exist vertices $u, v \in K \cup \{s_1, s_2\}$ and $u', v' \in K' \cup \{s'_1, s'_2\}$ which satisfy $\{u, v\} \not\subseteq \{s_1, s_2\}$ and $\{u', v'\} \not\subseteq \{s'_1, s'_2\}$, and furthermore $\chi^i_{G_\top}(u, v) = \chi^i_{G'_\top}(u', v')$ and $\chi^{i+1}_{G_\top}(u, v) \neq \chi^{i+1}_{G'_\top}(u', v')$.

Just like in the proof of Lemma 7.28, we can additionally assume $\chi^i_{G_\top}(u, u) = \chi^i_{G'_\top}(u', u')$ and $\chi^i_{G_\top}(v, v) = \chi^i_{G'_\top}(v', v')$. Thus, there must be a colour tuple (c_1, c_2) such that the sets

$$\begin{aligned} M \coloneqq \big\{x \;\big|\; & x \in V(G_\top) \setminus \{u, v\}, \\ & \big(\chi^i_{G_\top}(x, v), \chi^i_{G_\top}(u, x)\big) = (c_1, c_2)\big\} \end{aligned}$$

and

$$\begin{aligned} M' \coloneqq \big\{x' \;\big|\; & x' \in V(G'_\top) \setminus \{u', v'\}, \\ & \big(\chi^i_{G'_\top}(x', v'), \chi^i_{G'_\top}(u', x')\big) = (c_1, c_2)\big\} \end{aligned}$$

do not have the same cardinality. Let

$$\begin{aligned} D \coloneqq & \Big\{\big(\chi^i_G(x, v), \chi^i_G(u, x)\big) \;\Big|\; x \in M\Big\} \cup \\ & \Big\{\big(\chi^i_{G'}(x', v'), \chi^i_{G'}(u', x')\big) \;\big|\; x' \in M'\Big\}. \end{aligned}$$

We show that

$$\Big\{x \mid x \in V(G) \setminus \{u,v\}, \big(\chi_G^i(x,v), \chi_G^i(u,x)\big) \in D\Big\} = M$$

The inclusion "$\supseteq$" is clear by the definitions of M and D. For the inclusion "$\subseteq$", let $x \in V(G) \setminus \{u,v\}$ be a vertex with $\big(\chi_G^i(x,v), \chi_G^i(u,x)\big) \in D$. Then by the definition of D, there must be a vertex $x' \in M$ with $\big(\chi_G^i(x',v), \chi_G^i(u,x')\big) = \big(\chi_G^i(x,v), \chi_G^i(u,x)\big)$. Thus, $\Big(\chi_{G_\top}^i(x',v), \chi_{G_\top}^i(u,x')\Big) = (c_1, c_2)$ by the definition of M. Now using the induction assumption for i stated in (7.4), we obtain $\chi_{G_\top}^i(x,v) = \chi_{G_\top}^i(x',v) = c_1$ and $\chi_{G_\top}^i(u,x) = \chi_{G_\top}^i(u,x') = c_2$. This implies $x \in M$.

Similarly, we have

$$\Big\{x' \mid x' \in V(G') \setminus \{u',v'\}, \big(\chi_{G'}^i(x',v'), \chi_{G'}^i(u',x')\big) \in D\Big\} = M'.$$

Hence, these sets do not have equal cardinalities. Thus, $\chi_G^{i+1}(u,v) \neq \chi_{G'}^{i+1}(u',v')$. ⌟

Having shown the claim, it suffices to show that

$$\big\{\chi_{G_\top}(s_1,v) \bigm| v \in K\big\} \cap \big\{\chi_{G'_\top}(s_1',v') \bigm| v' \in K'\big\} = \emptyset.$$

For this, it suffices to prove that $\chi_{G_\top}(s_1) \neq \chi_{G'_\top}(s_1')$ holds.

The graphs $(G_\top^{(\{s_1,s_2\},K)}, \lambda_\top^{(\{s_1,s_2\},K)})$ and $(G_\top^{\prime(\{s_1',s_2'\},K')}, \lambda_\top^{\prime(\{s_1',s_2'\},K')})$ are 3-connected. Thus, since we have assumed that the k-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected graphs in $\mathcal{G}$, if it held that $\chi_{G_\top}(s_1) = \chi_{G'_\top}(s_1')$, there would have to be an isomorphism from the graph $(G_\top^{(\{s_1,s_2\},K)}, \lambda_\top^{(\{s_1,s_2\},K)})$ to $(G_\top^{\prime(\{s_1',s_2'\}',K')}, \lambda_\top^{\prime(\{s_1',s_2'\},K')})$ that maps s_1 to s_1', a contradiction to our assumption. □

Using the lemma, we can show the following.

Lemma 7.35. *Assume $k \in \mathbb{N}_{\geq 2}$ and suppose the k-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected graphs in $\mathcal{G}$. Suppose (G,λ) and (G',λ') with $G, G' \in \mathcal{G}$ are edge-coloured 2-connected graphs of minimum degree at least 3 that are not 3-connected and let $\{s_1,s_2\} \subseteq V(G)$ and $\{s_1',s_2'\} \subseteq V(G')$ be 2-separators of G and G', respectively.*

Suppose that no isomorphism from $\big(G_\top^{\{s_1,s_2\}}, \lambda_\top^{\{s_1,s_2\}}\big)$ to $\big(G_\top^{\prime\{s_1',s_2'\}}, \lambda_\top^{\prime\{s_1',s_2'\}}\big)$ maps s_1 to s_1'. Then $\chi_{G,k}(s_1,s_2) \neq \chi_{G',k}(s_1',s_2')$.

Proof: This proof is similar to the proof of Lemma 7.29. Suppose that

$$\{K_1, \dots, K_t\} = \big\{K \bigm| (\{s_1,s_2\},K) \in P_0(G)\big\}$$

and that

$$\{K_1', \dots, K_{t'}'\} = \big\{K' \bigm| (\{s_1',s_2'\},K') \in P_0(G')\big\}.$$

Since there is no isomorphism from $\big(G_{\top}^{\{s_1,s_2\}}, \lambda_{\top}^{\{s_1,s_2\}}\big)$ to $\big(G'^{\{s_1',s_2'\}}_{\top}, \lambda'^{\{s_1',s_2'\}}_{\top}\big)$ that maps s_1 to s_1', there is an edge-coloured graph (H, λ_H) such that the sets

$$I := \left\{ i \;\middle|\; \left(G_{\top}^{(\{s_1,s_2\},K_i)}, \lambda_{\top}^{(\{s_1,s_2\},K_i)}\right)_{(s_1)} \cong (H, \lambda_H) \right\}$$

and

$$I' := \left\{ j \;\middle|\; \left(G'^{(\{s_1',s_2'\},K_j')}_{\top}, \lambda'^{(\{s_1',s_2'\},K_j')}_{\top}\right)_{(s_1')} \cong (H, \lambda_H) \right\}$$

have different cardinalities. Note that all K_i with $i \in I$ and all K_j' with $j \in I'$ have the same cardinality. We know by Lemma 7.34 that for $v \in K_i$ with $i \in I$ and $v' \in K_j'$ with $j \notin I'$, we have $\chi_{G,k}(s_1, v) \neq \chi_{G',k}(s_1', v')$. Furthermore, using Lemma 7.33, for $v \in V\big(G_{\top}^{\{s_1,s_2\}}\big) \setminus \{s_1, s_2\}$ and $v' \notin V\big(G'^{\{s_1',s_2'\}}_{\top}\big) \setminus \{s_1', s_2'\}$, it holds that $\big(\chi_{G,k}(s_1, v), \chi_{G,k}(v, s_2)\big) \neq \big(\chi_{G',k}(s_1', v'), \chi_{G',k}(v', s_2')\big)$. Indeed, v is a vertex in $V(G) \setminus V(G_\perp)$ from which, for all $i \in \{1, 2\}$, there is a path to s_i that does not contain any vertex in a 2-separator apart from s_i, while the analogous statement does not hold for v'.

Letting

$$C := \big\{ \big(\chi_{G,k}(s_1, v), \chi_{G,k}(v, s_2)\big) \;\big|\; \exists i \in I \text{ s.t. } v \in K_i \big\},$$

the sets

$$\big\{ v \in V(G) \mid \big(\chi_{G,k}(s_1, v), \chi_{G,k}(v, s_2)\big) \in C \big\}$$

and

$$\big\{ v' \in V(G') \mid \big(\chi_{G',k}(s_1', v'), \chi_{G',k}(v', s_2')\big) \in C \big\}$$

do not have the same cardinality. We conclude that $\chi_{G,k}(s_1, s_2) \neq \chi_{G',k}(s_1', s_2')$. □

We can now deduce that the colouring computed by the 2-dimensional Weisfeiler-Leman algorithm is at least as fine as the $\perp$-colouring.

Corollary 7.36. *Assume $k \in \mathbb{N}_{\geq 2}$ and suppose that the k-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected graphs in $\mathcal{G}$. Let $G, G' \in \mathcal{G}$ be 2-connected graphs of minimum degree at least 3 that are not 3-connected and that have edge colourings λ and λ', respectively. If, for vertices $v_1, v_2 \in V(G_\perp)$ and $v_1', v_2' \in V(G_\perp')$, it holds that $\chi_{G_\perp,k}(v_1, v_2) \neq \chi_{G_\perp',k}(v_1', v_2')$, then $\chi_{G,k}(v_1, v_2) \neq \chi_{G',k}(v_1', v_2')$.*

Proof: Without loss of generality, we may assume that $\lambda(v_1, v_1) = \lambda'(v_1', v_1')$ and $\lambda(v_2, v_2) = \lambda'(v_2', v_2')$. Furthermore, by Lemma 7.33, with respect to the colourings $\chi_{G,k}$ and $\chi_{G',k}$, the vertices in $V(G_\perp)$ and $V(G_\perp')$ have other colours than the vertices in $V(G) \setminus V(G_\perp)$ and $V(G') \setminus V(G_\perp')$. Thus, it suffices to show that the colourings $\chi_{G,k}$ and $\chi_{G',k}$ refine the colourings $\lambda_\perp$ and $\lambda_\perp'$, respectively, on the domains of those. Thus, by the definition of $\lambda_\perp$ and $\lambda_\perp'$ and since these maps are only defined on tuples (u, v) for which either $u = v$ or $\{u, v\}$ forms a 2-separator or an edge, it suffices to show the following two statements.

1. If $\{v_1, v_2\}$, $\{v'_1, v'_2\}$ are 2-separators and $\text{ISOTYPE}\left((G_{\top}^{\{v_1,v_2\}}, \lambda_{\top}^{\{v_1,v_2\}})_{(v_1)}\right) \neq \text{ISOTYPE}\left((G'^{\{v'_1,v'_2\}}_{\top}, \lambda'^{\{v'_1,v'_2\}}_{\top})_{(v'_1)}\right)$, then $\chi_{G,k}(v_1, v_2) \neq \chi_{G',k}(v'_1, v'_2)$.
2. If $v_1 = v_2$ and $v'_1 = v'_2$, or $\{v_1, v_2\} \in E(G)$ and $\{v'_1, v'_2\} \in E(G')$, but the pairs $\{v_1, v_2\}$ and $\{v'_1, v'_2\}$ are not 2-separators, then, if $\lambda_{\perp}(v_1, v_2) \neq \lambda'_{\perp}(v'_1, v'_2)$, it holds that $\chi_{G,k}(v_1, v_2) \neq \chi_{G',k}(v'_1, v'_2)$.

The first part is exactly Lemma 7.35. For the second part, from the definition of $\lambda_{\perp}$ and $\lambda'_{\perp}$, we obtain $\lambda(v_1, v_2) \neq \lambda'(v'_1, v'_2)$, which implies $\chi_{G,k}(v_1, v_2) \neq \chi_{G',k}(v'_1, v'_2)$. □

We are ready to prove our reduction to 3-connected graphs.

Theorem 7.37. *Let $\mathcal{G}$ be a minor-closed graph class and assume $k \in \mathbb{N}_{\geq 2}$. Suppose the k-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected graphs in $\mathcal{G}$. Then the k-dimensional Weisfeiler-Leman algorithm distinguishes all non-isomorphic edge-coloured graphs in $\mathcal{G}$.*

Proof: Let $G, G' \in \mathcal{G}$ be non-isomorphic graphs with edge colourings λ and λ', respectively, such that $(G, \lambda) \ncong (G', \lambda')$ and suppose the k-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected graphs in $\mathcal{G}$. We prove the statement by induction on $|G| + |G'|$. The base case $|G| + |G'| = 2$ is trivial. For the induction step, if both graphs are 3-connected, then the statement follows directly from the assumptions since "determines orbits" is a stronger assumption than "distinguishes" (see Observation 3.16). If exactly one of the graphs is 3-connected, then exactly one of the graphs has a 2-separator and the statement follows from Corollary 7.8.

Thus, suppose that both graphs are not 3-connected and assume all pairs of edge-coloured graphs (H, λ_H) and $(H', \lambda_{H'})$ with $|V(H)| + |V(H')| < |G| + |V(G')|$ are distinguished. By Theorem 7.26, (with the graph class $\mathcal{G}$ from the theorem being the class of graphs containing every graph isomorphic to G, G', or a minor of one of them,) it suffices to show the statement for the case that G and G' are 2-connected. If G and G' do not have the same minimum degree, they are distinguished by their degree sequences.

Now first suppose both G and G' have minimum degree at least 3. By Corollary 7.36, the partition of the vertices and arcs induced by the colouring $\chi_{G,k}$ restricted to $V(G_{\perp})$ is finer than the partition induced by $\lambda_{\perp}$. Also, by Lemma 7.33, vertices in $V(G_{\perp})$ obtain different colours than vertices in $V(G) \setminus V(G_{\perp})$. Similar statements hold in G'. Thus, the k-dimensional Weisfeiler-Leman algorithm implicitly computes $\lambda_{\perp}$ and $\lambda_{\perp'}$.

Since (G, λ) and (G', λ') are not isomorphic, we know by Lemma 7.25 that also $(G_{\perp}, \lambda_{\perp}) \ncong (G'_{\perp}, \lambda'_{\perp})$. Thus, $(G_{\perp}, \lambda_{\perp})$ and $(G'_{\perp}, \lambda'_{\perp})$ are distinguished by the k-dimensional Weisfeiler-Leman algorithm by induction assumption. Hence, since it computes finer colourings on (G, λ) and (G', λ') than on $(G_{\perp}, \lambda_{\perp})$ and on $(G'_{\perp}, \lambda'_{\perp})$, respectively, the k-dimensional Weisfeiler-Leman algorithm also distinguishes (G,λ) from (G',λ').

By Lemma 7.31, the case that G and G' do not have minimum degree at least 3 reduces to the case of minimum degree at least 3, letting $\mathcal{G}$ be the class of graphs containing every graph isomorphic to G, G' or a minor of one of them. □

In the last two sections, we have concerned ourselves with distinguishing two graphs that are contained in the considered class rather than one graph being identified, i.e. distinguished from every second, non-isomorphic graph that is not necessarily from the considered class. However, our results also have corresponding versions concerning the latter notion.

Lemma 7.38. *Let $\mathcal{G}$ be a minor-closed graph class. The Weisfeiler-Leman dimension of graphs in $\mathcal{G}$ is at most $\max\{2,k\}$, where k is the minimal number ℓ such that the ℓ-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected graphs in $\mathcal{G}$ and identifies such graphs.*

The proof follows almost verbatim the lines of the entire proof of Theorem 7.37, replacing “distinguishes” with “identifies”.

7.5 Discussion

We have proved in this chapter that for $k \geq 2$, the k-dimensional Weisfeiler-Leman algorithm detects k-separators in graphs. As the example in Figure 7.6 shows, dimension $k = 1$ does not suffice to distinguish vertices contained in 2-separators from others. Thus, our upper bound of 2 is tight. Using this result, we showed that the k-dimensional Weisfeiler-Leman algorithm implicitly computes the decomposition of the input graph into its 3-connected components. This implies that, assuming dimension at least 2, the Weisfeiler-Leman dimension needed to distinguish two graphs is at most the dimension that suffices to distinguish non-isomorphic 3-connected components of the graphs.

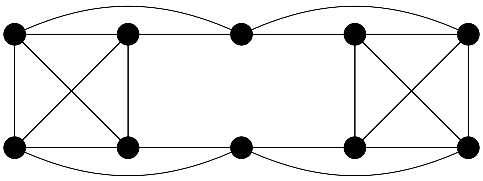


Figure 7.6: A 2-connected graph. The only vertices contained in a 2-separator are the two central ones. Being 4-regular, the graph is stable with respect to the 1-dimensional Weisfeiler-Leman algorithm.

Our results obtained for the 2-dimensional Weisfeiler-Leman algorithm can also be viewed in a combinatorial setting. In 1985, Brouwer and Mesner [29] proved that the vertex connectivity of a strongly regular graph equals its degree and that in fact, the only minimal disconnecting vertex sets are neighbourhoods. Later, Brouwer conjectured this to be true for more general relational structures (namely, constituent graphs of association schemes) and the validity of this statement is still open [27] (see also [28, 43, 111] for related work). Our findings in Theorem 7.17 imply that every connected constituent graph of an association scheme is either a cycle or 3-connected, which implies that, in order to prove Brouwer's conjecture, again, it suffices to consider 3-connected graphs.

As an application of the structural insights we have gained, [99] investigates the Weisfeiler-Leman dimension of graphs parameterised by their treewidth. Until the publication of our results, the best known upper bound on the Weisfeiler-Leman dimension of graphs of treewidth k was $k+2$ [67]. We could improve the upper bound and also provide a lower bound, which is asymptotically only a factor of 2 away from the upper bound, as stated in the following lemma. Here $\dim_{\mathsf{WL}}$ denotes the Weisfeiler-Leman dimension.

Lemma 7.39. *Let $k \in \mathbb{N}_{\geq 2}$. Then $\lceil \frac{k}{2} \rceil - 2 \leq \dim_{\mathsf{WL}}(\mathcal{T}_k) \leq k$, where $\mathcal{T}_k$ denotes the class of graphs of treewidth at most k.*

To keep the presentation compact, we refer the reader to [99] for the details of the proof. Roughly speaking, for the upper bound, we adapt the proof strategy from [67] and use the correspondence of the k-dimensional Weisfeiler-Leman algorithm to the bijective $(k+1)$-pebble game. Given a graph G of treewidth at most k and a second, non-isomorphic graph H, we detect the non-isomorphism by describing a winning strategy for Spoiler that forces a descent (towards the leaves) in a suitable tree decomposition of G. To define the winning strategy, we highly exploit the ability of the k-dimensional Weisfeiler-Leman algorithm to detect k-separators.

For the lower bound, we apply the Cai, Fürer, Immerman construction to the graph defined as the $((k+1) \times (k+1))$-square grid and obtain a graph and its non-isomorphic twist (cf. Section 6.4 and [30]). It can be shown that the treewidths of the two graphs are bounded by $2(k+1)+3 = 2k+5$ and from this, we directly deduce our linear bound.

In particular, the lemma yields a new bound on the Weisfeiler-Leman dimension of any graph class whose treewidth is bounded by a constant. For example, consider the class of outerplanar graphs, i.e. graphs that have a planar embedding in which every vertex lies on the outer (the infinite) face. It is easy to see that every outerplanar graph has treewidth at most 2.

Similarly, it is not difficult to see that every series-parallel graph has tree-width at most 2 (see, for example, [24]). Since the 6-cycle is outerplanar and series-parallel and not identified by the 1-dimensional Weisfeiler-Leman algorithm, we can conclude the following.

Corollary 7.40. *The Weisfeiler-Leman dimension of the class of outerplanar graphs is 2.*

Corollary 7.41. *The Weisfeiler-Leman dimension of the class of series-parallel graphs is 2.*

However, on the one hand, the precise dimension needed to identify arbitrary graphs of a certain treewidth remains open. It might even be possible to describe in an inductive fashion a winning strategy similar to ours that uses less pebbles, i.e. to force the descent with even less pebbles than $k + 1$ pairs. In that case, the Weisfeiler-Leman dimension of the class of graphs of treewidth at most k would be smaller than k.

On the other hand, the exact treewidth of the graphs obtained after applying the Cai, Fürer, Immerman construction is also open. A tighter bound than $2k+7$ would directly improve upon the lower bound stated in Lemma 7.39.

A natural use case of our results is the determination of bounds on the Weisfeiler-Leman dimension of graph classes that satisfy the requirements of Theorem 7.37, i.e. which allow us to focus on the 3-connected graphs in the class. In general, the decomposition of a graph into its 3-connected components is a very powerful tool. Hopcroft and Tarjan were the first to apply this technique to develop an isomorphism test for planar graphs that runs in quasi-linear time [81, 82, 83], which finally led to a linear-time isomorphism algorithm for planar graphs by Hopcroft and Wong [84].

In the next chapter, we apply Lemma 7.38 to deduce that the class of planar graphs has Weisfeiler-Leman dimension at most 3. It thus suffices to show that the 3-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected planar graphs and identifies every such graph.

8 The WL Dimension of Planar Graphs

In this chapter, we are concerned with the Weisfeiler-Leman dimension of the class of planar graphs. The class is a prototype for an application of the techniques presented in the previous chapter, since 3-connected planar graphs have the very useful property that their plane embeddings are unique up to homeomorphism (see [163]). For example, the quasi-linear isomorphism test by Hopcroft and Tarjan highly exploits the decomposition of a planar graph into its 3-connected components [81, 82, 83] and gave way to a linear-time isomorphism test soon afterwards.

Grohe, who introduced the notion of the Weisfeiler-Leman dimension of a graph, was the first to show that there is a dimension of the algorithm which identifies every planar graph [59]. In a tour de force, he generalised this result to graph classes with excluded minors [62, 64]. By these results, in particular, there is a k such that the k-dimensional Weisfeiler-Leman algorithm distinguishes every two non-isomorphic planar graphs from each other. In his Master's thesis [144], Redies analysed Grohe's proof, showing that k can be chosen to be 14 (see also [64, Subsection 18.4.4]). For 3-connected planar graphs, it was also shown earlier by Verbitsky [159] that one can additionally require the quantifier rank of the corresponding C^{1}5-formula that identifies the given graph to be logarithmic in the graph order.

In this chapter, building on the reduction from the previous chapter, we show that the Weisfeiler-Leman dimension of the class of planar graphs is at most 3 (see Theorem 8.12). Note that, with Theorem 7.37, it suffices for us to show that the 3-dimensional Weisfeiler-Leman algorithm determines vertex orbits on the class of all (edge-coloured) 3-connected planar graphs and identifies every such graph.

Using Tutte's Spring Embedding Theorem [156], we argue that if in an edge-coloured 3-connected planar graph, there are three vertices each with a unique colour (so-called *singletons*) that lie on a common face, then applying the 1-dimensional Weisfeiler-Leman algorithm yields a discrete colouring of the graph. Thus, as a nice first benefit from this chapter, the described application of Tutte's Spring Embedding Theorem yields a very compact proof that the 4-dimensional Weisfeiler-Leman algorithm identifies every planar graph. We show then that, in most 3-connected planar graphs, it suffices to individualise 2 vertices to obtain the same result.

Our proof actually characterises the exceptions, the 3-connected planar graphs in which we need to individualise 3 vertices. As a by-product, this yields a classification of the 3-connected planar graphs with fixing number 4. We can handle them separately to finish our proof.

The results presented in this chapter were published in [100] and [101].

8.1 Identifying Non-Exceptions with Tutte

Let G be a 3-connected planar graph. In this section, we show that we can always individualise three vertices in G so that applying the 1-dimensional Weisfeiler-Leman algorithm yields a discrete graph. Actually, as we will see at the end of this section, it very often suffices to individualise just two instead of three vertices.

For the remainder of this chapter, we use the following definition of an exception.

Definition 8.1. A graph G is an *exception* if G is a 3-connected planar graph in which there are no two vertices v and w such that $\chi^1_{G_{(v,w)}}$ is the discrete colouring.

Recall from Section 2.3 that, for an uncoloured graph G, we denote by $G_{(v_1,v_2,\dots,v_t)}$ the coloured graph obtained from G by individualising the vertices $v_1, v_2, \dots, v_t$ in that order. More specifically, we let $G_{(v_1,v_2,\dots,v_t)}$ be the coloured graph (G, λ) with

$$\lambda(v) := \begin{cases} i & \text{if } v = v_i \\ 0 & \text{if } v \notin \{v_1, \dots, v_t\}. \end{cases}$$

If G is initially coloured, we ensure that the individualised vertices still obtain fresh unique colours.

Lemma 8.2. *Let G be a 3-connected planar graph and let v_1, v_2, v_3 be vertices of G. If v_1, v_2, v_3 lie on a common face, then $\chi_{G_{(v_1,v_2,v_3)},1}$ is a discrete colouring.*

Proof: Since the dimension $k = 1$ is fixed, we drop it in the subscript of χ in this proof. We intend to use Tutte's Spring Embedding Theorem [156] (see [110, Section 12.3]). Let v_1, v_2, v_3 be vertices on a common face of G. Let $\mu_0 \colon V(G) \setminus \{v_1, v_2, v_3\} \to \mathbb{R}^2$ be an arbitrary mapping that satisfies $\mu_0(v_1) = (0, 0)$, $\mu_0(v_2) = (1, 0)$, and $\mu_0(v_3) = (0, 1)$. For $i \in \mathbb{N}_0$, we define μ_{i+1} recursively by setting

$$\mu_{i+1}(v) = \begin{cases} \frac{1}{d(v)} \sum_{w \in N(v)} \mu_i(w) & \text{if } v \notin \{v_1, v_2, v_3\}, \\ \mu_i(v) & \text{otherwise.} \end{cases}$$

Then Tutte's result says that this recursion converges to a barycentric planar embedding of G, that is, an embedding in which every vertex not contained in $\{v_1, v_2, v_3\}$ is contained in the convex hull of its neighbours [156, 110]. In particular, this implies that from a certain i on, the mapping μ_i is injective, i.e. no two vertices are mapped to the same location. Choose μ_0 with the requirements above and so that all vertices in $V(G) \setminus \{v_1, v_2, v_3\}$ have the same image. For example, set $\mu_0(v) = (1, 1)$ for $v \in V(G) \setminus \{v_1, v_2, v_3\}$. We argue the following statement by induction on i. For every pair of vertices v and v', it holds that

$$\mu_i(v) \neq \mu_i(v') \quad \Rightarrow \quad \chi^i_{G_{(v_1,v_2,v_3)}}(v) \neq \chi^i_{G_{(v_1,v_2,v_3)}}(v').$$

For $i = 0$, the statement holds by the definition of μ_0 and the fact that v_1, v_2, and v_3 are singletons in $G_{(v_1,v_2,v_3)}$. For $i > 0$, if $\sum_{w \in N(v)} \mu_i(w) \neq \sum_{w' \in N(v')} \mu_i(w')$, then $\{\!\{\mu_i(w) \mid w \in N(v)\}\!\}$ and $\{\!\{\mu_i(w') \mid w' \in N(v')\}\!\}$ are different and thus, by induction, the multisets $\left\{\!\!\left\{\chi^i_{G_{(v_1,v_2,v_3)}}(w) \mid w \in N(v)\right\}\!\!\right\}$ and $\left\{\!\!\left\{\chi^i_{G_{(v_1,v_2,v_3)}}(w') \mid w' \in N(v')\right\}\!\!\right\}$ are different.

By the Spring Embedding Theorem, for some integer i, the map μ_i is injective, implying that $\chi^i_{G_{(v_1,v_2,v_3)}}$ and therefore also $\chi_{G_{(v_1,v_2,v_3)}}$ is a discrete colouring. □

From the lemma, we directly conclude that for $k \geq 4$, the k-dimensional Weisfeiler-Leman algorithm determines the vertex orbits on the class of all edge-coloured 3-connected planar graphs and identifies them.

Corollary 8.3. *For $k \geq 4$, the k-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected planar graphs and identifies these graphs.*

Proof: Let G be an edge-coloured 3-connected planar graph and let $n := |G|$ be its order. Then, by Lemma 8.2, there are vertices v_1, v_2, v_3 such that $\chi_{G_{(v_1,v_2,v_3)},1}$ is discrete, since the additional edge colouring can only refine the stable colouring of the uncoloured graph. This implies that the multiset $C := \{\!\{\chi_{G,4}(v_1, v_2, v_3, x) \mid x \in V(G)\}\!\}$ contains n different colours. Let H be a second edge-coloured graph. If H contains vertices v'_1, v'_2, v'_3 such that $\{\!\{\chi_{H,4}(v'_1, v'_2, v'_3, x') \mid x' \in V(H)\}\!\} = C$, then G and H are isomorphic via an isomorphism that maps v_1 to v'_1. Otherwise, the colour $\chi_{G,4}(v_1, v_2, v_3, v_3)$ is for all $v'_1, v'_2, v'_3 \in V(H)$ different from the colour $\chi_{H,4}(v'_1, v'_2, v'_3, v'_3)$, implying that the sets of vertex colours computed by the 4-dimensional Weisfeiler-Leman algorithm in G and H are disjoint and thus, the two graphs are distinguished. Hence, G is identified. □

The statement of Corollary 8.3 also holds for $k = 3$. This will follow from the fact that, typically, it suffices to individualise just two instead of three vertices. There are some 3-connected planar graphs for which this is not the case (cf. Definition 8.1). However, we precisely determine the collection of such exceptions.

Theorem 8.4. *If G is an exception (i.e. G is a 3-connected planar graph without a pair of vertices v, w such that $\chi_{G_{(v,w)},1}$ is discrete), then G is isomorphic to one of the graphs in Figure 8.1.*

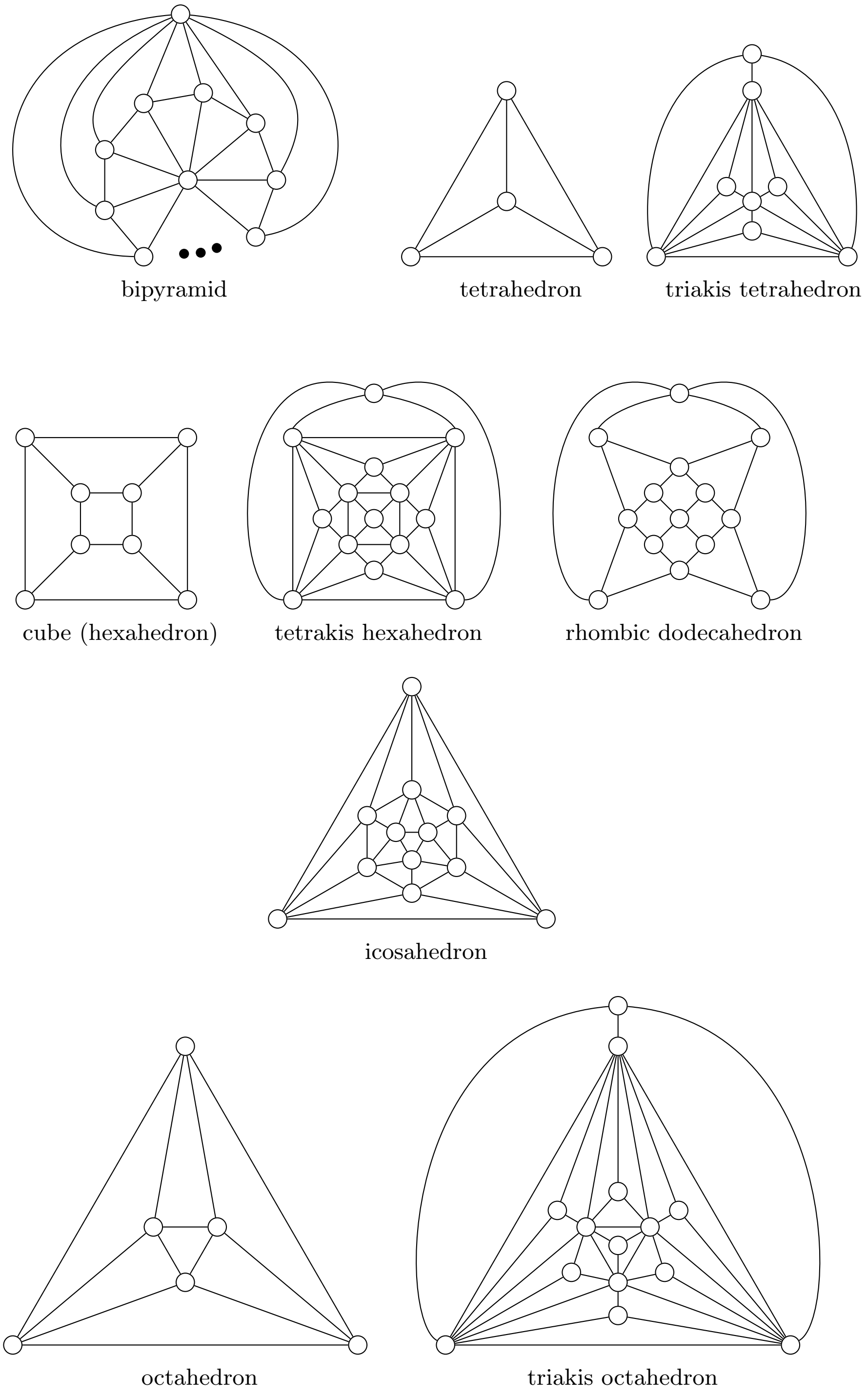


Figure 8.1: The 3-connected planar graphs with fixing number 3.

Before we present the lengthy proof of the theorem, we state its implications.

The fixing number of a graph G is the minimum size of a set of vertices S such that the only automorphism that fixes S pointwise is the identity.

Corollary 8.5. *Let G be a 3-connected planar graph. The fixing number of G is at most 3 with equality attained if and only G is isomorphic to an exception (i.e. one of the graphs depicted in Figure 8.1).*

Proof: If in a given graph G, there is a set of ℓ vertices such that individualising all vertices in the set and then applying the 1-dimensional Weisfeiler-Leman algorithm yields a discrete colouring, then ℓ is an upper bound on the fixing number of G. By Definition 8.1, a graph that is not an exception has fixing number at most 2. To conclude the corollary, it thus suffices to check that all exceptions have fixing number 3, which can be done using Theorem 8.4. □

Corollary 8.6. *For $k \geq 3$, the k-dimensional Weisfeiler-Leman algorithm determines vertex orbits on the class of all edge-coloured 3-connected planar graphs and identifies these graphs.*

Proof: Let G be an edge-coloured 3-connected planar graph that is not an exception. Then there are vertices v and w such that $\chi_{G_{(v,w)},1}$ is discrete. Analogously to the proof of Corollary 8.3, we obtain that for every second edge-coloured graph H, the 3-dimensional Weisfeiler-Leman algorithm only assigns equal colours to a vertex of G and a vertex of H if there is an isomorphism mapping the first to the second. Thus, G is identified and its vertex orbits are determined.

A thorough but tedious case analysis by hand for each exception (which could also be performed by a computer, e.g. using COCO2P [107]) shows that every edge-coloured exception is distinguished from all non-isomorphic edge-coloured graphs and that, on each edge-coloured exception, the χ_3-colouring induces the vertex orbit partition on the vertices. □

8.2 Taming the Exceptions

The task in the remainder of this section is to show Theorem 8.4. The proof of the theorem is a lot more involved than the proof of Corollary 8.3. Thus, at the expense of increasing k by 1 from 3 to 4 in our main result of this chapter (see Theorem 8.12), the reader may skip the following lengthy exposition.

To show Theorem 8.4, we characterise the exceptions in a case-by-case analysis with respect to the existence of vertices of certain degrees. The following two lemmas serve as general tools to deduce information about the structure of the input graphs.

For a subgraph G' of a graph G, we say that $v \in G'$ is *saturated in G' with respect to G* if $\deg_{G'}(v) = \deg_G(v)$. Thus, if a vertex is saturated, then its neighbours in G and in G' are the same.

Lemma 8.7. *Let G be a 3-connected planar graph.*

1. *Let G' be a subgraph of G and suppose that the sequence $v_1, \dots, v_t$ forms a facial cycle in the planar embedding of G' induced by a planar embedding of G. If in $\{v_1, \dots, v_t\}$, there are at most two vertices that are not saturated in G' with respect to G, then it holds that $V(G) = \{v_1, \dots, v_t\}$ or $v_1, \dots, v_t$ is a facial cycle of G.*

2. *If $v_1, \dots, v_t$ is a 3-cycle that contains a vertex of degree 3 in G or $v_1, \dots, v_t$ is an induced 4-cycle of G that contains at least two vertices of degree* 3 *in G, then $V(G) = \{v_1, \dots, v_t\}$ or $v_1, \dots, v_t$ is a facial cycle of G.*

Proof sketch: For Part 1, suppose $v_1, \dots, v_t$ forms a facial cycle of G' and there are at most two vertices v_i and v_j that are not saturated in G' with respect to G. Assume $v_1, \dots, v_t$ is not a facial cycle in G. Then the vertices v_i and v_j are the only ones among $v_1, \dots, v_t$ that have neighbours in G inside the region of the plane corresponding to the facial cycle of G' that is formed by $v_1, \dots, v_t$. Therefore, if $V(G) \neq \{v_1, ..., v_t\}$, then $\{v_i, v_j\}$ is a 2-separator of G, which contradicts the 3-connectedness.

For Part 2, consider an induced 4-cycle $v_1, \dots, v_4$ in which there exist two vertices v_i and v_j of degree 3 in G. If $v_1, \dots, v_4$ is not a facial cycle of G, then $\{v_1, \dots, v_4\} \setminus \{v_i, v_j\}$ is a separator of size 2. The argument for a 3-cycle v_1, v_2, v_3 is similar. □

To simplify notation, for every occurring graph G, we let $\chi_G \coloneqq \chi_{G,1}$ in the remainder of this chapter, i.e. we drop the (fixed) dimension $k = 1$ in the subscript of the colourings.

Lemma 8.8. *Let G be an exception and let v be a vertex of G. Let $u_1, \dots, u_{d(v)}$ be the cyclic ordering of the neighbours of v induced by a planar embedding of G. Then every pair of vertices u_i and u_{i+1} has a common neighbour of degree $d(v)$ other than v.*

Proof: If u_i and u_{i+1} do not have a common neighbour of degree $d(v)$ other than v, the colouring $\chi_{G_{(u_i,u_{i+1})}}$ has the three singletons u_i, v, u_{i+1}, which lie on a common face. Thus, by Lemma 8.2, the colouring $\chi_{G_{(u_i,u_{i+1})}}$ is discrete, contradicting the assumption that G is an exception. □

For an exception G with a distinguished vertex v and $N(v) \coloneqq \{u_1, \dots, u_t\}$, to reason about planarity, we will often consider the graph G^v obtained from G by inserting all edges $\{u_i, u_{i+1}\}$. It is easy to see that, since G is planar, the graph G' is also planar.

Now we can start determining the structure of the exceptions. By definition, every exception is 3-connected. Hence, no exception has a vertex of degree 1 or 2. Since every planar graph has a vertex of degree at most 5, every exception contains a vertex of degree 3, 4, or 5. We begin with the exceptions that contain a vertex of degree 5.

Lemma 8.9. *If G is an exception that has a vertex of degree 5, then it is isomorphic to the icosahedron or the bipyramid on 7 vertices.*

Proof: Let G be an exception with a vertex v of degree 5. Let $N := N(v)$ be the set of neighbours of v and let $(u_1, \dots, u_5)$ be their cyclic ordering induced by a planar embedding of G. For convenience, we will take indices modulo 5.

By Lemma 8.8, every pair of vertices u_i, u_{i+1} has a common neighbour $x_{i,i+1}$ of degree 5 other than v. (We remark that the vertices $x_{i,i+1}$ are not necessarily distinct or unique.)

Claim 1. For all $i \in [5]$, every pair of vertices u_i, u_{i+2} has a common neighbour $x_{i,i+2}$ of degree 5 other than v.

Proof of the claim: We show the claim for $i = 1$, the other cases follow analogously. Assume that u_1 and u_3 do not have a common neighbour of degree 5 in G other than v. Thus, in the colouring $\chi_{G_{(u_1,u_3)}}$, the vertex v is a singleton. If there are two consecutive vertices in N that are singletons, then $\chi_{G_{(u_1,u_3)}}$ is discrete by Lemma 8.2, which contradicts the assumption that G is an exception. It follows that N is the union of three colour classes of the colouring $\chi_{G_{(u_1,u_3)}}$, one of which is $\{u_2, u_4, u_5\}$. We distinguish cases.

(Case A: $x_{1,2} \in N$ or $x_{2,3} \in N$.) We only consider the case that $x_{1,2} \in N$, since the case $x_{2,3} \in N$ is analogous. We know that u_2, u_4 and u_5 have the same degree in $G[N]$. Thus, the vertices u_1 and u_3 must have the same degree in $G[N]$. Indeed, otherwise one of them would have a unique degree in $G[N]$ and then some u_i would be a singleton in $\chi_{G_{(v)}}$, rendering $\chi_{G_{(v,u_{i+1})}}$ discrete by Lemma 8.2.

Now suppose first $x_{1,2} = u_4$ or $x_{1,2} = u_5$. Either way, one and thus all of the vertices u_2, u_4, u_5 are adjacent to u_1 because they form a colour class of $\chi_{G_{(u_1,u_3)}}$. It follows that the degree of u_1 and hence also of u_3 in $G[N]$ is at least 3. Thus, u_1 and u_3 have a common neighbour among $\{u_2, u_4, u_5\}$, which must have degree 5 in G, since $x_{1,2}$ has degree 5.

In the case $x_{1,2} = u_3$, we can deduce that all vertices of u_2, u_4, u_5 are adjacent to u_3 and thus treat the case analogously by simply swapping the roles of u_1 and u_3.

(Case B: $x_{1,2} \notin N$ and $x_{2,3} \notin N$.) Since u_2 and u_4 have the same colour and u_1 has a unique colour in $\chi_{G_{(u_1,u_3)}}$, the vertices u_1 and u_4 have a common neighbour $x \notin N$ of degree 5 other than v. Similarly, u_3 and u_5 have a common neighbour $x' \notin N$ of degree 5 other than v. To show that $x = x'$, consider the graph $G' := G^v$ obtained from G by inserting edges between every pair u_i and u_{i+1}. If $x \neq x'$, then G' contains $K_{3,3}$ as a minor (by contracting the paths of length 2 from u_1 to u_4 via x, and from u_3 to u_5 via x'), contradicting its planarity.

Thus, we have $x = x'$, and therefore, x is a vertex of degree 5 adjacent to u_1 and to u_3. ⌟

Having proved the claim, we now finish the proof of Lemma 8.9. Again, we distinguish two cases.

(Case 1: $G[N]$ is non-empty and some vertex in N has degree 5 in G.) Since G is planar, there can be at most one vertex in N that has degree 4 within $G[N]$. Indeed, if there were two such vertices u_i and u_j, then each of v, u_i, u_j would be adjacent to all vertices in $N \setminus \{u_i, u_j\}$, yielding a $K_{3,3}$ minor. However, as seen in the analysis of Case A of Claim 1, no vertex in N can have a unique degree in $G[N]$, and thus, $G[N]$ has a maximum degree of at most 3.

Suppose that $G[N]$ contains an edge $\{u_i, u_{i+2}\}$ for some i, say $i = 1$. Due to Claim 1, the vertex u_2 must share a common neighbour with each of u_4 and u_5 apart from v. By the planarity of G', these common neighbours are in $\{u_1, u_3\}$. However, the two common neighbours must be different since otherwise, the respective vertex has degree $|\{u_1, u_2, u_4, u_5\}| = |\{u_3, u_2, u_4, u_5\}| = 4$ in $G[N]$. Thus, u_2 is adjacent to u_1 and u_3, and both u_1 and u_3 have degree 3 in $G[N]$.

By the planarity of G', the vertex u_2 has degree 2 in $G[N]$. Suppose u_4 has degree 3 in $G[N]$. Then it must be adjacent to u_1, u_3, and u_4. Since u_5 cannot have a unique degree in $G[N]$, it must be adjacent to u_3, i.e. it must have degree 2 in $G[N]$. (By planarity, it cannot be adjacent to u_1.) However, this would give u_3 a degree of 4 in $G[N]$, yielding a contradiction. The case that u_5 has degree 3 in $G[N]$ is symmetric. Thus, u_1 and u_3 are the only vertices of degree 3 in $G[N]$.

Then u_2 is the only vertex of N that is adjacent to two vertices of degree 3 in $G[N]$, making u_2 a singleton in $\chi_{G_{(v)}}$ and yielding a contradiction.

We conclude that there is no edge of the form $\{u_i, u_{i+2}\}$. This implies that, in $G[N]$, there is no vertex of degree 1. (Otherwise, we could individualise this vertex and v, and, together with their unique common neighbour, they would yield a discrete colouring by Lemma 8.2.) Consequently, since $G[N]$ is non-empty, we conclude that u_1, u_2, u_3, u_4, u_5 is an induced cycle in $G[N]$. Since some vertex in N has degree 5 and within $G[N]$ the two neighbours of each vertex must have the same degree, we conclude that all vertices in N have degree 5.

Thus, if a vertex fulfils Case 1, then all its neighbours also fulfil Case 1. Therefore, being connected, the entire graph G must be a 5-regular triangulated planar graph, since we have restricted ourselves to connected graphs. Being 5-regular, the graph has $m = \frac{5n}{2}$ edges. The Euler formula states that $2 = n - m + f$, where f is the number of faces of a plane embedding of the graph. Since G is triangulated, every facial cycle contains exactly 3 edges and every edge is contained in exactly 2 faces. Thus, we have $f = \frac{2m}{3}$, and hence, $m = 3n - 6$. We conclude that $n = 12$. There is only one 5-regular graph on 12 vertices, namely the icosahedron (see, for example, [76]).

(Case 2: $G[N]$ is empty or no vertex in N has degree 5 in G.) By Claim 1, we already know that every pair of vertices u_i and u_{i+2} has a common neighbour of

degree 5 other than v. In the current case, this vertex cannot be in N. Due to planarity, all these common neighbours for different i must be equal to a single vertex y adjacent to all vertices of N.

Observing that y has degree 5, consider the subgraph $H \coloneqq G[\{v, y, u_1, \ldots, u_5\}]$ of G. With the described cyclic ordering of the vertices in N, there is only one planar drawing of H up to equivalence. In this drawing, every face is a 4-cycle or a 3-cycle containing y and v. Since y and v have degree 5 in G, but they already have degree 5 in H, they are saturated. Thus, since both y and v belong to each face of H, by Part 1 of Lemma 8.7, no interior of a face of the drawing of H contains vertices of G. Therefore, $G = H = G[\{v, y, u_1, \ldots, u_5\}]$. Since G is 3-connected, $G[N]$ cannot be empty. Similarly as in Case 1, we conclude that $u_1, \ldots, u_5$ is a cycle rendering G the bipyramid on 7 vertices. □

Hence, we have characterised all exceptions that contain a vertex of degree 5. We continue with the exceptions that contain a vertex of degree 3.

Lemma 8.10. *If G is an exception that has a vertex of degree 3, then it is isomorphic to a tetrahedron, a cube, a rhombic dodecahedron, a triangular bipyramid, a triakis tetrahedron, or a triakis octahedron.*

Proof: Assume G is a 3-connected planar graph with a vertex of degree 3 and that G does not have two vertices whose individualisation followed by an application of the 1-dimensional Weisfeiler-Leman algorithm produces the discrete partition.

Let v be a vertex of degree 3 in G and let $N \coloneqq N(v) = \{u_1, u_2, u_3\}$ be its neighbours. By Lemma 8.2, no vertex of N can have a unique degree in G. Thus, we know that the graph $G[N]$ is either a triangle or empty.

By Lemma 8.8, for $i \in [3]$ (indices always taken modulo 3), the vertices u_i and u_{i+1} have a common neighbour $x_{i,i+1}$ of degree 3 other than v. (As in the previous proof, these $x_{i,i+1}$ are not necessarily distinct or unique.) If $x_{i,i+1} \in N$, then $x_{i,i+1} = u_{i+2}$ and $N(\{u_{i+2}, v\}) = \{u_i, u_{i+1}\}$. Thus, unless G only has 4 vertices, in which case it is the tetrahedron, the set $\{u_i, u_{i+1}\}$ forms a 2-separator, which contradicts G being 3-connected. We can therefore assume, for all $i \in \{1, 2, 3\}$, that $x_{i,i+1} \notin N$.

Either $v, u_i, x_{i,i+1}, u_{i+1}$ forms a facial cycle of G, or the vertices u_i and u_{i+1} are adjacent and both $u_i, u_{i+1}, x_{i,i+1}$ and u_i, u_{i+1}, v are facial cycles. Indeed, the vertex $x_{i,i+1}$ has degree 3, so the claim follows directly from Part 2 of Lemma 8.7.

(Case 1: $G[N]$ is empty.)

In this case, every face incident with v is a 4-cycle consisting of two non-adjacent vertices v and $x_{i,i+1}$ of degree 3 in G and two other vertices u_i and u_{i+1} of degree $d \geq 3$. By analogous arguments as for v, for every $i \in [3]$, the graph $G[N(x_{i,i+1})]$ must be either empty or a triangle. It cannot be a triangle because the edge $\{u_i, u_{i+1}\}$ is not present. So, by successively replacing v with its opposite vertices $x_{i,i+1}$ in the incident 4-cycles, we conclude that G is a $(3, d)$-biregular quadrangulation.

(Case 1.1: G is 3-regular.) In this subcase, G must be a 3-regular quadrangulation. Such a graph has $m = \frac{3n}{2}$ edges, since it is 3-regular, but, by the Euler formula, it also holds that $m = 2n - 4$, since the graph is a quadrangulation. Thus, $n = 8$. It is easy to verify that the only 3-regular planar quadrangulation on 8 vertices is the cube.

(Case 1.2: G is not 3-regular.) In this subcase, G is bipartite and biregular with degrees 3 and d. Let n_3 and n_d be the number of vertices of degree 3 and d, respectively. Then $3n_3 = dn_d$ by double-counting and $dn_d = m = 2n - 4$, since G is a quadrangulation. It follows that $dn_d = 2(n_3 + n_d) - 4 = 2(\frac{dn_d}{3} + n_d) - 4$, which gives that $4 = n_d(2 - \frac{d}{3})$. Thus, $d \leq 5$ and by 3-connectedness, $d \in [3, 5]$. The case $d = 3$ is Case 1.1 and $d = 5$ cannot occur due to Lemma 8.9, since neither the icosahedron nor the bipyramid on 7 vertices has vertices of degree 3.

We conclude that G is a $(3, 4)$-biregular quadrangulation. We have $3n_3 = 4n_4$. Then since $n = n_3 + n_4$, it holds that $m = 3n_3 = 3 \cdot \frac{4n}{7}$, but, by the Euler formula, also $m = 2n - 4$. Thus, $n = 14$ and $n_3 = 8$.

Every vertex of degree 3 is incident with 3 faces and thus has 3 “opposite vertices” in the bipartite graph G. Thus, if we add an edge between every pair of vertices of degree 3 that lie on a common face and remove all vertices that originally had degree 4, we obtain a new 3-regular planar quadrangulation on $n_3 = 8$ vertices. The only such graph is the cube. Undoing the modification, we deduce that G is the rhombic dodecahedron.

This concludes the case that $G[N]$ is empty.

(Case 2: $G[N]$ is a triangle.)

In this case, since we know that all $x_{i,i+1}$ have degree 3, by substituting $x_{i,i+1}$ for v, we obtain that every face of G is a 3-cycle consisting of a vertex of degree 3 and two other vertices of equal degree d.

(Case 2.1: G is 3-regular, i.e. $d = 3$.) Then the graph G is a 3-regular triangulation. Thus, $m = \frac{3n}{2}$ and, by the Euler formula, $m = 3n - 6$, hence $n = 4$. We conclude that G is the tetrahedron.

(Case 2.2: G is not 3-regular.) Consider the graph $\widetilde{G}$ obtained from G by removing all vertices of degree 3. The resulting graph is a planar $\frac{d}{2}$-regular triangulation. Indeed, since v was chosen arbitrarily among all vertices of degree 3 and the vertices $x_{i,i+1}$ have degree 3 as well, it is easy to see that the resulting graph is a triangulation. Moreover, one can verify that, in G, for every vertex of degree d, in the cyclic ordering of its neighbours, the degrees 3 and d alternate. Thus, a deletion of the vertices of degree 3 halves the degrees of each of the other vertices.

Since for $d' > 5$, no d'-regular graph is planar, it follows that $\frac{d}{2} \in [2, 5]$. For $\frac{d}{2} = 2$, we obtain a single 3-cycle. This implies that G is a triangular bipyramid.

For $\frac{d}{2} = 3$, since $\widetilde{G}$ is a triangulation, all vertices in N have (in $\widetilde{G}$) a single common neighbour w and every vertex in $G'[N \cup \{w\}]$ is saturated with respect to G'. Thus, we conclude that $\widetilde{G}$ is a tetrahedron. This implies that G is a triakis tetrahedron.

For $\frac{d}{2} = 4$, we obtain that $\widetilde{G}$ must be an octahedron. This implies that G is the triakis octahedron. For $\frac{d}{2} = 5$, we would obtain an icosahedron. This would imply that G is the triakis icosahedron. However, in this solid, there are two vertices u and u' (namely vertices of degree 3 of distance 2 that only have one common neighbour) such that $\chi_{G_{(u,u')},1}$ is discrete. Thus, this graph is not an exception. □

It remains to analyse the exceptions that contain no vertices of degree 3 or 5. Every such exception must contain a vertex of degree 4.

Lemma 8.11. *If G is an exception that has a vertex of degree 4, then it is isomorphic to a bipyramid or a tetrakis hexahedron.*

Proof: Assume G is an exception with a vertex of degree 4. First we make two observations that hold for every vertex u of degree 4 with neighbours v_1, v_2, v_3, v_4 in cyclic order.

Observation 1: It holds that $\deg(v_1) = \deg(v_3)$ and $\deg(v_2) = \deg(v_4)$. Otherwise, we can individualise u and a neighbour v_i of u so that v_{i+1} and v_{i-1} refine to a singleton class in the colouring resulting from an application of the 1-dimensional Weisfeiler-Leman algorithm, which by Lemma 8.2 yields a contradiction to the assumption of G being an exception.

Observation 2: By a similar argument, the induced graph $G[v_1, v_2, v_3, v_4]$ either is empty or contains a cycle in which v_i is adjacent to v_{i+1} for all $i \in [4]$ (indices taken modulo 4).

Due to Observation 2, if G is 4-regular, then either G is a triangulation or every face is of size at least 4. In the first case, G has $n = 6$ vertices, since $m = \frac{4n}{2}$ and, by the Euler formula, $m = 3n - 6$. Thus, G is the octahedron, which is a bipyramid.

The second case cannot occur. Indeed, a planar graph without triangular faces has at most $\frac{2m}{4} = \frac{m}{2}$ faces, since every edge is contained in exactly 2 faces and every facial cycle contains at least 4 edges. Thus, by the Euler formula, the graph has at most $2n - 4$ edges, but a 4-regular graph has $\frac{4n}{2} = 2n$ edges.

We can thus assume that G is not 4-regular. Hence, the graph has a vertex v of degree other than 4 that is adjacent to a vertex of degree 4. By Lemmas 8.9 and 8.10, we can assume that G neither has a vertex of degree 5 nor a vertex of degree 3. Thus, we can assume that v has degree at least 6. Let $N := N(v)$ be the set of neighbours of v, and let $N_4 \subseteq N$ be those neighbours of v that are of degree 4. Suppose $(u_1, \ldots, u_t)$ is the cyclic ordering of N_4 induced by the cyclic ordering of N obtained from a planar embedding of G.

(Case 1: $E(G[N_4])$ is non-empty.) Assume there are $u, u' \in N_4$ that are adjacent, i.e. $E(G[N_4])$ is non-empty. According to Observation 1, each vertex in N_4 must have two neighbours in G of degree $\deg(v) \neq 4$, i.e. outside of N_4, and thus, $G[N_4]$ has maximum degree 2. We argue that $G[N_4]$ cannot have a vertex of degree 1. Indeed, if u_i were a vertex of degree 1 in $G[N_4]$, then u_i would have two neighbours of degree $\deg(v)$ and two neighbours of degree 4, one of which contained in N_4, i.e. adjacent to v, and one of which non-adjacent to v. This means that in $\chi_{G_{(v,u_i)}}$, all neighbours of u_i would be singletons, since the two neighbours of u_i of degree 4 would disagree on being adjacent to v or not. This is impossible by Lemma 8.2. We conclude that $G[N_4]$ has only vertices of degrees 2 and 0. Since $G[N_4]$ is non-empty, this implies that there is some cycle in $G[N_4]$. Assume this cycle has an edge $\{u_i, u_j\}$ connecting two vertices that are not consecutive in the cyclic order $(u_1, \dots, u_t)$.

Let u_i^+ and u_i^- be the vertices of $G[N]$ following and preceding, respectively, the vertex u_i in the cyclic ordering of N (so they may or may not have degree 4). By Lemma 8.8, there must be vertices x^+ and x^- of degree $\deg(v) \neq 4$ such that x^+ is adjacent to both u_i and u_i^+, and x^- is adjacent to both u_i and u_i^-. However, $x^+ \neq x^-$, since the cycle v, u_i, u_j separates u_i^+ from u_i^-. We conclude that u_i has the following five neighbours: the vertex v, two neighbours in N_4, as well as x^+ and x^-. However, u_i has degree 4, which gives a contradiction. We conclude that u_i is adjacent to u_{i+1} for all $i \in [t]$.

Finally, by Observation 1, every pair of vertices $\{u_i, u_{i+1}\}$ must have a common neighbour of degree $\deg(v)$ other than v, since otherwise, all neighbours of u_i would be singletons in $\chi_{G_{(u_i,u_{i+1})}}$. Since u_i is adjacent to v, u_{i-1}, and u_{i+1} and has degree 4, it can only have one further neighbour. Thus, all these common neighbours for the pairs $\{u_i, u_{i+1}\}$ are indeed the same vertex x. Consider $\widetilde{G} := G[\{u_1, \dots, u_t, v, x\}]$. The graph $\widetilde{G}$ is a bipyramid, in particular, it is 3-connected and every face is a 3-cycle with two vertices saturated in $\widetilde{G}$ with respect to G. Part 1 of Lemma 8.7 implies that $G = \widetilde{G}$. We conclude that G is isomorphic to a bipyramid.

(Case 2: $E(G[N_4])$ is empty.) We now assume that the neighbours of v that have degree 4 form an independent set.

Claim 1. For every $i \in [t]$, there is a vertex $x_{i,i+1}$ such that either the sequence $v, u_i, x_{i,i+1}, u_{i+1}$ forms a facial cycle or both $v, u_i, x_{i,i+1}$ and $v, u_{i+1}, x_{i,i+1}$ form facial cycles.

Proof of the claim: Without loss of generality, we show the claim for $i = 2$. We first argue that there are vertices $u' \in N_4$ and x' such that v, u_2, x', u' is a 4-cycle. For this, let x' be the first neighbour preceding v in the cyclic order among the neighbours of u_2. Then, by Lemma 8.8, there must be a vertex u' other than u_2 of degree 4 that is adjacent to x' and v.

Let $u_j \in N_4$ be the first neighbour of v following u_2 in the cyclic order of vertices in N_4 that has a common neighbour with u_2 other than v. Note that u' is a candidate, thus, the existence of u_j is guaranteed.

We choose a common neighbour x of u_2 and u_j so that it is closest to v: more precisely, for a common neighbour $x \neq v$ of u_2 and u_j, consider the cycle u_2, v, u_j, x. It bounds two areas, one of which contains the vertices of N that follow u_2 but precede u_j, while the other one contains the vertices that follow u_j but precede u_2 in the cyclic ordering of N. (One of these sets may be empty.) We choose x so that the first of these areas is minimal with respect to inclusion and we call this area A.

We claim that the 4-cycle v, u_2, x, u_j is a facial cycle of G or a facial cycle after removing the diagonal $\{v, x\}$ (i.e. the 3-cycles v, u_2, x and v, u_j, x form faces). Note that the edge $\{u_2, u_j\}$ cannot be present since $G[N_4]$ is empty. Now, towards a contradiction, suppose that u_2 has a neighbour that lies within A. Choose as such a neighbour z the vertex that precedes v in the cyclic ordering of the neighbours of u_2. Then by Lemma 8.8, for some vertex $\bar{u} \in N_4$, the sequence $v, u_2, z, \bar{u}$ forms a 4-cycle. Now, by planarity, since z lies in A, the vertex $\bar{u}$ cannot lie outside A. Thus, either $\bar{u}$ precedes u_j in the cyclic ordering of N_4 starting from u_2, or $\bar{u} = u_j$. In both cases, the cycle through v, u_2, z, u_j bounds an area that is a proper subset of A, which contradicts the minimal choices of u_j and x.

Finally, assume that u_j has a neighbour z that lies within A. Choose z to be the vertex that follows v in the cyclic order of the neighbours of u_j. Since $\{v, x\}$ is not a separator and u_2 does not have a neighbour inside A, there must be a path from z to u_2 avoiding both v and x, thus, a path via u_j that leaves A. So the vertex u_j must have a neighbour outside of A. Therefore, since u_j has degree 4 and is also adjacent to v and x, the vertex z is the only neighbour of u_j inside A.

By Lemma 8.8, there is a vertex $\bar{u} \in N_4$ such that the sequence $v, \bar{u}, z, u_j$ forms a 4-cycle. By the minimality of x, we cannot have that $\bar{u} = u_2$. Consider the colouring $\chi_{G_{(u_2,\bar{u})}}$. In this colouring, v is a singleton. Indeed, since z lies in A, the vertex $\bar{u}$ must also lie in A. Thus, by the minimality of u_j, the vertices u_2 and $\bar{u}$ cannot have a common neighbour apart from v. Furthermore, u_j is the only vertex in N_4 that has simultaneously a common neighbour with u_2 other than v and a common neighbour with $\bar{u}$ other than v. Hence, u_j is a singleton with respect to $\chi_{G_{(u_2,\bar{u})}}$. Moreover, since $N(u_j) \cap A = \{z\}$ and $\{u_2, x\} \cap N(\bar{u}) = \emptyset$, the vertices $\bar{u}$ and u_j only have one common neighbour other than v, namely z, which is thus a singleton as well. The singletons v, u_j, and z lie on a common face by the choice of z, which, by Lemma 8.2, yields a contradiction to the assumption that G is exception.

Thus, neither u_2 nor u_j has a neighbour inside the cycle. This implies that the sequence v, u_2, x, u_j forms a facial cycle or it becomes a facial cycle after removing the diagonal $\{v, x\}$, since otherwise, the set $\{v, x\}$ would be a separator. We conclude that $u_j = u_3$ and that the vertex $x_{2,3} \coloneqq x$ justifies the claim. ⌟

Overall, the claim implies that in the cyclic order of N, between every pair $u_i, u_{i+1} \in N_4$, there is at most one vertex, namely $x_{i,i+1}$. Thus, at least every second neighbour of v is of degree 4 (in particular $|N| \leq 2|N_4|$) and hence, being of degree at least 6, the vertex v has at least 3 neighbours of degree 4, i.e. $|N_4| \geq 3$.

Recall that u_1 and u_3 are the two vertices that are closest to u_2 in the cyclic ordering of N_4. In the following, we call an edge of G a *diagonal* if neither of its endpoints has degree 4.

We distinguish several cases according to the size of N_4.

(Case 2.1: $|N_4| = 3$.) Since v has degree at least 6 and $|N_4| \geq \frac{|N|}{2}$, we conclude that v also has exactly 3 neighbours of degree larger than 4. Thus, the degree of v is 6. Let $u_1, t_1, u_2, t_2, u_3, t_3$ be the neighbours of v in the cyclic ordering. Then, by Claim 1, the vertices v, u_i, t_i form a facial cycle for every $i \in [3]$. Likewise, the vertices v, t_i, u_{i+1} form a facial cycle. Thus, the graph induced by $N \cup \{v\}$ is a wheel with 7 vertices. By Observation 2, the neighbourhood of u_i contains a cycle (because u_i has degree 4). Moreover, by Observation 1, the vertices t_1, t_2, and t_3 all have the same degree d. Recall that our initial assumption implies that G contains no vertex of degree 3 or 5.

We argue that $d = 6$. Since all u_i have degree 4, if every pair of vertices u_i and u_{i+1} had a common neighbour other than v and t_i, this would have to be a single vertex x adjacent to u_1, u_2, and u_3 (it cannot be adjacent to any t_i since $t_i \notin N_4$). However, such x does not exist. Indeed, otherwise $\widetilde{G} := G[u_1, u_2, u_3, t_1, t_2, t_3, x, v]$ would be a 3-connected graph in which every facial cycle is either a triangle with a saturated vertex or a 4-cycle with two saturated vertices. Then Part 2 of Lemma 8.7 would imply $G = \widetilde{G}$, which cannot happen, since $\widetilde{G}$ has a vertex of degree 3, namely x.

Thus, some pair u_i, u_{i+1} does not have a common neighbour different from v and t_i, which in turn implies $\deg(t_i) = \deg(v) = 6$, using Lemma 8.8. Therefore, all neighbours of v have degree 4 or 6. More strongly, we conclude that the vertex degrees appearing among the neighbours of t_i are the same as the vertex degrees appearing among neighbours of v, including multiplicities. If this did not hold for every $i \in [3]$, there would be a vertex $u_j \in N_4$ such that, in $\chi_{G_{(u_j,v)}}$, every u_i would obtain a unique colour, resulting also in a unique colour for some t_i. Thus, by Lemma 8.2, the graph G would not be an exception. Hence, each t_i also has 3 neighbours of degree 4.

By Observation 1, all vertices in N_4 only have neighbours of degree 6 (since we already know that they have 3 neighbours of degree 6). By a similar argument, these degree 6 vertices have themselves 3 neighbours of degree 4 and 3 neighbours of degree 6. We conclude that the entire graph is $(4, 6)$-biregular. Since every face incident with v is a triangle, and v is arbitrary among the vertices of degree 6, we deduce that G is a triangulation. Moreover, every vertex of degree 4 has exactly 4 neighbours of degree 6 and every vertex of degree 6 has exactly 3 neighbours of degree 4. By double-counting the edges incident with both a vertex of degree 4 and a vertex of degree 6, we conclude that $4n_4 = 3n_6$, where n_i is the number of vertices of G of degree i. Thus, $n_6 = \frac{4n_4}{3}$, and, since $n_4 + n_6 = n = |G|$, we obtain $n_4 = \frac{3n}{7}$. By the structure of G, its number of edges is $4n_4 + \frac{3n_6}{2}$, which equals $n + 3n_4 + \frac{n_6}{2} = n + \frac{11n_4}{3}$. Indeed, $4n_4$ is the number of edges incident with vertices of degree 4 and $\frac{3n_6}{2}$ is the number of edges incident only with vertices of degree 6. We conclude that G has $\frac{18n}{7}$ edges. Since G is a triangulation, by the Euler formula, it has $3n - 6$ edges. Thus, $\frac{18n}{7} = 3n - 6$, and hence, G is a graph on 14 vertices. Furthermore, the graph $\widetilde{G}$ induced by the vertices of degree 6 is a 3-regular graph on $14 - 6 = 8$ vertices. By the structure of the triangulated graph G, all faces in the induced drawing of $\widetilde{G}$ are 4-cycles. We conclude that $\widetilde{G}$ is the cube. (There is only one triangle-free planar 3-regular graph on 8 vertices.) Each face of $\widetilde{G}$ contains, within G, a vertex of degree 4. We conclude that G is the tetrakis hexahedron.

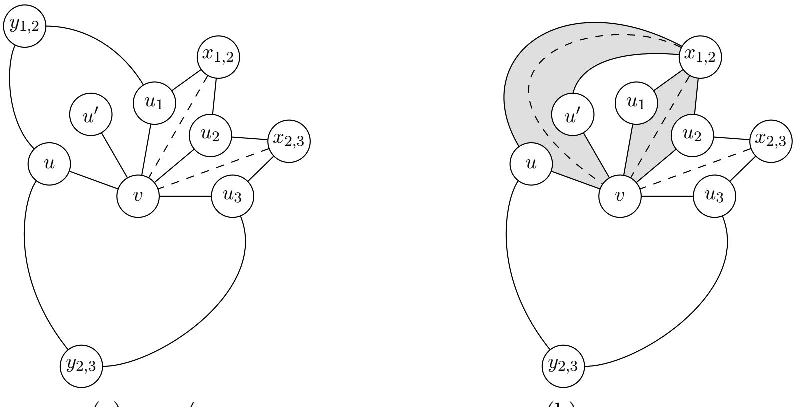
(Case 2.2: $|N_4| = 4$.) In this subcase, v must be adjacent to some of the $x_{i,i+1}$, since otherwise, v has degree 4. Thus, for every $i \in [4]$, the diagonal $\{v, x_{i,i+1}\}$ must be present, since by Observation 2, each neighbourhood $N(u_i)$ contains a cycle in which either every or no vertex is adjacent to its predecessor and its successor in the cyclic ordering of $N(u_i)$. It follows that v must have degree 8 and that $|N_4| = \frac{|N|}{2}$. By a similar argument, using the presence of the diagonals $\{v, x_{i,i+1}\}$ and Observation 2, for every neighbour of v of degree other than 4, at most half of its neighbours have degree 4. We obtain that G is a triangulation that has at most as many vertices of degree 4 as of degree at least 8. (Every vertex of degree $d \geq 8$ is adjacent to at most $\frac{d}{2}$ vertices of degree 4 and every vertex of degree 4 is adjacent to 4 vertices of degree at least 8.) Let n_d be the number of vertices of degree d and let $n_{\geq 8}$ be the number of vertices of degree at least 8. Then, by the 3-connectivity of G and Lemmas 8.9 and 8.10, we can assume $n_1 = n_2 = n_3 = n_5 = 0$ and thus, $n = n_4 + n_{\geq 8} + n_6 + n_7$. We show that G has at least $3n$ edges. Indeed, by summing over the degrees, we have $6n = 6n_4 + 6n_{\geq 8} + 6n_6 + 6n_7 \leq 4n_4 + 8n_{\geq 8} + 6n_6 + 7n_7 \leq 2m$, where the first inequality follows from our observation that $n_4 \leq n_{\geq 8}$. Thus, $m \geq 3n$, which contradicts the bound $m \leq 3n - 6$ obtained from the Euler formula.

(Case 2.3: $|N_4| \geq 5$.) Recall that $t = |N_4|$. We first show the following claim.

Claim 2. For each $i \in [t]$, the vertices u_i and u_{i+2} have a common neighbour other than v.

Proof of the claim: We show the statement for $i = 1$. Assuming otherwise implies that v is a singleton in $\chi_{G_{(u_1,u_3)}}$. We first argue that, in this colouring, the vertex u_2 is also a singleton. Again, assume otherwise. Then there must be a vertex $u \in N_4 \backslash \{u_2\}$ that has the same colour as u_2. By Claim 1, for $i \in \{1, 2\}$, the vertices u_i and u_{i+1} have a common neighbour $x_{i,i+1}$ so that the sequence $v, u_i, x_{i,i+1}, u_{i+1}$ forms a facial cycle or a facial cycle after removing a diagonal.

Thus, in order not to be distinguished from u_2, the vertex u must have a neighbour $y_{1,2}$ other than v that is adjacent to u_1 and a neighbour $y_{2,3}$ other than v that is adjacent to u_3. Moreover, for $i \in \{1, 2\}$, the vertex $y_{i,i+1}$ should have the



(a) $x_{1,2} \neq y_{1,2}$

(b) $x_{1,2} = y_{1,2}$

Figure 8.2: An illustration of Case 2.3 in the proof of Lemma 8.11. Some possible diagonals are shown as dashed lines. The highlighted regions in the drawing on the right cannot contain other vertices, implying that $\{x_{1,2}, v\}$ separates u_1 from u.

same colour in $\chi_{G_{(u_1,u_3)}}$ as $x_{i,i+1}$. See Figure 8.2a. (Note that $y_{1,2} \neq y_{2,3}$, since we have assumed that u_1 and u_3 do not have a common neighbour other than v.)

Since it holds that $|N_4| \geq 5$, we know that $u_4 \neq u_t$. Therefore, $u \neq u_4$ or $u \neq u_t$. By symmetry, we can assume the latter. (To see the symmetry, recall that u_4 is the successor of u_3 in N_4 and u_t is the predecessor of u_1 in N_4.) Note that the cycle $v, u, y_{1,2}, u_1$ separates u_t from u_2. Consider the area A' bounded by the cycle $v, u, y_{1,2}, u_1$ which contains u_t. Inside A' lies the vertex $u' \in N_4$ that follows u in the cyclic ordering of N_4.

Consider the set $M \coloneqq N(\{u_3\}) \cup \{u_1, u_3\}$ and note that M is a union of colour classes. By the assumption of Case 2 and since $N(\{u_1\}) \cap N(\{u_3\}) = \{v\}$, the cycle $v, u, y_{1,2}, u_1$ contains only the two vertices v and u_1 of M and there are no vertices in the interior of A' that are in M. Due to the 3-connectedness of G, there must be a path from u_3 to u' that avoids v and u_1. Therefore, there must be a path from u or $y_{1,2}$ to u' that avoids M (and contains neither u nor $y_{1,2}$ as an inner vertex). Note that u' does not have the same colour as u_2, since it cannot share a neighbour other than v with u_3. Indeed, the vertex would have to be $y_{1,2}$, but then u_1 and u_3 would have $y_{1,2}$ as a common neighbour apart from v.

Suppose $x_{1,2} \neq y_{1,2}$. Then every path from u_2 to a vertex in $N_4 \setminus \{u_1, u_2, u_3, u\}$ that does not have an inner vertex in M must contain u or $y_{1,2}$ as an inner vertex. The same holds for every path from $x_{1,2}$ to a vertex in $N_4 \setminus \{u_1, u_2, u_3, u\}$. However, this condition can be detected by the k-dimensional Weisfeiler-Leman algorithm and, as described in the previous paragraph, at least one of u and $y_{1,2}$ does not satisfy it. This implies that u and u_2 do not have the same colour or that $x_{1,2}$ and $y_{1,2}$ do not have the same colour, contradicting our construction. We conclude that $x_{1,2} = y_{1,2}$.

By Claim 1, the vertices u and u' have a common neighbour other than v. If this neighbour were not $x_{1,2}$, then u could not have the same colour as u_2, since there would be a path from u to $u' \in N_4 \setminus \{u_1, u_2, u_3, u\}$ none of whose inner vertices would be contained in $M \cup \{x_{1,2}\}$, whereas there would be no such path from u to any vertex in $N_4 \setminus \{u_1, u_2, u_3, u\}$. Figure 8.2b depicts this situation. We conclude that u' is a neighbour of $y_{1,2} = x_{1,2}$. Then, however, the pair $\{x_{1,2}, v\}$ separates u_1 from u, since by Claim 1, neither u' nor u_2 has a neighbour both inside and outside the cycle $v, u', x_{1,2}, u_2$. Therefore, $\{x_{1,2}, v\}$ is a 2-separator, contradicting the 3-connectedness of G.

Up to this point, regarding our efforts to prove the claim, we have shown that u_2 is a singleton in $\chi_{G_{(u_1,u_3)}}$. If $x_{1,2}$ were not adjacent to v, or $x_{2,3}$ were not adjacent to v, then $\chi_{G_{(u_1,u_3)}}$ would have three singletons lying on the same face. So assume otherwise. We can also assume that neither $x_{1,2}$ nor $x_{2,3}$ is a singleton. But this cannot happen, because the copies rendering $x_{1,2}$ and $x_{2,3}$ non-singletons (i.e. the other, necessarily existing vertices that have the same colours as $x_{1,2}$ and $x_{2,3}$, respectively) should also be non-equal and adjacent to u_2, which would force u_2 to have degree at least 5. This proves the claim. ⌟

Since $G[N_4]$ is empty, every common neighbour of u_i and u_{i+2} other than v must be equal to every common neighbour of u_{i+1} and u_{i+3} other than v. This means that there is a vertex v' other than v adjacent to all vertices of N_4.

Consider now the area bounded by the cycle v, u_i, v', u_{i+1} which does not contain u_{i+2}. If both u_i and u_{i+1} have two neighbours inside this area, then they do not have any neighbours outside the area, making $\{v, v'\}$ a separator. If u_i only has one neighbour inside the area, then this neighbour coincides with $x_{i,i+1}$ (since v' cannot be $x_{i,i+1}$ in this case, because v, u_i, v', u_{i+1} would not be a facial cycle, even after removing the possible diagonal $\{v, v'\}$) and must therefore be adjacent to u_{i+1}. We conclude that $v, u_i, x_{i,i+1}, u_{i+1}$ forms a facial cycle or becomes a facial cycle after removing the diagonal $\{x_{i,i+1}, v\}$. A symmetric argument can be applied with respect to v' in place of v. It follows that, inside the cycle v, u_i, v', u_{i+1}, there is at most one vertex, namely $x_{i,i+1}$. However, we already ruled out vertices of degree 3 at the beginning of the proof, so $x_{i,i+1}$ must be adjacent to all vertices of the cycle and thus has degree 4. This cannot be, since it would then lie in N_4. We conclude that u_i does not have a neighbour inside the cycle v, u_i, v', u_{i+1}. A similar observation holds for the cycle v, u_{i-1}, v', u_i. Still, due to the 3-connectedness of G, the vertex u_i must have some neighbours within the area bounded by the cycle v, u_i, v', u_{i+1} or within the area bounded by the cycle v, u_{i-1}, v', u_i, yielding the final contradiction. □

Putting the ingredients together, we obtain our classification of the exceptions.

Proof of Theorem 8.4: Recalling that every 3-connected planar graph has a vertex of degree 3, 4, or 5, the proof follows immediately by combining Lemmas 8.9, 8.10, and 8.11. □

We can finally prove the main theorem of this chapter.

Theorem 8.12. *The Weisfeiler-Leman dimension of the class of planar graphs is at most 3.*
(Equivalently, every planar graph is definable by a sentence in first-order logic with counting that uses only 4 variables.)

Proof: Let $k \in \mathbb{N}_{\geq 3}$. By Corollary 8.6, the k-dimensional Weisfeiler-Leman algorithm determines orbits on the class of all edge-coloured 3-connected planar graphs and identifies them. Thus, by Lemma 7.38, since the class of all planar graphs is minor-closed, its Weisfeiler-Leman dimension is at most 3. □

8.3 Discussion

In this chapter, we showed that the Weisfeiler-Leman dimension of the class of planar graphs is either 2 or 3 (where the lower bound follows from considering the hexagon). Thus, planar graphs form one of the rare families of graphs for which the known bounds on the Weisfeiler-Leman dimension are tight up to an additive constant of 1.

Still, the precise dimension remains open. The usual approaches for showing lower bounds on the Weisfeiler-Leman dimension rely on a construction due to Cai, Fürer, and Immerman, which yields “hard” graphs (cf. Section 6.4 and [30]). Towards improving the lower bound to 3, one would have to prove the existence of at least one planar graph which the 2-dimensional Weisfeiler-Leman algorithm fails to identify. However, the Cai, Fürer, Immerman construction will usually yield non-planar graphs.

Suspecting the exact dimension might be 2, to improve on the upper bound, one has to show that already the 2-dimensional Weisfeiler-Leman algorithm distinguishes every planar graph from every other graph. As stated in Theorem 7.37, for the reduction to edge-coloured 3-connected graphs, the 2-dimensional Weisfeiler-Leman algorithm suffices. Thus, to show that the Weisfeiler-Leman dimension of the class of planar graphs is 2, following the strategy outlined in this chapter, it would suffice to prove that the 2-dimensional Weisfeiler-Leman algorithm determines vertex orbits on the class of all edge-coloured 3-connected planar graph and identifies every such graph. However, our proof for the determination of vertex orbits in edge-coloured 3-connected planar graphs seems to require at least dimension 3. Hence, this part remains open for future work.

In the next chapter, we generalise the results from this chapter to a parametrisation of the Weisfeiler-Leman dimension by the Euler genus of the input graph.

9 The WL Dimension of Graphs of Euler Genus g

In the proof of Theorem 8.12, we used the results from Chapter 7 to show that the Weisfeiler-Leman dimension of the class of planar graphs is at most 3. In this chapter, we establish bounds for graphs that can be embedded into an arbitrary surface, for example, a torus or a projective plane. By the classification theorem for surfaces (see [130, Theorem 3.1.3]), up to homeomorphism (that is, topological equivalence) all surfaces fall into only two countably infinite families, the family $(\boldsymbol{S}_k)_{k\geq 0}$ of orientable surfaces and the family $(\boldsymbol{N}_\ell)_{\ell\geq 1}$ of non-orientable surfaces. For example, the sphere $\boldsymbol{S}_0$, the torus $\boldsymbol{S}_1$, and the double torus $\boldsymbol{S}_2$ are the first three orientable surfaces, and the projective plane $\boldsymbol{N}_1$ and the Klein bottle $\boldsymbol{N}_2$ are the first two non-orientable surfaces. The *Euler genus* $\mathrm{eg}(\boldsymbol{S})$ of a surface $\boldsymbol{S}$ is $2k$ if $\boldsymbol{S}$ is homeomorphic to the orientable surface $\boldsymbol{S}_k$, and ℓ if $\boldsymbol{S}$ is homeomorphic to the non-orientable surface $\boldsymbol{N}_\ell$. The *Euler genus* of a graph G is the least g such that G is embeddable (that is, can be drawn without edge crossings) in a surface of Euler genus g. As an example, the embedding visualised in Figure 9.1 shows that the Euler genus of the complete graph on 5 vertices is bounded by 2.

Our main result in this chapter is the linear bound of $4g+3$ on the Weisfeiler-Leman dimension of graphs of Euler genus g (see Theorem 9.24). For graphs embeddable in orientable surfaces, we can improve the bound further to $2g+3$ (see Corollary 9.57).

It was first proved in [61] that the Weisfeiler-Leman dimension of graphs of bounded genus is bounded. A more detailed proof of the same result can be found in the

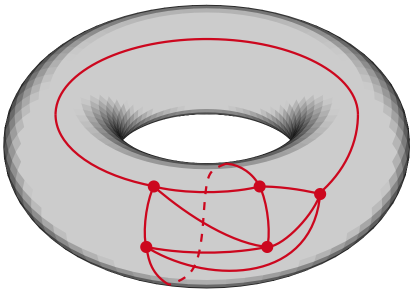


Figure 9.1: An embedding of K_5 into a torus.

journal paper [62]. Still, neither of the two papers gives an explicit bound on the Weisfeiler-Leman dimension. The proof of [62] only yields a bound that is quadratic in the Euler genus. It seems that the proof of [61] gives a linear bound, albeit with a large constant factor of at least 80 (not all details are worked out there, so it is difficult to determine the exact bound). The proof in both of these papers is based on the fact that sufficiently large graphs of minimum degree at least 3 embedded in a surface will have a facial cycle of length at most 6. Our approach is completely different. It exploits the connection of the Weisfeiler-Leman algorithm to the logics C^k and is based on a simplified version of a construction from [64, Chapter 15], applied there to graphs “almost embeddable” in a surface. We intend to remove a non-contractible cycle to reduce the genus and then applying induction. The problem with this attempt is that we cannot define non-contractible cycles in our counting logic fragments, only families of such cycles that may intersect in complicated patterns. Understanding these leads to significant technical complications, but in the end enables us to obtain a much better bound than the simpler proofs of [61] and [62].

The results presented in this chapter were published in [66].

9.1 Surface Topology and Graph Embeddings

In this section, we review the basic notions of surface topology and graph embeddings that are relevant for our purposes, some of which have already been mentioned in Chapter 2. In our presentation and notation, we follow [64, Chapter 9]. Many more details can be found there, in [130], and in [40, Appendix B].

We denote topological spaces like surfaces, curves, and embedded graphs by boldface letters. A *simple curve* in a topological space is a homeomorphic image of the real interval $[0, 1]$, equipped with the usual topology. Similarly, a *simple closed curve* is a homeomorphic image of the 1-sphere. A *closed disk* is a homeomorphic image of $\{x \in \mathbb{R}^2 \mid \|x\| \leq 1\}$ equipped with the usual topology, and an *open disk* is a topological space that is homeomorphic to $\mathbb{R}^2$, viewing the latter as a topological space. A topological space $\boldsymbol{X}$ is *arcwise connected* if for every two points $x, y \in \boldsymbol{X}$, there is a simple curve with endpoints x and y. For a subset $\boldsymbol{Y} \subseteq \boldsymbol{X}$, we define the *boundary* of $\boldsymbol{Y}$ in $\boldsymbol{X}$ to be the set $\boldsymbol{bd_X}(\boldsymbol{Y})$ of all points $x \in \boldsymbol{X}$ such that every neighbourhood of x has a non-empty intersection with both $\boldsymbol{Y}$ and $\boldsymbol{X} \setminus \boldsymbol{Y}$. The *interior* of $\boldsymbol{Y}$ is $\boldsymbol{int_X}(\boldsymbol{Y}) := \boldsymbol{Y} \setminus \boldsymbol{bd_X}(\boldsymbol{Y})$, and the *closure* of $\boldsymbol{Y}$ is $\boldsymbol{cl_X}(\boldsymbol{Y}) := \boldsymbol{Y} \cup \boldsymbol{bd_X}(\boldsymbol{Y})$. We omit the subscript $\boldsymbol{X}$ if the space is clear from the context.

A *surface* is an arcwise connected 2-manifold (intuitively, a space in which small neighbourhoods of points look like disks).[1] Recall from the introduction of this

[1] In this thesis, we only consider surfaces without boundary.

chapter that, up to homeomorphism, there are only two families $(\boldsymbol{S}_g)_{g\geq 0}$ and $(\boldsymbol{N}_g)_{g\geq 1}$ of surfaces. $\boldsymbol{S}_0$ is the 2-sphere, and for $g \in \mathbb{N}$, the surface $\boldsymbol{S}_g$ is obtained from the 2-sphere by adding g handles, and $\boldsymbol{N}_g$ is the surface obtained from the 2-sphere by adding g crosscaps. Intuitively, adding a *handle* to a surface means punching two holes into the surface and gluing a cylinder to these holes. Adding a *crosscap* means punching a hole into the surface and gluing a Möbius strip to this hole. The *Euler genus* $\operatorname{eg}(\boldsymbol{S})$ of a surface $\boldsymbol{S}$ is $2g$ if $\boldsymbol{S}$ is homeomorphic to $\boldsymbol{S}_g$, and g if $\boldsymbol{S}$ is homeomorphic to $\boldsymbol{N}_g$.

Let $\boldsymbol{g}$ be a simple closed curve in a surface $\boldsymbol{S}$. Then $\boldsymbol{g}$ is *contractible* if it is the boundary of a closed disk in $\boldsymbol{S}$, otherwise g is *non-contractible*. If $\boldsymbol{g}$ is non-contractible, we can obtain one or two surfaces of strictly smaller Euler genus by the following construction: we cut the surface along $\boldsymbol{g}$; what remains is a surface with one or two holes in it. Then we glue a disk onto each hole and obtain one or two simpler surfaces. For a more detailed description of this construction, see [40, Appendix B].

It will be important for us to distinguish between graphs in their standard combinatorial form—we refer to them as *abstract graphs*—and *embedded graphs.* Formally, an embedded graph in a surface $\boldsymbol{S}$ is a pair $G = \big(V(G), E(G)\big)$ where $V(G) \subseteq \boldsymbol{S}$ is a finite set and $E(G)$ is a set of simple curves in $\boldsymbol{S}$ such that for all $\boldsymbol{e} \in E(G)$, both endpoints and no internal point of $\boldsymbol{e}$ are in $V(G)$ and any two distinct $\boldsymbol{e}, \boldsymbol{e}' \in E(G)$ have at most one endpoint and no internal points in common. $\boldsymbol{G}$ denotes the point set $V(G) \cup \bigcup_{\boldsymbol{e}\in E(G)} \boldsymbol{e} \subseteq \boldsymbol{S}$. Sometimes, we also regard $\boldsymbol{G}$ as a topological (sub)space (of $\boldsymbol{S}$). The *underlying graph* of an embedded graph G is the graph with vertex set $V(G)$ and edge set $\{\boldsymbol{e} \cap V(G) \mid \boldsymbol{e} \in E(G)\}$. We usually blur the distinction between an embedded graph G and its underlying "abstract" graph. The *faces* of G are the arcwise connected components of the space $\boldsymbol{S} \setminus \boldsymbol{G}$. It is easy to see that for every face $\boldsymbol{f}$ of G, there is a subgraph $B \subseteq G$ such that the (topological) boundary $\boldsymbol{bd}(\boldsymbol{f})$ of $\boldsymbol{f}$ in $\boldsymbol{S}$ is precisely $\boldsymbol{B}$. We call B a *facial subgraph* of G.

We say that an (abstract) graph G is *embeddable* into a surface $\boldsymbol{S}$ if it is isomorphic to (the underlying graph of) a graph embedded in $\boldsymbol{S}$. The *Euler genus* $\operatorname{eg}(G)$ of a graph G is the smallest g such that G is embeddable into a surface of Euler genus g. It is useful to also define the *orientable genus* $\operatorname{og}(G)$ of a graph G to be the smallest g such that G is embeddable into $\boldsymbol{S}_g$ and the *non-orientable genus* $\operatorname{ng}(G)$ of G to be the smallest g such that G is embeddable into $\boldsymbol{N}_g$. Then $\operatorname{eg}(G) = \min\{2\operatorname{og}(G), \operatorname{ng}(G)\}$.

The graphs of Euler genus 0 are precisely the *planar graphs* because a graph can be embedded into the 2-sphere $\boldsymbol{S}_0$ if and only if it can be embedded into the real plane $\boldsymbol{R}^2$. The class of all graphs of Euler genus at most g is denoted by $\mathcal{E}_g$.

A *non-contractible cycle* in a graph G embedded in $\boldsymbol{S}$ is a cycle $C \subseteq G$ such that $\boldsymbol{C}$ is a non-contractible simple closed curve in $\boldsymbol{S}$.

We collect some useful topological facts for our purposes.

Fact 9.1. *Let $\boldsymbol{S}$ be a surface, and let $\boldsymbol{D}, \boldsymbol{D'} \subseteq \boldsymbol{S}$ be closed disks such that $\boldsymbol{bd}(\boldsymbol{D'}) \cap \boldsymbol{D}$ is a simple curve. Then $\boldsymbol{D} \cup \boldsymbol{D'}$ is a closed disk.*

Fact 9.2 (see Fact 9.1.14, [64])**.** *Let G be a graph embedded in a surface $\boldsymbol{S}$. Then either G contains a non-contractible cycle or there is a closed disk $\boldsymbol{D} \subseteq \boldsymbol{S}$ such that $\boldsymbol{G} \subseteq \boldsymbol{D}$.*

In the latter case, if G is 2-connected, the disk $\boldsymbol{D}$ can be chosen in such a way that there is a cycle $C \subseteq G$ such that $\boldsymbol{bd}(\boldsymbol{D}) = \boldsymbol{C}$.

Let $\boldsymbol{S}$ be a surface and let G be a graph embedded in $\boldsymbol{S}$. A set $\boldsymbol{X} \subseteq \boldsymbol{S}$ is *G-normal* if $\boldsymbol{X} \cap \boldsymbol{G} \subseteq V(G)$. The *representativity* $\rho(G)$ of G is the maximum $r \in \mathbb{N}_0$ such that every G-normal non-contractible simple closed curve $\boldsymbol{g}$ in $\boldsymbol{S}$ intersects $\boldsymbol{G}$ in at least r vertices. G is *polyhedrally embedded* in $\boldsymbol{S}$ if G is 3-connected and $\rho(G) \in \mathbb{N}_{\geq 3}$. In particular, every 3-connected plane graph (i.e., every graph embedded in $\boldsymbol{S}_0$) is polyhedrally embedded in $\boldsymbol{S}_0$. Polyhedrally embedded graphs have several useful properties (see [64, Fact 9.1.17]). In particular, all facial subgraphs of a polyhedrally embedded graph are chordless and non-separating cycles [146]. Conversely, for every graph embedded in a surface, all contractible, chordless, and non-separating cycles are facial subgraphs (see [64, Lemma 9.1.15]). (Here, a cycle $C \subseteq G$ is *non-separating* if $G - V(C)$ is connected.) This is a generalisation of the well-known theorem that the facial subgraphs of a 3-connected plane graph are precisely the chordless and non-separating cycles. It implies Whitney's Theorem [163] that all plane embeddings of 3-connected planar graphs have the same facial cycles and that, up to homeomorphism, a 3-connected planar graph has a unique embedding into the sphere $\boldsymbol{S}_0$.

9.2 Shortest Path Systems, Patches, and Necklaces

Here we introduce the graph-theoretic machinery necessary to prove our main theorem. Essentially, the definitions and results of this section are from [64, Chapter 15]. In fact, things are simpler here because [64, Chapter 15] deals with graphs *almost* embedded in a surface, whereas we only need to consider surface graphs. Sometimes, we need to change the definitions in order to improve the resulting bounds on the Weisfeiler-Leman dimension later. Notably, our *necklaces* play the role of the *belts* in [64], but the definition is slightly different. This also requires an adaptation of the proof that reducing necklaces exist.

Definition 9.3. Let G be a graph and $u, u' \in V(G)$. A *shortest path system (sps) from u to u'* is a family $\mathcal{Q}$ of shortest paths in G from u to u' such that every shortest path from u to u' in the subgraph $\bigcup_{Q \in \mathcal{Q}} Q$ is contained in $\mathcal{Q}$.

We also let $V(\mathcal{Q}) := \bigcup_{Q \in \mathcal{Q}} V(Q)$ and $E(\mathcal{Q}) := \bigcup_{Q \in \mathcal{Q}} E(Q)$. Furthermore, $G(\mathcal{Q}) := \big(V(\mathcal{Q}), E(\mathcal{Q})\big) = \bigcup_{Q \in \mathcal{Q}} Q$. We call $\mathcal{Q}$ *trivial* if $|V(\mathcal{Q})| \leq 2$, that is, if $G(Q)$ consists of a single vertex or a single edge.

The *height* $\mathrm{ht}^{\mathcal{Q}}(v)$ of $v \in V(\mathcal{Q})$ is the distance from u to v. The vertices in $\bigcap_{Q\in\mathcal{Q}} V(Q)$ are the *articulation vertices* of $\mathcal{Q}$. An articulation vertex v is *proper* if $v \neq u$ and $v \neq u'$. We denote the set of all articulation vertices of $\mathcal{Q}$ by $\mathrm{art}(\mathcal{Q})$.

For all u, $u' \in V(G)$ such that there is a path from u to u' in G, the *canonical sps from u to u' in G* is the set $\mathcal{Q}^G(u,u')$ of all shortest paths from u to u' in G.

For a path Q and vertices $u, v \in V(Q)$, we denote by uQv the segment of Q from u to v. With every sps $\mathcal{Q}$ from u to u', we can associate a partial order $\trianglelefteq^{\mathcal{Q}}$ on $V(\mathcal{Q})$ by letting $v \trianglelefteq^{\mathcal{Q}} w$ if v appears before w on some path $Q \in \mathcal{Q}$. For $v \trianglelefteq^{\mathcal{Q}} w$, we define the *segment* $\mathcal{Q}[v,w]$ to be the set of segments vQw from v to w of all paths $Q \in \mathcal{Q}$ that contain both v and w. Observe that $\mathcal{Q}[v,w]$ is an sps from v to w.

Lemma 9.4 ([64], Lemma 15.2.3)**.** *Let $\mathcal{Q}$ be an sps. Then $\mathcal{Q}$ is non-trivial and has no proper articulation vertices if and only if the graph $G(\mathcal{Q})$ is 2-connected.*

Lemma 9.5 ([64], Lemma 15.2.4)**.** *Let $\mathcal{Q}$ be a non-trivial sps that has no proper articulation vertices. Then there are internally disjoint paths $Q, Q' \in \mathcal{Q}$.*

While shortest paths systems are defined with respect to abstract graphs, the following notions are defined with respect to embedded graphs. For the rest of the section, we fix an embedding of the considered graph G.

Assumption 9.6. *G is a graph polyhedrally embedded in a surface $\boldsymbol{S}$ of Euler genus $g \in \mathbb{N}$.*

Definition 9.7. A *patch* in G is an sps $\mathcal{Q}$ in G such that:

- $\mathcal{Q}$ has no proper articulation vertices, and
- there is a closed disk $\boldsymbol{D} \subseteq \boldsymbol{S}$ such that $\boldsymbol{G(\mathcal{Q})} \subseteq \boldsymbol{D}$.

Fact 9.2 and Lemma 9.4 imply that, if $\mathcal{Q}$ is a non-trivial patch (i.e. a patch that does not consist in a single vertex or a single edge), then there is a unique disk $\boldsymbol{D}(\mathcal{Q})$ such that $\boldsymbol{G(\mathcal{Q})} \subseteq \boldsymbol{D}(\mathcal{Q})$ and there is a cycle $C(\mathcal{Q}) \subseteq G(\mathcal{Q})$ such that $\boldsymbol{bd}\big(\boldsymbol{D}(\mathcal{Q})\big) = \boldsymbol{C}(\mathcal{Q})$. Furthermore, there are two paths $Q, Q' \in \mathcal{Q}$ such that $C(\mathcal{Q}) = Q \cup Q'$.

Definition 9.8. A subgraph $H \subseteq G$ is *simplifying* if every connected component of $G - H$ is contained in $\mathcal{E}_{g-1}$. Otherwise, H is non-simplifying.

A patch $\mathcal{Q}$ is *simplifying* if the graph $G(\mathcal{Q})$ is simplifying.

Lemma 9.9 ([64], Corollary 15.3.5)**.** *For every non-simplifying subgraph $H \subseteq G$, there is exactly one connected component A^* of $G - H$ with $A^* \notin \mathcal{E}_{g-1}$, and all other connected components are planar.*

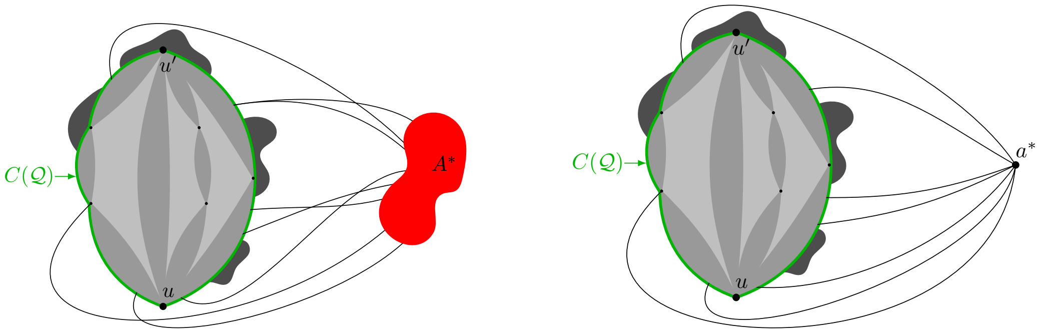


Figure 9.2: Left: A patch $\mathcal{Q}$ with non-planar component A^* and boundary cycle $C(\mathcal{Q})$. The curve $\boldsymbol{C}(\mathcal{Q})$ is the boundary of the disk $\boldsymbol{D}(\mathcal{Q})$, which consists of the light gray and medium gray areas. Right: the (planar) factor graph $G(\mathcal{Q})/A^*$.

It turns out that non-simplifying patches form the basic building blocks of our theory. Let $\mathcal{Q}$ be a non-trivial non-simplifying path. Let A^* be the unique connected component of $G - V(\mathcal{Q})$ that is not planar, whose existence and uniqueness of A^* follow from Lemma 9.9. Recall that G/A^* is the graph obtained from G by contracting the subgraph A^* to a single vertex a^*. By [64, Corollary 15.4.5], G/A^* is a 3-connected planar graph. Figure 9.2 displays a schematic view of a patch $\mathcal{Q}$ with some attached (planar) connected components as well as the non-planar component A^*, the disk $\boldsymbol{D}(\mathcal{Q})$, and the boundary cycle $\boldsymbol{C}(\mathcal{Q})$.

We define the *internal graph* of a non-trivial non-simplifying patch $\mathcal{Q}$ to be the graph $I := I(\mathcal{Q})$ with vertex set $V(I) := V(G) \cap \boldsymbol{D}(\mathcal{Q})$ and edge set $E(I) := \{\boldsymbol{e} \in E(G) \mid \boldsymbol{e} \subseteq \boldsymbol{D}(\mathcal{Q})\}$. Note that $C(\mathcal{Q}) \subseteq I$. Formally, the definitions of the graphs $C(\mathcal{Q})$ and $I(\mathcal{Q})$ do not only depend on the abstract graph G and the sps $\mathcal{Q}$, but on the embedding of G in $\boldsymbol{S}$. However, it can be proved that, actually, the graphs are invariant under embeddings.

Lemma 9.10 ([64])**.** *Let $\mathcal{Q}$ be a non-trivial non-simplifying patch in G. Let G' be a graph embedded in a surface $\boldsymbol{S}'$ of Euler genus g such that G and G' are isomorphic (as abstract graphs), and let f be an isomorphism from G to G'. Then $\mathcal{Q}' := f(\mathcal{Q}')$ is a non-simplifying patch in G', and it holds that $f\big(C(\mathcal{Q})\big) = C(\mathcal{Q}')$ and $f\big(I(\mathcal{Q})\big) = I(\mathcal{Q}')$.*

This follows from [64, Lemma 15.4.10]. Intuitively, the reason why this holds is that the 3-connected planar graph G/A^* has a unique embedding.

Corollary 9.11. *Let $u, u' \in V(G)$ and $\mathcal{Q} := \mathcal{Q}^G(u, u')$ such that $\mathcal{Q}$ is a non-trivial non-simplifying patch. Let f be an automorphism of G such that $f(u) = u$ and $f(u') = u'$. Then $f\big(C(\mathcal{Q})\big) = C(\mathcal{Q})$ and $f\big(I(\mathcal{Q})\big) = I(\mathcal{Q})$.*

We remark that the analogue of Corollary 9.11 for simplifying patches does not hold. (Figure 9.4 in Section 9.4.2 shows an example.) The analysis of simplifying

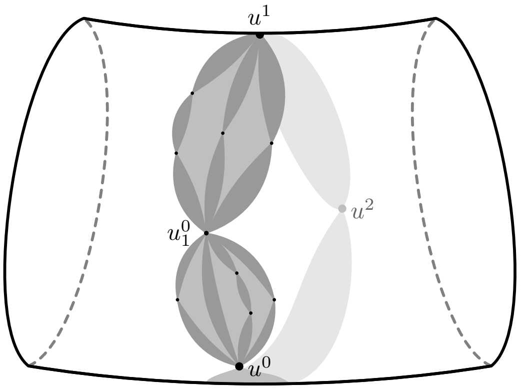


Figure 9.3: A reducing necklace on a torus section.

patches is much more involved, and we defer it to Section 9.4.2.

The final objects we define in this section are *necklaces.*

Definition 9.12. A *necklace* in the graph G is a tuple $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$, where $u^0, u^1, u^2 \in V(G)$ and $\mathcal{Q}^i := \mathcal{Q}^G(u^i, u^{i+1})$ (indices taken modulo 3) is the canonical sps from u^i to u^{i+1}, such that the following conditions are satisfied for every $i \in \{0, 1, 2\}$.

1. u^0, u^1, u^2 are pairwise distinct.
2. $V(\mathcal{Q}^i) \cap V(\mathcal{Q}^{i+1}) = \{u^{i+1}\}$ (indices modulo 3).
3. There is a disk $\boldsymbol{D}^i \subseteq \boldsymbol{S}$ such that $\boldsymbol{G(\mathcal{Q}^i)} \subseteq \boldsymbol{D}^i$.

For a necklace $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$, we write $V(\mathcal{B})$ for $\bigcup_{i=0}^2 \bigcup_{Q \in \mathcal{Q}^i} V(Q)$ and $E(\mathcal{B})$ for $\bigcup_{i=0}^2 \bigcup_{Q \in \mathcal{Q}^i} E(Q)$. We let $G(\mathcal{B}) := \big(V(\mathcal{B}), E(\mathcal{B})\big)$. Moreover, we define the set of *articulation vertices* of $\mathcal{B}$ to be $\operatorname{art}(\mathcal{B}) := \bigcup_{i=0}^2 \operatorname{art}(\mathcal{Q}^i)$.

Definition 9.13. A necklace $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$ is *reducing* if, for $i \in \{0, 1, 2\}$, there is a path $Q^i \in \mathcal{Q}^i$ such that $B := Q^0 \cup Q^1 \cup Q^2$ is a non-contractible cycle.

We can think of a reducing necklace as a necklace around a handle of our surface (or a crosscap in the non-orientable case). The beads of the necklace are the disks of the patches that form the necklace. Figure 9.3 shows a reducing necklace on a torus with articulation vertices u^0, u^0_1, u^1, u^2.

Lemma 9.14 (Necklace Lemma)**.** *G has a reducing necklace.*

Essentially, this is [64, Lemma 15.5.8], with the necklaces corresponding to the *belts* there. But since, apart from a renaming, we have also slightly changed the content of the definition of a necklace/belt, the proof needs to be adapted, too. For the proof of the lemma, we need one well-known fact and a more complicated lemma from [64].

Fact 9.15. *Let $\boldsymbol{S}$ be a surface, and let $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3 \subseteq \boldsymbol{S}$ be simple curves with the same endpoints and mutually disjoint interiors. Then $\boldsymbol{g}_1 \cup \boldsymbol{g}_2$, $\boldsymbol{g}_2 \cup \boldsymbol{g}_3$, and $\boldsymbol{g}_1 \cup \boldsymbol{g}_3$ are simple closed curves, and if $\boldsymbol{g}_1 \cup \boldsymbol{g}_2$ and $\boldsymbol{g}_2 \cup \boldsymbol{g}_3$ are contractible, then $\boldsymbol{g}_1 \cup \boldsymbol{g}_3$ is contractible as well.*

For a proof, see [130, Proposition 4.3.1].

Lemma 9.16 ([64], Lemma 15.5.9)**.** *Let $\mathcal{Q}$ be an sps in G such that there is no disk $\boldsymbol{D} \subseteq \boldsymbol{S}$ with $\boldsymbol{G(\mathcal{Q})} \subseteq \boldsymbol{D}$, but for every proper segment $\mathcal{Q}'$ of $\mathcal{Q}$, there is a disk $\boldsymbol{D'} \subseteq \boldsymbol{S}$ with $\boldsymbol{G(\mathcal{Q}')} \subseteq \boldsymbol{D'}$. Then there are internally disjoint paths $Q, Q' \in \mathcal{Q}$ such that $\boldsymbol{Q} \cup \boldsymbol{Q'}$ is a non-contractible simple closed curve in $\boldsymbol{S}$.*

With these tools at hand, we can now prove the existence of a reducing necklace in G. Recall that for a graph G, $\|G\|$ denotes $|E(G)|$.

Proof of Lemma 9.14: By Fact 9.2, there is a cycle $C \subseteq G$ such that $\boldsymbol{C}$ is a non-contractible simple closed curve in $\boldsymbol{S}$. We choose such a cycle C of minimum length. We let $u^0, u^1, u^2 \in V(C)$ such that

$$\left\lfloor \frac{\|C\|}{3} \right\rfloor \leq \operatorname{dist}^C(u^i, u^j) \leq \left\lceil \frac{\|C\|}{3} \right\rceil . \tag{9.1}$$

Also we set $\mathcal{Q}^i := \mathcal{Q}^G(u^i, u^{i+1})$ and $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$. Here and throughout the proof, indices i, j are taken from $\mathbb{Z}_3$ with addition modulo 3.

It follows from (9.1) that the u^i are mutually distinct. Thus, $\mathcal{B}$ satisfies Condition 1 of Definition 9.12.

Let Q^i be the segment of C from u^i to u^{i+1} that does not contain u^{i+2}. Then $C = Q^0 \cup Q^1 \cup Q^2$.

Claim 1. Let $P \subseteq G$ be a shortest path with distinct endvertices $u, u' \in V(C)$ and no internal vertices in C. Let Q, Q' be the two segments of C from u to u'. Then $P \cup Q$ or $P \cup Q'$ is a non-contractible cycle. Furthermore, if $P \cup Q$ is a non-contractible cycle, then $\|Q'\| = \|P\|$, and if $P \cup Q'$ is a non-contractible cycle, then $\|Q\| = \|P\|$.

Proof of the claim: Clearly, since P has no internal vertices in C, both $P \cup Q$ and $P \cup Q'$ are cycles. By Fact 9.15, we know that $P \cup Q$ or $P \cup Q'$ is non-contractible. By symmetry, we can assume that $P \cup Q$ is non-contractible. Since C is a shortest non-contractible cycle, we have $\|P \cup Q\| \geq \|C\| = \|Q \cup Q'\|$. Thus $\|Q'\| \leq \|P\|$, and, since P is a shortest path, equality holds. ⌟

Claim 2. Let $Q \in \mathcal{Q}^i$. Then $V(Q) \cap V(C) \subseteq V(Q^i)$.

Proof of the claim: By symmetry, it suffices to prove the claim for $i = 0$. Suppose, towards a contradiction, that there is a path $Q \in \mathcal{Q}^0$ with $V(Q) \cap V(C) \nsubseteq V(Q^0)$. Fix Q to be such a path with the maximum number of edges in $E(C)$.

Note that $Q \neq Q^0$. Thus, $Q \nsubseteq C$, because the only paths in C from u^0 to u^1 are Q^0 and $Q^1 \cup Q^2$. However, $Q \neq Q^0$ and $\|Q\| \leq \|Q^0\| < \|Q^1\| + \|Q^2\| = \|Q^1 \cup Q^2\|$, which implies $Q \neq Q^1 \cup Q^2$.

Recall that for a path Q and vertices $v, w \in V(Q)$, we denote by vQw the segment of Q from v to w. Throughout this proof, for a second path Q' with $v, w \in V(Q')$, we denote by $uQvQ'w$ the walk from u to w obtained by following Q from u to v and then following Q' from v to w. (We also use this notation style to compose multiple segments of paths.)

Let $u \neq u'$ be vertices in $V(C)$ and let $P = uQu'$ be a segment of Q with endvertices $u, u' \in V(C)$ and all internal vertices and edges not in C. Then P is a shortest path from u to u'. Let R, R' be the two segments of C with endpoints u, u'. Then by Fact 9.15, one of $R \cup P$ and $R' \cup P$ must be a non-contractible cycle, say $R' \cup P$. Then $\|P\| = \|R\|$.

(Case 1: Q has an empty intersection with the interior of R.) Then the segment $u^0QuRu'Qu^1$ is a path from u^0 to u^1 that has the same length as Q, but more edges in $E(C)$. This contradicts the maximality of Q.

(Case 2: The segment u^0Qu contains an internal vertex that lies in R.) Let v be the first vertex of Q in R. Then v appears on Q before u. Let w be the last vertex of Q in R (possibly, $w = u'$). Then $u^0QvRwQu^1$ is a path from u^0 to u^1 that is shorter than Q, which contradicts Q being a shortest path.

(Case 3: The segment $u'Qu^1$ contains an internal vertex that lies in R.) Let the vertex w be the last one of Q in R. Then w appears on Q after u'. Let v be the first vertex of Q in R (possibly, $v = u$). Then $u^0QvRwQu^1$ is a path from u^0 to u^1 that is shorter than Q, which again contradicts Q being a shortest path.

Thus, the segment P does not exist, which implies the claim. ⌟

Claim 3. $Q^i \in \mathcal{Q}^i$.

Proof of the claim: Again, by symmetry, it suffices to prove the claim for $i = 0$. Let $Q \in \mathcal{Q}^0$ with a maximum number of edges in $E(C)$. Arguing with similar techniques as in the proof of Claim 2, we can show that $Q = Q^0$. ⌟

Claim 4. Let $Q \in \mathcal{Q}^i$ and $Q' \in \mathcal{Q}^{i+1}$. Then $V(Q) \cap V(Q') = \{u^{i+1}\}$.

Proof of the claim: As usual, we assume $i = 0$.

Suppose that $v_0 = u^0, v_1, v_2, \ldots, v_m = u^1$ are the vertices in $Q \cap Q^0$ in the order in which they appear on Q^0. Then the vertices appear on Q in the same order, because, by Claim 3, both Q^0 and Q are shortest paths. For $i \in [0, m-1]$, let P_i be the segment of Q from v_i to v_{i+1}. Note that no internal vertex of P_i is in $V(C)$.

Thus, either $P_i = v_i Q^0 v_{i+1}$ is a single edge or $P_i \cup v_i Q^0 v_{i+1}$ is a cycle. Since this cycle is shorter than C, it must be contractible. Let $C_0 := C$, and for $i \in [m-1]$, let C_i be the cycle obtained from C_{i-1} by replacing the segment $v_i Q^0 v_{i+1}$ with P_i. It follows from Claim 1 applied to the cycle C_{i-1} and the path P_i that each C_i is non-contractible. In particular,

$$C' := C_{m-1} = u^0 Q u^1 Q^1 u^2 Q^2 u^0$$

is a non-contractible cycle of the same length as C.

Thus, C' is also a shortest non-contractible cycle through the vertices u^0, u^1, u^2 and $\mathrm{dist}^{C'}(u^i, u^j) = \mathrm{dist}^{C}(u^i, u^j)$. This means that we can apply all previous claims to C' instead of C. In particular, it follows from Claim 2 applied to C' and Q' that $V(Q') \cap V(Q) = \{u^1\}$. ⌟

Claim 5. Let $Q, Q' \subseteq G$ be paths from u^i to u^{i+1} such that $\|Q\|, \|Q'\| \leq \|Q^i\|$. Then there is a disk $\boldsymbol{D} \subseteq \boldsymbol{S}$ such that $\boldsymbol{Q} \cup \boldsymbol{Q'} \subseteq \boldsymbol{int}(\boldsymbol{D})$.

Proof of the claim: We have $\|Q \cup Q'\| \leq \|Q\| + \|Q'\| \leq 2\|Q^i\| < \|C\|$. Thus, the graph $Q \cup Q'$ does not contain a non-contractible cycle, and, by Fact 9.2, there is a closed disk $\boldsymbol{D'} \subseteq \boldsymbol{S}$ such that $\boldsymbol{Q}, \boldsymbol{Q'} \subseteq \boldsymbol{D'}$. We can slightly increase $\boldsymbol{D'}$ to obtain a disk $\boldsymbol{D}$ such that $\boldsymbol{Q}, \boldsymbol{Q'} \subseteq \boldsymbol{int}(\boldsymbol{D})$. ⌟

Claim 6. There is a disk $\boldsymbol{D} \subseteq \boldsymbol{S}$ such that $\boldsymbol{G(\mathcal{Q}^i)} \subseteq \boldsymbol{D}$.

Proof of the claim: Suppose, towards a contradiction, that there is no such disk. Let $\mathcal{Q}$ be a segment of $\mathcal{Q}^i$ such that there is no disk $\boldsymbol{D} \subseteq \boldsymbol{S}$ with $\boldsymbol{G(\mathcal{Q})} \subseteq \boldsymbol{D}$, but for every proper segment $\mathcal{Q}'$ of $\mathcal{Q}$, there is a disk $\boldsymbol{D'} \subseteq \boldsymbol{S}$ with $\boldsymbol{G(\mathcal{Q}')} \subseteq \boldsymbol{D'}$. Then, by Lemma 9.16, there are paths $Q, Q' \in \mathcal{Q}$ such that $\boldsymbol{Q} \cup \boldsymbol{Q'}$ is a non-contractible simple closed curve in $\boldsymbol{S}$. This contradicts Claim 5. ⌟

We have already noted that $\mathcal{B}$ satisfies Condition 1 of Definition 9.12. It follows from Claim 4 that it satisfies Condition 2, and Claim 6 implies that it satisfies Condition 3 as well. Thus, $\mathcal{B}$ is a necklace. Claim 3 implies that this necklace is reducing. □

9.3 Technical Ingredients

In this chapter, we use the characterisation of the Weisfeiler-Leman dimension of a graph via the number of variables needed to define the graph in first-order logic with counting quantifiers. To obtain a good bound, we need to reuse variables as excessively as possible. To this end, recall the definition of the *width* of a C-formula from Section 2.4: a formula $\varphi \in \mathsf{C}$ has width k if every subformula of φ has at most k free variables, and we denote the C-formulae of width k by $\mathsf{C}^k_{\mathsf{w}}$.

In Chapter 8, we showed that the Weisfeiler-Leman dimension of a planar graph is at most 3. Using Corollary 3.21, we can rephrase this as follows.

Theorem 9.17. *For every coloured planar graph G, there is a C^4-sentence iso_G that identifies G.*

We introduce a few technical results that constitute building blocks of the formulae that we construct throughout this chapter.

The following lemma bounds the number of variables needed for avoiding definable subsets.

Lemma 9.18. *Let $\varphi(x_1, \dots, x_k, y) \in \mathsf{C}^{\ell}_{\mathsf{w}}$. Then there is a $\mathsf{C}^{\max\{k+3,\ell\}}_{\mathsf{w}}$-formula $\mathsf{comp}_{\varphi}(x_1, \dots, x_k, y, y')$ such that for all graphs G of order $|G| \leq n$ and all vertices $u_1, \dots, u_k, v, v' \in V(G)$,*

$$G \models \mathsf{comp}_{\varphi}(u_1, \dots, u_k, v, v') \iff \text{v and v' belong to the same connected component of } G - \varphi[G, u_1, \dots, u_k, y].$$

Proof: Without loss of generality, we assume that in all formulae of the form $\exists^{\geq p} z \vartheta$ that we consider, the variable z occurs free in ϑ. We let $\psi(x_1, \dots, x_k, y, y')$ be the formula obtained from the formula $\mathsf{dist}_{\leq n-1}(y, y')$ of Example 2.4 by replacing each subformula $\exists^{\geq p} z \vartheta$ with $\exists^{\geq p} z(\neg\varphi(x_1, \dots, x_k, z) \wedge \vartheta)$. Then, letting $U := \varphi[G, u_1, \dots, u_k, y]$, for all $v, v' \in V(G) \setminus U$, we have $G \models \psi(u_1, \dots, u_k, v, v')$ if and only if v and v' belong to the same connected component of $G - U$. Note that $\psi(x_1, \dots, x_k, y, y') \in \mathsf{C}^{\max\{k+3,\ell\}}_{\mathsf{w}}$, because the formula $\vartheta \in \mathsf{C}^3_{\mathsf{w}}$ has at most two free variables besides z.

Now we can set

$$\mathsf{comp}_{\varphi}(x_1, \dots, x_k, y, y') := \neg\varphi(x_1, \dots, x_k, y) \wedge \neg\varphi(x_1, \dots, x_k, y') \wedge \psi(x_1, \dots, x_k, y, y'). \qquad \square$$

Lemma 9.19. *Let $n \geq 1$, $\ell \geq 3$, and $k \in [\ell]$. Then for every $\mathsf{C}^{\ell}_{\mathsf{w}}$-formula $\psi(x_1, \dots, x_k)$, there is a $\mathsf{C}^{\ell}_{\mathsf{w}}$-formula $\widehat{\psi}(x_1, \dots, x_k)$ such that for every graph G of order $|G| \leq n$, every connected component A of G, and all $u_1, \dots, u_k \in V(A)$, it holds that*

$$G \models \widehat{\psi}(u_1, \dots, u_k) \iff A \models \psi(u_1, \dots, u_k).$$

Proof: We construct $\widehat{\psi}$ by induction on ψ. If ψ is atomic, then we simply let $\widehat{\psi} := \psi$. If $\psi = \neg\varphi$, we let $\widehat{\psi} := \neg\widehat{\varphi}$, and if $\psi = \varphi_1 \vee \varphi_2$, we let $\widehat{\psi} := \widehat{\varphi}_1 \vee \widehat{\varphi}_2$. The only interesting case is that $\psi(x_1, \dots, x_k) = \exists^{\geq p} y \varphi(x_1, \dots, x_k, y)$. Note that the variable y may be among $x_1, \dots, x_k$. If this is the case, $\varphi(x_1, \dots, x_k, y)$ is the same formula as $\varphi(x_1, \dots, x_k)$. Without loss of generality, we may assume that there is a $j \leq k$ such that $y \neq x_j$. This is obvious if $k \geq 2$. If $k = 1$, we can rename the bound variable y and choose $j = 1$. We let $\widehat{\psi}(x_1, \dots, x_k) = \exists^{\geq p} y\big(\mathsf{dist}_{\leq n-1}(x_j, y) \wedge \widehat{\psi}(x_1, \dots, x_k, y)\big)$, where $\mathsf{dist}_{\leq n-1}$ is the $\mathsf{C}^3_{\mathsf{w}}$-formula defined in Example 2.4. $\square$

Recall that the notation $\psi(x_1, \dots, x_k)$ merely says that the free variables of the formula ψ are *among* $x_1, \dots, x_k$; not all of these variables actually have to appear. Thus we can also apply the lemma to sentences ψ and obtain the following corollary.

Corollary 9.20. *Let $n \geq 1$ and $\ell \geq 3$. Then for every $\mathsf{C}^{\ell}_{\mathsf{w}}$-sentence φ, there is a $\mathsf{C}^{\ell}_{\mathsf{w}}$-formula $\widehat{\varphi}(x)$ such that for every graph G of order $|G| \leq n$ and every $u \in V(G)$, we have $G \models \widehat{\varphi}(u)$ if and only if $A \models \varphi$ for the connected component A of u in G.*

Corollary 9.21. *Let $\ell \geq 3$, and let G be a graph such that every connected component of G is identified by a $\mathsf{C}^{\ell}_{\mathsf{w}}$-sentence. Then G is identified by a $\mathsf{C}^{\ell}_{\mathsf{w}}$-sentence.*

Observe that the corollary fails for $\ell = 2$. An example is the graph G that is the disjoint union of two triangles.

Lemma 9.22. *Let $k \geq 0$, $n \geq 1$, $\ell \geq 3$ and let $\psi \in \mathsf{C}^{\ell}_{\mathsf{w}}$ and $\varphi(x_1, \dots, x_k, y) \in \mathsf{C}^{m}_{\mathsf{w}}$. Then there is a formula $\widetilde{\psi}(x_1, \dots, x_k, y) \in \mathsf{C}^{\max\{k+\ell,m\}}_{\mathsf{w}}$ such that for all graphs G of order $|G| \leq n$ and all $u_1, \dots, u_k, v \in V(G)$, the following holds. Let $U :=\varphi[G, u_1, \dots, u_k, y]$, and let A_v be the connected component of v in $G - U$ (assuming $v \notin U$). Then*

$$G \models \widetilde{\psi}(u_1, \dots, u_k, v) \iff v \notin U \text{ and } A_v \models \psi.$$

Proof: Again, without loss of generality, we assume that in all formulae of the form $\exists^{\geq p} z \vartheta$ that we consider, the variable z occurs free in ϑ. We apply Corollary 9.20 to ψ and obtain a $\mathsf{C}^{\ell}_{\mathsf{w}}$-formula $\widehat{\psi}(y)$ such that for every graph H of order at most n and every $v \in V(H)$, we have $H \models \widehat{\psi}(v)$ if and only if $A_v \models \psi$, where A_v is the connected component of v in H. In particular, this holds for the graph $H := G - U$.

Without loss of generality, we may assume that the variables $x_1, \dots, x_k$ do not appear in $\widehat{\psi}(y)$. We let $\widehat{\psi}'(x_1, \dots, x_k, y)$ be the $\mathsf{C}^{\max\{k+\ell,m\}}_{\mathsf{w}}$-formula obtained from $\widehat{\psi}(y)$ by replacing each subformula $\exists^{\geq p} z \vartheta$ with $\exists^{\geq p} z(\neg\varphi(x_1, \dots, x_k, z) \wedge \vartheta)$. Then for all $v \in V(H)$, we have $G \models \widehat{\psi}'(v) \iff H \models \widehat{\psi}(v)$. We let

$$\widetilde{\psi}(x_1, \dots, x_k, y) := \neg\varphi(x_1, \dots, x_k, y) \wedge \widehat{\psi}'(x_1, \dots, x_k, y). \qquad \square$$

We need one more technical lemma which will be applied in one case of the proof of our main theorem in Section 9.4.1. The reason we put it here is that we do not want to interrupt the flow of the main argument later. The reader may safely skip the lemma on first reading the chapter and get back to it later.

For the purposes of the lemma, we need a way to prevent some free variables from counting towards the width of a formula. We shall use the symbol $\circ$ as a special placeholder that can be substituted for the free occurrences of variables with the effect that this placeholder does not count as a variable for the width. For example, for $\psi(x, z) := \exists y\big(E(x, y) \wedge E(y, z)\big) \in \mathsf{C}^{3}_{\mathsf{w}}$, we have $\psi(\circ, z) \in \mathsf{C}^{2}_{\mathsf{w}}$ and $\psi(\circ, \circ) \in \mathsf{C}^{1}_{\mathsf{w}}$. Recall that for a graph G and a subgraph $A \subseteq G$, by G/A we denote the graph obtained from G by identifying all vertices of A. Furthermore, we let a be the vertex of G/A corresponding to A.

Lemma 9.23. *Let $0 \leq \ell < m < k$ and $\xi(x_1, \ldots, x_\ell, y), \psi(x_1, \ldots, x_m, z) \in \mathsf{C}^k_\mathsf{w}$ such that $\psi(\circ, \ldots, \circ, x_{\ell+1}, \ldots, x_m, \circ) \in \mathsf{C}^{k-\ell}_\mathsf{w}$. Then there is a formula $\varphi(x_1, \ldots, x_m) \in \mathsf{C}^k_\mathsf{w}$ such that the following holds.*

Let G be a graph and let $A \subseteq G$. Suppose $u_1, \ldots, u_m \in V(G) \setminus V(A)$ such that $V(A) = \xi[G, u_1, \ldots, u_\ell, y]$. Then

$$G \models \varphi(u_1, \ldots, u_m) \iff G/A \models \psi(u_1, \ldots, u_m, a).$$

Proof: Without loss of generality, we assume that every bound variable in ψ does not occur free in ψ or ξ and is not bound by a second quantifier in ψ.

We let $\varphi := \psi^*$, where we define the transformation $*$ inductively to eliminate the variable z as follows.

For atoms α that do not mention z, we let $\alpha^* := \alpha$. Atoms with z are treated as follows, where x denotes a variable distinct from z. For equality atoms, we define $(z = z)^* := \text{TRUE}$ and $(x = z)^* := (z = x)^* := \text{FALSE}$. For atoms with predicate symbol E, we let $E(z, z)^* := \text{FALSE}$, and $E(x, z)^* := \exists z\big(\xi(x_1, \ldots, x_\ell, z) \wedge E(x, z)\big)$, and define $E(z, x)^*$ analogous to $E(x, z)^*$. For atoms with predicate symbol R_C, where the colour C is a multiset with r distinct elements $c_1, \ldots, c_r$ of multiplicities $p_1, \ldots, p_r$, we define $R_C(z, z)^* := \text{FALSE}$, and $R_C(x, z)^* := \bigwedge_{j=1}^r \exists^{=p_j} z\big(\xi(x_1, \ldots, x_\ell, z) \wedge R_{c_j}(x, z)\big)$, and $R_C(z, x)^*$ analogous to $R_C(x, z)^*$.

Inductively, for formulae ϑ, ϑ_1, ϑ_2, we further define $(\neg\vartheta)^* := \neg\vartheta^*$ and $(\vartheta_1 \vee \vartheta_2)^* := (\vartheta_1^* \vee \vartheta_2^*)$. For the case that $\exists^{\geq p} x \vartheta(x_1, \ldots, x_n, x, z)$ for $p \geq 2$, we define

$$\begin{aligned}(\exists^{\geq p} x \vartheta)^* := \Big(&\vartheta(x_1, \ldots, x_n, z, z)^* \wedge \exists^{\geq p-1} x\big(\neg\xi(x_1, \ldots, x_\ell, x) \wedge \vartheta(x_1, \ldots, x_n, x, z)^*\big)\Big) \\ &\vee \exists^{\geq p} x\big(\neg\xi(x_1, \ldots, x_\ell, x) \wedge \vartheta(x_1, \ldots, x_n, x, z)^*\big). \qquad (9.2)\end{aligned}$$

Note that the formula $\vartheta(x_1, \ldots, x_n, z, z)^*$ is obtained by first substituting z for x in ϑ and then applying $*$ to the resulting formula to eliminate z. The case $p = 1$ is dealt with analogously.

To prove the correctness of the construction, we need to show that the free variables of ψ^* are among $\{x_1, \ldots, x_m\}$ and $\psi^* \in \mathsf{C}^k_\mathsf{w}$, and that ψ^* has the correct meaning.

First, observe that a straightforward induction obtains that, for every formula ϑ,

$$\text{free}(\vartheta^*) \subseteq \big(\text{free}(\vartheta) \setminus \{z\}\big) \cup \{x_1, \ldots, x_\ell\}, \qquad (9.3)$$

where free(ϑ) denotes the free variables of ϑ. Thus, free(ψ^*) $\subseteq \{x_1, \ldots, x_m\}$.

Second, observe that the condition $\psi(\circ, \ldots, \circ, x_{\ell+1}, \ldots, x_m, \circ) \in \mathsf{C}^{k-\ell}_\mathsf{w}$ expresses that no subformula of ψ (including ψ itself) has more than $k - \ell$ free variables that are not contained in the set $\{x_1, \ldots, x_\ell, z\}$. So we can assume that all subformulae of ψ satisfy this condition.

Now we are ready to prove $\psi^* \in \mathsf{C}_{\mathsf{w}}^k$ by induction on ψ. For the base steps, note that $E(x,z)^*, E(z,x)^*, R_C(x,z)^*, R_C(z,x)^* \in \mathsf{C}_{\mathsf{w}}^{\ell+2}$ and $\ell + 2 \leq k$; the other base cases are trivial.

For the inductive step, the case $\neg\vartheta$ is trivial. For the case $\vartheta_1 \vee \vartheta_2$, we exploit the observation (9.3) and that ψ has at most $k - \ell$ free variables not in $\{x_1, \dots, x_\ell, z\}$. The case $\exists^{\geq p} x \vartheta(x_1, \dots, x_n, x, z)$ follows immediately by induction, since it holds that $\vartheta(x_1, \dots, x_n, z, z)^* \in \mathsf{C}_{\mathsf{w}}^k$ and $\vartheta(x_1, \dots, x_n, x, z)^* \in \mathsf{C}_{\mathsf{w}}^k$.

Finally, we show the following statement for every formula $\vartheta(x_1, \dots, x_n, z)$, where $n \geq \ell$ and every bound variable in ϑ does not occur bound in ϑ or ξ and is not bound by a second quantifier in ϑ: for every graph G, every subgraph $A \subseteq G$, and all $u_1, \dots, u_n \in V(G) \setminus V(A)$ such that $V(A) = \xi[G, u_1, \dots, u_\ell, y]$ we have

$$G \models \vartheta^*(u_1, \dots, u_n) \iff G/A \models \vartheta(u_1, \dots, u_n, a). \tag{9.4}$$

The proof is by induction on ϑ. This statement in particular applies to the formula $\psi(x_1, \dots, x_m, z)$ and thus, completes the proof of the lemma.

The base step for atomic formulae follows from the fact that $u_i \neq a$ for every $1 \leq i \leq n$ and the definition of G/A and its colouring.

In the inductive step, the negation and disjunction cases are trivial. Now consider the case $\exists^{\geq p} x \vartheta(x_1, \dots, x_n, x, z)$. Recall the definition in (9.2). To understand the following argument, it is crucial to know exactly which variables occur free in $\big(\exists^{\geq p} x \vartheta(x_1, \dots, x_n, x, z)\big)^*$ and its constituent formulae. The formula $\vartheta^1 := \vartheta(x_1, \dots, x_n, x, z)^*$ has free variables among $x_1, \dots, x_n, x$; we write $\vartheta^1(x_1, \dots, x_n, x)$ to make this explicit. The formula $\vartheta^2 := \vartheta(x_1, \dots, x_n, z, z)^*$ has free variables among $x_1, \dots, x_n$; we write $\vartheta^2(x_1, \dots, x_n)$. The formula $(\exists^{\geq p} x \vartheta)^*$ has free variables among $x_1, \dots, x_n$; we write $(\exists^{\geq p} x \vartheta)^*(x_1, \dots, x_n)$.

Let G be a graph, $A \subseteq G$, and all $u_1, \dots, u_n \in V(G) \setminus V(A)$ such that $V(A) = \xi[G, u_1, \dots, u_\ell, y]$. By the induction hypothesis, for all $u \in V(G) \setminus V(A)$, we have

$$G \models \vartheta^1(u_1, \dots, u_n, u) \iff G/A \models \vartheta(u_1, \dots, u_n, u, a) \tag{9.5}$$

and

$$G \models \vartheta^2(u_1, \dots, u_n) \iff G/A \models \vartheta(u_1, \dots, u_n, a, a). \tag{9.6}$$

To prove the forward direction of (9.4), suppose that $G \models (\exists^{\geq p} x \vartheta)^*(u_1, \dots, u_n)$.

(Case 1.) Suppose that $G \models \vartheta^2(u_1, \dots, u_n)$ and there are pairwise distinct $u^1, \dots, u^{p-1} \in V(G)$ such that for all j, it holds that $G \not\models \xi(u_1, \dots, u_\ell, u^j)$ and $G \models \vartheta^1(u_1, \dots, u_n, u^j)$.

Then it holds that $u^j \neq a$ by the assumption that $V(A) = \xi[G, u_1, \dots, u_\ell, y]$. Also, $G/A \models \vartheta(u_1, \dots, u_n, a, a)$ by (9.6) and $G/A \models \vartheta(u_1, \dots, u_n, u^j, a)$ by (9.5). Thus, $a, u^1, \dots, u^{p-1}$ witness that $G/A \models \exists^{\geq p} x \vartheta(u_1, \dots, u_n, x, a)$.

(Case 2.) Now suppose that there are pairwise distinct $u^1, \dots, u^p \in V(G)$ such that for all j, it holds that $G \not\models \xi(u_1, \dots, u_\ell, u^j)$ and $G \models \vartheta^2(u_1, \dots, u_n, u^j)$.

Then it holds that $u^j \neq a$ by the assumption that $V(A) = \xi[G, u_1, \dots, u_\ell, y]$. Furthermore, $G/A \models \vartheta(u_1, \dots, u_n, u^j, a)$ by (9.5). Thus, $u^1, \dots, u^p$ witness that $G/A \models \exists^{\geq p} x \vartheta(u_1, \dots, u_n, x, a)$.

The backward direction of (9.4) is proved by reverting the same argument. □

In the remainder of this chapter, we apply the results from this section in order to construct more complex formulae, meanwhile maintaining good bounds on their numbers of variables. This will allow us to eventually define the input graph and determine a bound on the number of variables of the corresponding formula.

9.4 Upper Bound on the WL Dimension

In this section, we give the proof of the main result of this chapter, which reads as follows.

Theorem 9.24. *The Weisfeiler-Leman dimension of the class of graphs of Euler genus g is at most $4g + 3$.*

By the correspondence between the k-dimensional Weisfeiler-Leman algorithm and the logic C^{k+1} as stated in Corollary 3.21, we need to prove that every graph of Euler genus at most g can be identified by a C^{4g+4}-sentence. The proof proceeds by induction on g. The base step $g = 0$ is Theorem 9.17.

For the inductive step, we make the following assumption.

Assumption 9.25. *Assume $g \geq 1$ and there is a natural number $s \geq 4$ such that every graph in $\mathcal{E}_{g-1}$ is identified by a C^s-sentence.*

Our goal is to prove the following lemma (under Assumption 9.25). The lemma implies Theorem 9.24 by induction.

Lemma 9.26 (Inductive Step). *For every coloured graph G in $\mathcal{E}_g$, there is a sentence $\mathsf{iso}_G \in \mathsf{C}_{\mathsf{w}}^{s+4}$ that identifies G.*

The proof will proceed in a sequence of lemmas. Eventually, it will diverge into two main cases, to be dealt with in Subsections 9.4.1 and 9.4.2. We first show that we can assume without loss of generality that $\rho(G) \geq 3$.

Lemma 9.27. *Let G be a coloured graph that has an embedding of representativity at most 2 into a surface of Euler genus at most g. Then there is a sentence $\mathsf{iso}_G \in \mathsf{C}_{\mathsf{w}}^{s+2}$ that identifies G.*

Proof: Suppose that G is embedded in a surface $\boldsymbol{S}$ of Euler genus g with representativity $\rho(G) \leq 2$. Let $\boldsymbol{g}$ be a G-normal non-contractible simple closed curve in $\boldsymbol{S}$, i.e. $U \coloneqq \boldsymbol{g} \cap V(G)$ contains at most two vertices. We only consider the case that $U = \{u_1, u_2\}$ for some $u_1, u_2 \in V(G)$ (possibly equal), the case $U = \emptyset$ follows similarly. Let $H_1, \ldots, H_m$ be the connected components of $G - U$. Every H_i can be embedded into a simpler surface obtained from $\boldsymbol{S}$ by cutting along $\boldsymbol{g}$ and gluing a disk on each hole. This means that $\text{eg}(H_i) \leq g - 1$. We colour the vertices of H_i so as to encode the adjacencies to u_1 and u_2. By Assumption 9.25, there is a C^s_w-sentence ψ_i that identifies the coloured version of H_i. Thus, by Corollary 9.21, there is a C^s_w-sentence ψ that identifies the disjoint union of the coloured H_i, that is, the coloured version of $G - \{u_1, u_2\}$. Now we can identify G via a sentence expressing that there exist vertices x_1, x_2 such that the deletion of these vertices produces a graph satisfying ψ and having the correct adjacencies to x_1, x_2. This requires $s + 2$ variables. □

So we can restrict our attention to graphs that only have embeddings of representativity at least 3. We now show that we can further restrict ourselves to 3-connected graphs at the cost of 1 more variable. For the following lemma, we assume that the reader is familiar with graph minors. We remark that for every $g \geq 0$, the class $\mathcal{E}_g$ of all graphs of Euler genus at most g is closed under taking minors. We will only apply the lemma to these classes.

For a class $\mathcal{C}$ of (uncoloured) graphs, we let $\mathcal{C}^*$ be the class of all coloured graphs with underlying graph in $\mathcal{C}$.

Lemma 9.28. *Let $\mathcal{C}$ be a minor-closed graph class. Let $k \in \mathbb{N}$. Suppose the k-dimensional Weisfeiler-Leman algorithm identifies all 3-connected graphs in $\mathcal{C}^*$. Then the $(k+1)$-dimensional Weisfeiler-Leman algorithm identifies all graphs in $\mathcal{C}^*$.*

Proof: By Theorem 7.37, the $(k+1)$-dimensional Weisfeiler-Leman algorithm identifies all graphs in $\mathcal{C}^*$ if the $(k+1)$-dimensional Weisfeiler-Leman algorithm determines vertex orbits on the class of all 3-connected graphs in $\mathcal{C}^*$ and identifies every such graph. Note that the identification is given by the assumptions. Thus, the statement follows from Proposition 3.17. □

Recall that a *polyhedral embedding* is an embedding of representativity at least 3 of a 3-connected graph. Thus, to prove Lemma 9.26 and thereby complete the proof of Theorem 9.24, it remains to prove the following lemma.

Lemma 9.29. *Let G be a coloured graph polyhedrally embedded in a surface $\boldsymbol{S}$ of Euler genus g. Then there is a sentence $\mathsf{iso}_G \in \mathsf{C}^{s+3}_\mathsf{w}$ that identifies G.*

For the rest of the section, we fix a positive integer n. The intended meaning of n is that it is the order of the target graph G. At this point, we have fixed three numerical parameters: the Euler genus g, the number s of variables required to identify graphs of smaller Euler genus, and the order n.

To prepare for the proof of Lemma 9.29, we define a number of useful concepts in $\mathsf{C}^k_{\mathsf{w}}$ for sufficiently small k.

We start the proof with a simple lemma that follows immediately from Assumption 9.25.

Lemma 9.30. *Let $h < g$. Then there is a sentence $\mathsf{genus}_h \in \mathsf{C}^s_{\mathsf{w}}$ such that for every graph G of order $|G| \leq n$, the following holds:*

$$G \models \mathsf{genus}_h \iff G \in \mathcal{E}_h.$$

Proof: Since there are only finitely many graphs of order at most n, we can let genus_h be a disjunction over the $\mathsf{iso}_H \in \mathsf{C}^s_{\mathsf{w}}$ for all $H \in \mathcal{E}_h$ with $|H| \leq n$. □

In the following lemmas, we are going to study the definability of shortest path systems, patches, and necklaces. Our strategy will then be to remove either a (definable) reducing necklace or a (definable) simplifying patch from the graph, then apply the induction hypothesis (Assumption 9.25) to the resulting simpler graph, and finally lift the identifying sentence to the original graph.

Lemma 9.31. *There are formulae*

$$\mathsf{csps\text{-}vert}(x, x', y) \in \mathsf{C}^3_{\mathsf{w}}, \qquad \mathsf{csps\text{-}edge}(x, x', y_1, y_2) \in \mathsf{C}^4_{\mathsf{w}}, \qquad \mathsf{csps\text{-}art}(x, x', y) \in \mathsf{C}^4_{\mathsf{w}},$$

and, for $i \geq 0$, formulae

$$\mathsf{csps\text{-}height}_k(x, x', y) \in \mathsf{C}^3_{\mathsf{w}}, \qquad \mathsf{csps\text{-}art}_i(x, x', y) \in \mathsf{C}^4_{\mathsf{w}}$$

such that for all connected graphs G of order $|G| \leq n$ and all $u, u' \in V(G)$,

$$\begin{aligned}
\mathsf{csps\text{-}vert}[G, u, u', y] &= V\big(\mathcal{Q}^G(u, u')\big),\\
\mathsf{csps\text{-}edge}[G, u, u', y_1, y_2] &= E\big(\mathcal{Q}^G(u, u')\big),\\
\mathsf{csps\text{-}art}[G, u, u', y] &= \mathrm{art}\left(\mathcal{Q}^G(u, u')\right),\\
\mathsf{csps\text{-}height}_i[G, u, u', y] &= \Big\{v \in V\big(\mathcal{Q}^G(u, u')\big) \;\Big|\; \mathrm{ht}^{\mathcal{Q}^G(u,u')}(v) = i\Big\},\\
\mathsf{csps\text{-}art}_i[G, u, u', y] &= \{v\}, \text{ where } v \text{ is the } i\text{-th vertex when sort-}\\
&\qquad\qquad \text{ing } \mathrm{art}\left(\mathcal{Q}^G(u, u')\right) \text{ by height.}
\end{aligned}$$

Recall that $\mathcal{Q}^G(u, u')$ is the canonical sps from u to u', that is, the set of all shortest paths from u to u'.

Proof: We let

$$\mathsf{csps\text{-}vert}(x, x', y) := \bigvee_{k=0}^{n} \left(\mathsf{dist}_{=k}(x, x') \wedge \bigvee_{i=0}^{k} \big(\mathsf{dist}_{=i}(x, y) \wedge \mathsf{dist}_{=k-i}(y, x')\big)\right),$$

where $\mathsf{dist}_{=k}(x, x')$ is the $\mathsf{C}^3_{\mathsf{w}}$-formula defined in Example 2.4. Note that we have $\mathsf{csps\text{-}vert}(x, x', y) \in \mathsf{C}^3_{\mathsf{w}}$.

Since a vertex v lies on a shortest path from u to u' if and only if taking the shortest path from u to v and then to u' yields no detour, the formula $\mathsf{csps\text{-}vert}$ defines the desired set of vertices.

An edge is contained in $E\big(\mathcal{Q}^G(u,u')\big)$ if and only if it connects an sps-vertex of a certain height h with an sps-vertex of height $h+1$. Thus, it is easy to see that the formula for $\mathsf{csps\text{-}edge}$ can be constructed to have width 4.

A vertex v is an articulation vertex of $\mathcal{Q}^G(u,u')$ if every shortest path from u to u' contains v:

$$\begin{aligned}\mathsf{csps\text{-}art}(x,x',y) :=\ & \mathsf{csps\text{-}vert}(x,x',y) \wedge \\ & \forall z\Big(\mathsf{csps\text{-}vert}(x,x',z) \to \big(\mathsf{csps\text{-}vert}(x,y,z) \vee \mathsf{csps\text{-}vert}(y,x',z)\big)\Big).\end{aligned}$$

This formula has width 4.

Similarly, the height of v in $\mathcal{Q}^G(u,u')$ is i if and only if v is contained in the sps and $G \models \mathsf{dist}_{=i}(u,v)$. Thus, we can construct $\mathsf{csps\text{-}height}_i(x,x',y)$ with width 3.

By employing $\mathsf{csps\text{-}art}$ and $\mathsf{csps\text{-}height}_i$, we can also construct $\mathsf{csps\text{-}art}_i$ with width 4. □

The lemma shows how to define canonical shortest paths systems. We would also like to define patches and necklaces, but they depend on the embedding and since the embedding may not be unique, in general the property of an sps being a patch is not definable in a logic which only has access to the abstract graph and not the embedding. We therefore define "pseudo-patches" and "pseudo-necklaces" purely in terms of the abstract graph; in some situations they may serve as substitutes for the real object.

Definition 9.32. Let G be a graph.

1. A *pseudo-patch* in G is an sps that has no articulation vertices.
2. A *pseudo-necklace* in G is a tuple $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$, for which $u^0, u^1, u^2 \in V(G)$ and $\mathcal{Q}^i = \mathcal{Q}^G(u^i, u^{i+1})$ (indices taken modulo 3) is the canonical sps from u^i to u^{i+1} such that u^0, u^1, u^2 are pairwise distinct and $V(\mathcal{Q}^i) \cap V(\mathcal{Q}^{i+1}) = \{u^{i+1}\}$. ⌟

All the definitions for general sps apply to pseudo-patches, and we can generalise all definitions that do not refer to the embedding (for example, $V(\mathcal{B})$, $E(\mathcal{B})$, $\mathrm{art}(\mathcal{B})$, etc.) from necklaces to pseudo-necklaces. Observe that every patch is a pseudo-patch and every necklace is a pseudo-necklace.

Corollary 9.33. *There are $\mathsf{C}^4_{\mathsf{w}}$-formulae*

$$\begin{aligned} &\mathsf{nl\text{-}vert}(x^0,x^1,x^2,y), && \mathsf{nl\text{-}edge}(x^0,x^1,x^2,y),\\ &\mathsf{nl\text{-}art}(x^0,x^1,x^2,y), && \mathsf{nl\text{-}art}_i(x^0,x^1,x^2,y), \end{aligned}$$

such that for all connected graphs G of order $|G| \leq n$ and all $u^0, u^1, u^2 \in V(G)$, the following holds. If $\mathcal{B} := \big(u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2\big)$ is a pseudo-necklace in G, then

$$\begin{aligned} \mathsf{nl\text{-}vert}[G, u^0, u^1, u^2, y] &= V(\mathcal{B}); \\ \mathsf{nl\text{-}edge}[G, u^0, u^1, u^2, y_1, y_2] &= E(\mathcal{B}); \\ \mathsf{nl\text{-}art}[G, u^0, u^1, u^2, y] &= \operatorname{art}(\mathcal{B}); \\ \mathsf{nl\text{-}art}_i[G, u^0, u^1, u^2, y] &= \{v\}, \text{ \textit{where v is the i-th vertex in the linear order}} \\ &\quad \text{\textit{of} } \operatorname{art}(\mathcal{B}) \text{ \textit{that sorts the articulation vertices of}} \\ &\quad \mathcal{Q}^0,\ \mathcal{Q}^1, \text{ \textit{and} } \mathcal{Q}^2 \text{ \textit{by increasing height and puts}} \\ &\quad \text{\textit{the articulation vertices of} } \mathcal{Q}^0 \text{ \textit{before those of}} \\ &\quad \mathcal{Q}^1 \text{ \textit{and the latter ones before those of} } \mathcal{Q}^2. \end{aligned}$$

Proof: For a vertex $v \in V(G)$, we have that $v \in V(\mathcal{B})$ if and only if $v \in \mathcal{Q}^G(u^i, u^{i+1})$ for some $i \in \{0, 1, 2\}$ (indices taken modulo 3). Similarly, an edge is a necklace edge if and only if for some i, it connects a vertex $v \in V\big(\mathcal{Q}^G(u^i, u^{i+1})\big)$ of a certain height h with a vertex of height $h + 1$ in $V\big(\mathcal{Q}^G(u^i, u^{i+1})\big)$. Thus, containment in $V(\mathcal{B})$ and in $E(\mathcal{B})$ is definable in $\mathsf{C}^4_{\mathsf{w}}$.

A vertex v is an articulation vertex of $\mathcal{B}$ if v is an articulation vertex of $\mathcal{Q}^G(u^i, u^{i+1})$ for an $i \in \{0, 1, 2\}$. Thus, we can construct $\mathsf{nl\text{-}art}$ to have width at most 4.

We can construct $\mathsf{nl\text{-}art}_i$ in a straightforward manner by employing the subformulae $\mathsf{nl\text{-}art}$ and $\mathsf{csps\text{-}height}_i$. □

From Lemmas 9.18 and 9.31, we obtain the following corollary.

Corollary 9.34. *There is a formula $\mathsf{csps\text{-}comp}(x, x', y, y') \in \mathsf{C}^5_{\mathsf{w}}$ such that for all connected graphs G of order $|G| \leq n$ and all $u, u', v, v' \in V(G)$,*

$$G \models \mathsf{csps\text{-}comp}(u, u', v, v') \iff \text{\textit{v and v' belong to the same connected component of} } G - V\big(\mathcal{Q}^G(u, u')\big).$$

From Lemma 9.22 applied to the $\mathsf{C}^s_{\mathsf{w}}$-sentence $\psi := \mathsf{genus}_h$ of Lemma 9.30 and the $\mathsf{C}^3_{\mathsf{w}}$-formula $\varphi(x, x', y) := \mathsf{csps\text{-}vert}(x, x', y)$ of Lemma 9.31, we obtain the following corollary.

Corollary 9.35. *Let $h < g$. Then there is a formula $\mathsf{csps\text{-}comp\text{-}genus}_h(x, x', y) \in \mathsf{C}^{s+2}_{\mathsf{w}}$ such that for all connected graphs G of order $|G| \leq n$ and all $u, u', v \in V(G)$, the following holds. Let $\mathcal{Q} := \mathcal{Q}^G(u, u')$, and let A be the connected component of v in $G - V(\mathcal{Q})$ (assuming $v \notin V(\mathcal{Q})$). Then*

$$G \models \mathsf{csps\text{-}comp\text{-}genus}_h(u, u', v) \iff v \notin V(\mathcal{Q}) \text{ \textit{and} } \operatorname{eg}(A) \leq h.$$

Corollary 9.36. *There is a formula $\mathsf{csps\text{-}simplifying}(x, x') \in \mathsf{C}^{s+2}_{\mathsf{w}}$ such that for all connected graphs $G \in \mathcal{E}_g$ of order $|G| \leq n$ and all $u, u' \in V(G)$,*

$$G \models \mathsf{csps\text{-}simplifying}(u, u') \iff \mathcal{Q}^G(u, u') \text{ \textit{is simplifying.}}$$

The formulae we have defined so far make no reference to an embedding of the input graph. However, if we want to talk about patches and necklaces, we need to take the embedding into account. For the rest of the section, we fix a specific embedded graph G.

Assumption 9.37. *G is a coloured graph of order $|G| = n$ that is polyhedrally embedded in a surface $\boldsymbol{S}$ of Euler genus g.*

It is our goal to construct a $\mathsf{C}_{\mathsf{w}}^{s+3}$-sentence that identifies G.

Intuitively, the following lemma says that even though the logical formulae only have access to the abstract graph, whereas the disk of a patch and the internal graph depend on the embedding, we can still define the internal graph. This is non-trivial and somewhat surprising.

Lemma 9.38. *There are $\mathsf{C}_{\mathsf{w}}^{7}$-formulae*

$$\begin{array}{ll} \mathsf{int\text{-}vert}(x, x', y), & \mathsf{int\text{-}edge}(x, x', y_1, y_2), \\ \mathsf{bd\text{-}vert}(x, x', y), & \mathsf{bd\text{-}edge}(x, x', y_1, y_2) \end{array}$$

such that for all vertices $u, u' \in V(G)$ for which $\mathcal{Q} := \mathcal{Q}^G(u, u')$ is a non-trivial non-simplifying patch, the following hold:

$$\begin{aligned} \mathsf{int\text{-}vert}[G, u, u', y] &= V\big(I(\mathcal{Q})\big), \\ \mathsf{int\text{-}edge}[G, u, u', y_1, y_2] &= E\big(I(\mathcal{Q})\big), \\ \mathsf{bd\text{-}vert}[G, u, u', y] &= V\big(C(\mathcal{Q})\big), \\ \mathsf{bd\text{-}edge}[G, u, u', y_1, y_2] &= E\big(C(\mathcal{Q})\big). \end{aligned}$$

Proof: Let $u, u' \in V(G)$ such that $\mathcal{Q} := \mathcal{Q}^G(u, u')$ is a non-trivial non-simplifying patch. Let $\boldsymbol{D} := \boldsymbol{D}(\mathcal{Q})$, $C := C(\mathcal{Q})$, and $I := I(\mathcal{Q})$ (see Section 9.2).

By Lemma 9.9, the graph $G - V(\mathcal{Q})$ has a unique non-planar connected component A^*. We let

$$\mathsf{planar\text{-}comp}(x, x', y) := \mathsf{csps\text{-}comp\text{-}genus}_0(x, x', y) \qquad \text{(see Corollary 9.35)}$$

and

$$\mathsf{astar}(x, x', y) := \neg\mathsf{csps\text{-}vert}(x, x', y) \wedge \neg\mathsf{planar\text{-}comp}(x, x', y).$$

Note that both $\mathsf{planar\text{-}comp}(x, x', y)$ and $\mathsf{astar}(x, x', y)$ are $\mathsf{C}_{\mathsf{w}}^{s+2}$-formulae. In fact, since we can identify planar graphs in the logic $\mathsf{C}_{\mathsf{w}}^{4}$, we can construct these formulae as $\mathsf{C}_{\mathsf{w}}^{6}$-formulae. For $v \in V(G) \setminus V(\mathcal{Q})$, we have $G \models \mathsf{planar\text{-}comp}(u, u', v)$ if and only if the connected component of v in $G - V(\mathcal{Q})$ is planar, and $G \models \mathsf{astar}(u, u', v)$ if and only if the connected component of v in $G - V(\mathcal{Q})$ is A^*.

Let v_1 be a vertex in $V(\mathcal{Q})$ that is adjacent to A^* and among all such vertices has minimal height in the sps, and let $h := \mathrm{ht}^{\mathcal{Q}}(v_1)$. Since A^* is embedded outside of the disk $\boldsymbol{D}$, the vertex v_1 must be on the boundary cycle C of $\boldsymbol{D}$. There is at most one other vertex of height h on this cycle. Thus, even though v_1 is not unique, there are at most two choices. If there is a second vertex of height h adjacent to A^*, let us call it v_1'. Let

$$\begin{aligned}\varphi_1(x,x',y_1) := \mathsf{csps\text{-}height}_h(x,x',y_1) \wedge \exists z^*\big(E(z^*,y_1) \wedge \mathsf{astar}(x,x',z^*)\big) \wedge \\ \neg\exists y_1'\left(\bigvee_{i=0}^{h-1} \mathsf{csps\text{-}height}_i(x,x',y_1') \wedge \exists z^*\big(E(z^*,y_1') \wedge \mathsf{astar}(x,x',z^*)\big)\right).\end{aligned}$$

Then v_1 and possibly v_1' are the only vertices in $\varphi_1[G,u,u',y_1]$. Note that $\varphi_1 \in \mathsf{C}^6_\mathsf{w}$.

Recall that G/A^* denotes the graph obtained from G by contracting the connected subgraph A^* to a single vertex, which we call a^*, and that the graph G/A^* is a 3-connected planar graph. By Whitney's Theorem, the facial subgraphs of a 3-connected plane graph are precisely the chordless non-separating cycles. In particular, they are independent of the embedding. Furthermore, every edge is contained in exactly two of these facial cycles. Let us consider the edge $\{v_1, a^*\}$ in the graph G/A^*. Let F and F' be the two facial cycles that contain this edge. Both F and F' contain exactly one neighbour of a^* distinct from v_1. Let v_2 and v_2' be these neighbours.

By Lemma 8.2, if we have a 3-connected planar graph H and three vertices w_1, w_2, w_3 on a common facial cycle, then after individualising these three vertices, the 1-dimensional Weisfeiler-Leman algorithm computes a discrete colouring. By Theorem 3.19, this implies that for every $w \in V(H)$, there is a formula $\psi_{H,w}(z_1,z_2,z_3,y) \in \mathsf{C}^5_\mathsf{w}$ such that $\psi_{H,w}[H,w_1,w_2,w_3,y] = \{w\}$. We apply Lemma 8.2 to the graph G/A^* and the three vertices a^*, v_1, v_2 and obtain, for every vertex $w \in V(G/A^*) = (V(G) \setminus A^*) \cup \{a^*\}$, a formula $\psi_w(z^*,y_1,y_2,y) \in \mathsf{C}^5_\mathsf{w}$ such that $\psi_w[G/A^*,a^*,v_1,v_2,y] = \{w\}$.

Let $w \in V(G) \setminus A^*$. Recall that $\mathsf{astar}(x,x',y) \in \mathsf{C}^6_\mathsf{w}$ and $\psi_w(z^*,y_1,y_2,y) \in \mathsf{C}^5_\mathsf{w}$. By Lemma 9.23 (applied to $k=6$, $\ell=2$, $m=5$ and the formulae $\xi(x,x',z^*) := \mathsf{astar}(x,x',z^*)$ and $\psi(y_1,y_2,y,z^*) := \psi_w(z^*,y_1,y_2,y)$), we know that there is a formula $\widetilde{\psi}_w(x,x',y_1,y_2,y) \in \mathsf{C}^6_\mathsf{w}$ such that $\widetilde{\psi}_w[G,u,u',v_1,v_2,y] = \{w\}$.

Since $A^* \cap \boldsymbol{D} = \emptyset$, we have $V(I) = V(G) \cap \boldsymbol{D} \subseteq V(G - A^*)$. We let

$$\delta(x,x',y_1,y_2,y) := \bigvee_{w \in V(I)} \widetilde{\psi}_w(x,x',y_1,y_2,y).$$

Then $\delta[G,u,u',v_1,v_2,y] = V(I)$. Thus $\delta(x,x',y_1,y_2,y)$ is almost the formula $\mathsf{int\text{-}vert}(x,x',y)$ we want, except that it has two additional parameters v_1, v_2 which we have to get rid of.

We will apply Corollary 8.6, which says that the 3-dimensional Weisfeiler-Leman algorithm determines vertex orbits on the class of all edge-coloured 3-connected planar graphs and identifies these. This implies that within a given edge-coloured 3-connected planar graph, the 3-dimensional Weisfeiler-Leman algorithm distinguishes two vertices if and only if they belong to different orbits of the automorphism group of the graph. It follows that for every 3-connected planar graph H and for every orbit O of the automorphism group of H, there is a formula $\xi_{H,O}(y_2) \in \mathsf{C}^4_{\mathsf{w}}$ such that $\xi_{H,O}[H, y_2] = O$.

To eliminate the parameter v_2, we apply the corollary to the graph G/A^*, but only after individualising the vertices a^* and v_1. That is, we modify the colouring such that each of the two vertices has its own colour and is thus fixed by all automorphisms. Let O_2 be the orbit of v_2 in this coloured graph. By the definition of v_2, either $O_2 = \{v_2, v_2'\}$ or $O_2 = \{v_2\}$. Since the graph G/A^* is 3-connected, by eliminating the colour relations for a^* and v_1 at the cost of new free variables z^* and y_1, we obtain a new formula $\psi_2(z^*, y_1, y_2) \in \mathsf{C}^6_{\mathsf{w}}$ such that $\psi_2[G/A^*, a^*, v_1, y_2] = O_2$. Since $\mathsf{astar}(x, x', z^*) \in \mathsf{C}^6_{\mathsf{w}}$ and $\xi_{H,O}(y) \in \mathsf{C}^4_{\mathsf{w}}$, by Lemma 9.23 (with $k = 6$, $\ell = 2$, $m = 4$ and the formulae $\underset{\sim}{\xi}(x, x', z^*) := \mathsf{astar}(x, x', z^*)$ and $\underset{\sim}{\psi}(y_1, y_2, z^*) := \psi_2(z^*, y_1, y_2)$), there is a formula $\widetilde{\psi}_2(x, x', y_1, y_2) \in \mathsf{C}^6_{\mathsf{w}}$ such that $\widetilde{\psi}_2[G, u, u', v_1, y_2] = O_2$. We let

$$\delta'(x, x', y_1, y) := \exists y_2\big(\widetilde{\psi}_2(x, x', y_1, y_2) \wedge \delta(x, x', y_1, y_2, y)\big).$$

If $O_2 = \{v_2\}$, then clearly $\delta'[G, u, u', v_1, y] = \delta[G, u, u', v_1, v_2, y] = V(I)$. So suppose that $O_2 = \{v_2, v_2'\}$, and let f be an automorphism of G with $f(u) = u$, $f(u') = u'$, $f(v_1) = v_1$, and $f(v_2) = v_2'$. By Corollary 9.11, we have $f\big(V(I)\big) = V(I)$ and thus

$$\begin{aligned}\delta[G, u, u', v_1, v_2', y] &= \delta[f(G), f(u), f(u'), f(v_1), f(v_2), y]\\ &= f\big(\delta[G, u, u', v_1, v_2, y]\big)\\ &= f\big(V(I)\big) = V(I).\end{aligned}$$

It follows that

$$\delta'[G, u, u', v_1, y] = \delta[G, u, u', v_1, v_2, y] \cup \delta[G, u, u', v_1, v_2', y] = V(I).$$

So we have eliminated the parameter v_2. To eliminate v_1, we use essentially the same argument. Let O_1 be the orbit of v_1 in the graph G/A^* with the vertices a^*, u, u' individualised. Then either $O_1 = \{v_1, v_1'\}$ for some $v_1' \neq v_1$ or $O_1 = \{v_1\}$.

By Corollary 8.6, there is a formula $\xi_{G/A^*,O_1}(y_1) \in \mathsf{C}^4_{\mathsf{w}}$ such that

$$\xi_{G/A^*,O_1}[(G/A^*)_{(a^*,u,u')}, y_1] = O_1.$$

Then by eliminating the colour relations for a^*, u, u' at the cost of new free variables z^*, x, x', we obtain a new formula $\psi_1(z^*, x, x', y_1) \in \mathsf{C}^7_{\mathsf{w}}$ such that $\psi_1[G/A^*, a^*, u, u', y_1] = O_1$. Since $\mathsf{astar}(x, x', z^*) \in \mathsf{C}^6_{\mathsf{w}}$ and $\xi_{G/A^*,O}(y_1) \in \mathsf{C}^4_{\mathsf{w}}$, by Lemma 9.23 (with $k = 7$, $\ell = 2$, $m = 3$ and $\xi(x, x', z^*) := \mathsf{astar}(x, x', z^*)$ and $\underset{\sim}{\psi}(x, x', y_1, z^*) := \psi_1(z^*, x, x', y_1)$), there is a formula $\widetilde{\psi}_1(x, x', y_1) \in \mathsf{C}^7_{\mathsf{w}}$ such that $\widetilde{\psi}_1[G, u, u', y_1] = O_1$.

We let

$$\mathsf{int\text{-}vert}(x, x', y) := \exists y_1\big(\widetilde{\psi}_1(x, x', y_1) \wedge \delta'(x, x', y_1, y)\big).$$

Now a similar argument as above shows that $\mathsf{int\text{-}vert}[G, u, u', z] = V(I)$. Moreover, since $\delta', \widetilde{\psi}_1 \in \mathsf{C}^7_{\mathsf{w}}$, we have $\mathsf{int\text{-}vert} \in \mathsf{C}^7_{\mathsf{w}}$.

The formulae $\mathsf{int\text{-}edge}(x, x', y_1, y_2)$, $\mathsf{bd\text{-}vert}(x, x', y)$, $\mathsf{bd\text{-}edge}(x, x', y_1, y_2)$ can be constructed similarly. □

Now we branch into two cases, depending on whether G contains a simplifying patch or not.

9.4.1 Case 1: Absence of Simplifying Patches

Throughout this subsection, in addition to Assumption 9.37, we assume the following.

Assumption 9.39. *G does not contain any simplifying patches.*

By Lemma 9.14, G contains a reducing necklace $\mathcal{B}$. We are going to define a subgraph $\mathrm{Cut}(\mathcal{B})$ of G that is obtained from G by “cutting through the beads”. Since the necklace is reducing, the Euler genus of every connected component of $\mathrm{Cut}(\mathcal{B})$ is at most $g-1$ and we can identify it with a C^s-sentence using Assumption 9.25. We colour $\mathrm{Cut}(\mathcal{B})$ in such a way that we can reconstruct G and identify it using only 3 more variables.

For a necklace $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$ in G, let $u^i = u^i_0, u^i_1, \ldots, u^i_{n_i} = u^{i+1}$ be the articulation vertices of $\mathcal{Q}^i$, ordered by height, and for $j \in [0, n_i - 1]$, let $\mathcal{Q}^i_j := \mathcal{Q}^i[u^i_j, u^i_{j+1}]$ be the segment of $\mathcal{Q}^i$ between u^i_j and u^i_{j+1}. If the patch $\mathcal{Q}^i_j$ is trivial, we denote its unique edge by e^i_j. If Q^i_j is non-trivial, we let $\boldsymbol{D}^i_j := \boldsymbol{D}(\mathcal{Q}^i_j)$.

The *region* of $\mathcal{B}$ is the point set

$$\boldsymbol{R}(\mathcal{B}) := \bigcup_{i=0}^{2}\Big(\bigcup_{\substack{j\in[0,n_i-1]\\ \mathcal{Q}^i_j \text{ non-trivial}}} \boldsymbol{D}^i_j \cup \bigcup_{\substack{j\in[0,n_i-1]\\ \mathcal{Q}^i_j \text{ trivial}}} \boldsymbol{e}^i_j\Big).$$

Recall that the internal graph of a non-trivial non-simplifying patch $\mathcal{Q}$ is the graph $I := I(\mathcal{Q})$ with vertex set $V(I) := V(G) \cap \boldsymbol{D}(\mathcal{Q})$ and edge set $E(I) := \{\boldsymbol{e} \in E(G) \mid \boldsymbol{e} \subseteq \boldsymbol{D}(\mathcal{Q})\}$. We associate the following three subgraphs of G with $\mathcal{B}$.

Definition 9.40. The *inside* of $\mathcal{B}$ is $I(\mathcal{B}) := \bigcup_{i=0}^{2}\bigcup_{j=0}^{n_i-1} I(\mathcal{Q}^i_j)$.

The *outside* of $\mathcal{B}$ is the graph $O(\mathcal{B})$ defined by

$$\begin{aligned} V\big(O(\mathcal{B})\big) &:= V(G) \setminus \boldsymbol{int}(\boldsymbol{R}),\\ E\big(O(\mathcal{B})\big) &:= E(G) \setminus \big\{\boldsymbol{e} \in E(G) \bigm| \boldsymbol{e} \cap \boldsymbol{int}(\boldsymbol{R}) \neq \emptyset\big\}. \end{aligned}$$

The *cut graph* of $\mathcal{B}$ is $\mathrm{Cut}(\mathcal{B}) := O(\mathcal{B}) \setminus \mathrm{art}(\mathcal{B})$.

Lemma 9.41. *Suppose $\mathcal{B}$ is a reducing necklace in G. Then every connected component of* $\mathrm{Cut}(\mathcal{B})$ *is in* $\mathcal{E}_{g-1}$.

Proof: This proof is a slight adaptation of the proof of [64, Lemma 15.5.6].

Let $\boldsymbol{R} \coloneqq \boldsymbol{R}(\mathcal{B})$. For all i, j such that $\mathcal{Q}^i_j$ is non-trivial, we let $\boldsymbol{D}^i_j \coloneqq \boldsymbol{D}(\mathcal{Q}^i_j)$.

Let B be a non-contractible cycle in $\mathcal{B}$, whose existence is guaranteed by Definition 9.13. Then $\boldsymbol{B} \subseteq \boldsymbol{R}$ is a simple closed curve, and for all i, j such that $\mathcal{Q}^i_j$ is non-trivial, the intersection $\boldsymbol{B}^i_j \coloneqq \boldsymbol{D}^i_j \cap \boldsymbol{B}$ is a simple curve in the disk $\boldsymbol{D}^i_j$ with endpoints u^i_j and u^i_{j+1}. By slightly perturbing $\boldsymbol{B}$, we obtain a homotopic simple closed curve $\boldsymbol{b} \subseteq \boldsymbol{R}$ such that for all i, j with non-trivial $\mathcal{Q}^i_j$, we have $\boldsymbol{b} \cap \boldsymbol{bd}(\boldsymbol{D}^i_j) = \{u^i_j, u^i_{j+1}\}$. This new curve $\boldsymbol{b}$ is still non-contractible, and it intersects $\boldsymbol{bd}(\boldsymbol{R})$ only in the articulation vertices u^i_j of $\mathcal{B}$ and in the edges $\boldsymbol{e}^i_j$ of the trivial patches $\mathcal{Q}^i_j$.

This implies that for $H \coloneqq \mathrm{Cut}(\mathcal{B})$, we have $\boldsymbol{b} \cap \boldsymbol{H} = \emptyset$. Thus $\boldsymbol{H} \subseteq G - \boldsymbol{b}$, and since $\boldsymbol{b}$ is non-contractible, this implies that every connected component of H is embeddable in a surface of Euler genus at most $g - 1$ obtained from $\boldsymbol{S}$ by cutting along $\boldsymbol{g}$ and gluing a disk on each hole. □

Our next goal is to show that the cut graph is definable in $\mathsf{C}^{s+3}_{\mathsf{w}}$.

From Lemma 9.38, we obtain that $\mathsf{C}^7_{\mathsf{w}}$ distinguishes the internal graph of a reducing necklace from the remainder of the graph.

Corollary 9.42. *There are* $\mathsf{C}^7_{\mathsf{w}}$*-formulae*

$$\mathsf{nl\text{-}int\text{-}vert}(x^0, x^1, x^2, y), \qquad \mathsf{nl\text{-}int\text{-}edge}(x^0, x^1, x^2, y_1, y_2),$$

such that for $u^0, u^1, u^2 \in V(G)$, the following holds.

If $\mathcal{B} \coloneqq (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$ is a necklace in G, then

$$\begin{aligned} \mathsf{nl\text{-}int\text{-}vert}[G, u^0, u^1, u^2, y] &= V\big(I(\mathcal{B})\big), \\ \mathsf{nl\text{-}int\text{-}edge}[G, u^0, u^1, u^2, y_1, y_2] &= E\big(I(\mathcal{B})\big). \end{aligned}$$

Proof: Remember that we suppose Assumption 9.39 holds. Thus, we can simply define

$$\mathsf{nl\text{-}int\text{-}vert}(x^0, x^1, x^2, y) \coloneqq \bigvee_{i=0}^{2} \mathsf{int\text{-}vert}(x^i, x^{i+1}, y).$$

Similarly, we obtain the formula $\mathsf{nl\text{-}int\text{-}edge}$ with the desired width. □

In the following, we show that C^7 distinguishes vertices in the outside and the cut graph of $\mathcal{B}$ from the rest of the graph.

Lemma 9.43. *There are* $\mathsf{C}^7_{\mathsf{w}}$*-formulae*

$$\begin{array}{ll} \mathsf{nl\text{-}out\text{-}vert}(x^0,x^1,x^2,y), & \mathsf{nl\text{-}out\text{-}edge}(x^0,x^1,x^2,y), \\ \mathsf{nl\text{-}cut\text{-}vert}(x^0,x^1,x^2,y), & \mathsf{nl\text{-}cut\text{-}edge}(x^0,x^1,x^2,y_1,y_2), \end{array}$$

such that for all vertices $u^0,u^1,u^2 \in V(G)$*, the following holds: if* $\mathcal{B} := (u^0,\mathcal{Q}^0,u^1,\mathcal{Q}^1,u^2,\mathcal{Q}^2)$ *is a necklace in* G*, then*

$$\begin{aligned} \mathsf{nl\text{-}out\text{-}vert}[G,u^0,u^1,u^2,y] &= V\big(O(\mathcal{B})\big), \\ \mathsf{nl\text{-}out\text{-}edge}[G,u^0,u^1,u^2,y_1,y_2] &= E\big(O(\mathcal{B})\big), \\ \mathsf{nl\text{-}cut\text{-}vert}[G,u^0,u^1,u^2,y] &= V\big(\operatorname{Cut}(\mathcal{B})\big), \\ \mathsf{nl\text{-}cut\text{-}edge}[G,u^0,u^1,u^2,y_1,y_2] &= E\big(\operatorname{Cut}(\mathcal{B})\big). \end{aligned}$$

Proof: Let $\boldsymbol{R} := \boldsymbol{R}(\mathcal{B})$ and recall that $u^i = u^i_0, u^i_1, \ldots, u^i_{n_i} = u^{i+1}$ denote the articulation vertices of $\mathcal{B}$, ordered by height, and that for $j \in [0, n_i - 1]$, we denote the segment of $\mathcal{Q}^i$ between u^i_j and u^i_{j+1} by $\mathcal{Q}^i_j := \mathcal{Q}^i[u^i_j, u^i_{j+1}]$. Since by Assumption 9.39, all subpatches are non-simplifying, it holds that

$$V(G) \cap \boldsymbol{int}(\boldsymbol{R}) = \bigcup_{i=0}^{2} \bigcup_{\substack{j\in[0,n_i-1] \\ \mathcal{Q}^i_j \text{ non-trivial}}} V\left(I(\mathcal{Q}^i_j) \setminus C(\mathcal{Q}^i_j)\right).$$

Therefore,

$$\begin{aligned} V\big(O(\mathcal{B})\big) &= V(G) \setminus \bigcup_{i=0}^{2} \bigcup_{\substack{j\in[0,n_i-1] \\ \mathcal{Q}^i_j \text{ non-trivial}}} V\big(I(\mathcal{Q}^i_j) \setminus C(\mathcal{Q}^i_j)\big), \\ E\big(O(\mathcal{B})\big) &= E(G) \setminus \bigcup_{i=0}^{2} \bigcup_{\substack{j\in[0,n_i-1] \\ \mathcal{Q}^i_j \text{ non-trivial}}} E\big(I(\mathcal{Q}^i_j) \setminus C(\mathcal{Q}^i_j)\big). \end{aligned}$$

Thus, we can just let

$$\begin{aligned} \mathsf{nl\text{-}out\text{-}vert}(x^0,x^1,x^2,y) := \bigvee_{i=1}^{n} & \exists z \exists z' \big(\mathsf{nl\text{-}art}_i(x^0,x^1,x^2,z) \wedge \mathsf{nl\text{-}art}_{i+1}(x^0,x^1,x^2,z') \\ & \wedge \neg \mathsf{nl\text{-}edge}(x^0,x^1,x^2,z,z') \wedge \mathsf{bd\text{-}vert}(z,z',y)\big) \\ & \vee \mathsf{nl\text{-}art}(x^0,x^1,x^2,y) \vee \neg \mathsf{nl\text{-}int\text{-}vert}(x^0,x^1,x^2,y), \end{aligned}$$

where the big disjunction expresses that the given vertex lies on the boundary of some disk of a non-trivial patch.

Similarly, we obtain the formula $\mathsf{nl\text{-}out\text{-}edge}$ of width 7.

To define that a vertex is contained in the cut graph, we just need to guarantee that it is contained in $O(\mathcal{B})$ and that is not an articulation vertex of the necklace. Similarly, for an edge contained in $O(\mathcal{B})$, to appear in $\mathrm{Cut}(\mathcal{B})$, its incident vertices must not be articulation vertices of $\mathcal{B}$. We obtain the C^7_w-formulae $\mathsf{nl\text{-}cut\text{-}vert}$ and $\mathsf{nl\text{-}cut\text{-}edge}$. □

We have collected all ingredients to show the statement from Lemma 9.29 in case G contains no simplifying patches.

Proof of Lemma 9.29, Case 1: We show that the statement holds if $g \geq 1$ and G does not contain any simplifying patches.

Recall that, by Assumption 9.25, for every coloured graph $H \in \mathcal{E}_{g-1}$, we assume the existence of a formula $\mathsf{iso}_H \in \mathsf{C}^s_\mathsf{w}$ such that for all graphs G', it holds that

$$G' \models \mathsf{iso}_H \iff G' \cong H.$$

Let G be a coloured graph that does not contain any simplifying patches and is polyhedrally embedded in a surface $\boldsymbol{S}$ of Euler genus $g \geq 1$. Let $\widehat{G}$ be a second coloured graph such that there is no formula in C^{s+3} which distinguishes G and $\widehat{G}$. We show that $G \cong \widehat{G}$.

We may assume $|\widehat{G}| = |G|$, otherwise we can distinguish G and $\widehat{G}$ via the formula $\exists^{=|G|} v(v = v)$.

Theorem 7.26 implies that, if for some $k \geq 2$, the logic C^k distinguishes all non-isomorphic pairs of coloured 2-connected graphs in a hereditary graph class $\mathcal{G}$, then it distinguishes all pairs of non-isomorphic graphs in $\mathcal{G}$. Thus, if C^{s+3} distinguishes (the 2-connected graph) G from every non-isomorphic 2-connected coloured graph, then the same logic distinguishes G from every arbitrary non-isomorphic coloured graph and thus, it identifies G. Hence, we can assume $\widehat{G}$ to be 2-connected.

Moreover, if $\widehat{G}$ is not 3-connected, then it has a separator of size 2, whereas G does not. Since for $k \geq 2$, the k-dimensional Weisfeiler-Leman algorithm distinguishes 2-separators from other pairs of vertices (see Theorem 7.7), by Corollary 3.21, there is a formula in C^3_w which distinguishes G and $\widehat{G}$.

Hence, without loss of generality, we may assume that $\widehat{G}$ is 3-connected.

By Lemma 9.14, there is a reducing necklace $\mathcal{B} := (u^0, \mathcal{Q}^0, u^1, \mathcal{Q}^1, u^2, \mathcal{Q}^2)$ in G, which we fix for the remainder of this proof. For a pseudo-necklace $\widehat{\mathcal{B}} := (\widehat{u}^0, \widehat{\mathcal{Q}}^0, \widehat{u}^1, \widehat{\mathcal{Q}}^1, \widehat{u}^2, \widehat{\mathcal{Q}}^2)$ in $\widehat{G}$, we say $\mathcal{B}$ and $\widehat{\mathcal{B}}$ are *isomorphic*, and write $\mathcal{B} \cong \widehat{\mathcal{B}}$, if there is an isomorphism from $G(\mathcal{B})$ to $\widehat{G}(\widehat{\mathcal{B}})$ mapping u^i to $\widehat{u}^i$ for $i \in \{0, 1, 2\}$.

Claim 1. There is a formula $\mathsf{nl\text{-}iso}(x^0, x^1, x^2) \in \mathsf{C}^6_\mathsf{w}$ (not depending on $\widehat{G}$) such that $\widehat{G} \models \mathsf{nl\text{-}iso}(\widehat{u}^0, \widehat{u}^1, \widehat{u}^2)$ if and only if $\widehat{\mathcal{B}} := (\widehat{u}^0, \widehat{\mathcal{Q}}^0, \widehat{u}^1, \widehat{\mathcal{Q}}^1, \widehat{u}^2, \widehat{\mathcal{Q}}^2)$ is a pseudo-necklace with $\widehat{\mathcal{B}} \cong \mathcal{B}$.

Proof of the claim: $\widehat{\mathcal{B}}$ is a pseudo-necklace isomorphic to $\mathcal{B}$ if and only if for all $i \in \{0,1,2\}$, the following two conditions hold for $\widehat{\mathcal{Q}}^i := \mathcal{Q}(\widehat{u}^i, \widehat{u}^{i+1})$.

1. $\widehat{G}(\widehat{\mathcal{Q}}^i) \cong G(\mathcal{Q}^i)$ via an isomorphism mapping $\widehat{u}^i$ to u^i and $\widehat{u}^{i+1}$ to u^{i+1}.
2. $V(\widehat{\mathcal{Q}}^{i-1}) \cap V(\widehat{\mathcal{Q}}^{i+1}) = \{\widehat{u}^i\}$.

Condition 2 is easy to express in C^6_w. To treat Condition 1, let $\mathsf{sps\text{-}iso}'_i \in \mathsf{C}^4_\mathsf{w}$ be the sentence from Theorem 9.17, which identifies the planar coloured graph $\mathcal{Q}^i_{u^i,u^{i+1}} := G(\mathcal{Q}^i)_{(u^i,u^{i+1})}$. Let R^i and R^{i+1} be the relations representing the unique colours of u^i and u^{i+1} in $\mathcal{Q}^i_{u^i,u^{i+1}}$. We transform $\mathsf{sps\text{-}iso}'_i$ into a formula $\mathsf{sps\text{-}iso}_i(x^i, x^{i+1})$ such that $\widehat{G} \models \mathsf{sps\text{-}iso}_i(\widehat{u}^i, \widehat{u}^{i+1})$ if and only if $\widehat{G}(\widehat{\mathcal{Q}}^i) \cong G(\mathcal{Q}^i)$ via an isomorphism that maps $\widehat{u}^i$ to u^i and $\widehat{u}^{i+1}$ to u^{i+1}. To this end, we first replace in $\mathsf{sps\text{-}iso}'_i$ every $R^i(z,z)$ with the formula $z = x^i$ and every $R^{i+1}(z,z)$ with $z = x^{i+1}$. To relativise $\mathsf{sps\text{-}iso}'_i$ to the i-th shortest path system, we also replace subformulae of the form $\exists y \varphi$ with $\exists y(\mathsf{csps\text{-}vert}(x^i, x^{i+1}, y) \wedge \varphi)$ and $E(y_1, y_2)$ with $\mathsf{csps\text{-}edge}(x^i, x^{i+1}, y_1, y_2)$. By Lemma 9.31, the resulting formula $\mathsf{sps\text{-}iso}_i(x^i, x^{i+1})$ is in C^6_w.

Now we can define the desired C^6_w-formula

$$\mathsf{nl\text{-}iso}(x^0,x^1,x^2) := \bigwedge_{i=0}^{2} \mathsf{sps\text{-}iso}_i(x^i,x^{i+1}) \wedge \forall y \bigwedge_{i=0}^{2} \Big(\big(\mathsf{csps\text{-}vert}(x^{i-1},x^i,y) \wedge \mathsf{csps\text{-}vert}(x^i,x^{i+1},y)\big) \to y = x^i \Big),$$

where we take indices modulo 3. ⌟

For the remainder of this proof, let $\widehat{\mathcal{B}} := (\widehat{u}^0, \widehat{\mathcal{Q}}^0, \widehat{u}^1, \widehat{\mathcal{Q}}^1, \widehat{u}^2, \widehat{\mathcal{Q}}^2)$ be a pseudo-necklace in $\widehat{G}$ such that $\mathcal{B} \cong \widehat{\mathcal{B}}$. If no such pseudo-necklace exists, we can distinguish G and $\widehat{G}$ in C^6_w using Claim 1. Let $u^i = u^i_1, u^i_2, \ldots, u^i_{n_i}$ be the articulation vertices of $\mathcal{Q}^i$ ordered by increasing height in $\mathcal{B}$. Since $\widehat{\mathcal{B}} \cong \mathcal{B}$, there is a bijection from $\operatorname{art}(\mathcal{B})$ to $\operatorname{art}(\widehat{\mathcal{B}})$ mapping each articulation vertex to one of equal height in $\widehat{\mathcal{B}}$. Thus, for simplicity, we use the same name for the two articulation vertices in $\mathcal{B}$ and $\widehat{\mathcal{B}}$ of equal height. In the following, let $\mathcal{Q}_{i,j} := \mathcal{Q}^G(u^i_j, u^i_{j+1})$ and $\widehat{\mathcal{Q}}_{i,j} := \mathcal{Q}^{\widehat{G}}(u^i_j, u^i_{j+1})$. Note that the $\widehat{\mathcal{Q}}_{i,j}$ are pseudo-patches. By our assumption that $\widehat{\mathcal{B}} \cong \mathcal{B}$, the pseudo-patch $\widehat{\mathcal{Q}}_{i,j}$ is trivial if and only if the patch $\mathcal{Q}_{i,j}$ is.

Let $I := I(\mathcal{B})$. Let $\widehat{I}$ be the graph with vertex set $V(\widehat{I}) := \mathsf{nl\text{-}int\text{-}vert}[\widehat{G}, \widehat{u}^0, \widehat{u}^1, \widehat{u}^2, y]$ and edge set $E(\widehat{I}) := \mathsf{nl\text{-}int\text{-}edge}[\widehat{G}, \widehat{u}^0, \widehat{u}^1, \widehat{u}^2, y_1, y_2]$. Note that, since $\widehat{\mathcal{B}}$ need not be a proper necklace (it might not comply with the third condition in Definition 9.12), the graph $\widehat{I}$ is not necessarily the inside of a necklace. For simplicity, we call I and $\widehat{I}$ *isomorphic*, and we write $I \cong \widehat{I}$, if $I_{(u^0,u^1,u^2)} \cong \widehat{I}_{(\widehat{u}^0,\widehat{u}^1,\widehat{u}^2)}$, i.e. if there is an isomorphism from I to $\widehat{I}$ mapping u^i to $\widehat{u}^i$ for every $i \in \{0,1,2\}$. Note that every such isomorphism induces an isomorphism from $\mathcal{B}$ to $\widehat{\mathcal{B}}$. We also define a "pseudo-inside" for all the pseudo-patches $\widehat{\mathcal{Q}}_{i,j}$: we let $I(\widehat{\mathcal{Q}}_{i,j})$ be the graph with vertex set $\mathsf{int\text{-}vert}[\widehat{G}, u^i_j, u^i_{j+1}, y]$ and edge set $\mathsf{int\text{-}edge}[\widehat{G}, u^i_j, u^i_{j+1}, y]$.

Claim 2. There is a formula $\mathsf{inside\text{-}iso}(x^0, x^1, x^2) \in \mathsf{C}^7_{\mathsf{w}}$ (not depending on $\widehat{G}$) such that $\widehat{G} \models \mathsf{inside\text{-}iso}(\widehat{u}^0, \widehat{u}^1, \widehat{u}^2)$ if and only if $\widehat{I} \cong I$.

Proof of the claim: We have that $\widehat{I} \cong I$ if and only if $\widehat{G}$ satisfies the following conditions for all $i \in \{0, 1, 2\}$.

(1a) $I(\widehat{\mathcal{Q}}_{i,j})_{(u^i_j, u^i_{j+1})} \cong I(\mathcal{Q}_{i,j})_{(u^i_j, u^i_{j+1})}$ for all $j \in [n_i - 1]$.

(1b) If $i' = i+1 \mod 3$ and $j = n_i-1$ and $j' = 1$, it holds that $I(\widehat{\mathcal{Q}}_{i,j}) \cap I(\widehat{\mathcal{Q}}_{i',j'}) = \{\widehat{u}^{i+1}\}$.

(1c) If $j \in [n_i - 1]$ and $j' = j + 1$, it holds that $I(\widehat{\mathcal{Q}}_{i,j}) \cap I(\widehat{\mathcal{Q}}_{i,j'}) = \{u^i_{j+1}\}$.

(1d) If $i' = i+1 \mod 3$ and ($j \neq n_i-1$ or $j' \neq 1$), it holds that $I(\widehat{\mathcal{Q}}_{i,j}) \cap I(\widehat{\mathcal{Q}}_{i',j'}) = \emptyset$. If $|j' - j| \geq 2$, it holds that $I(\widehat{\mathcal{Q}}_{i,j}) \cap I(\widehat{\mathcal{Q}}_{i,j'}) = \emptyset$.

The "only if" follows from the definition of I and the C-definability of $I(\mathcal{Q}_{i,j})$. We now show the "if"-part. Consider isomorphisms $\pi_{i,j}$ witnessing Part (1a). We define an isomorphism π from $\widehat{I}$ to I by letting $\pi(v)$ be $\pi_{i,j}(v)$ where i and j are such that $I(\widehat{\mathcal{Q}}_{i,j})$ contains v. Note that, by Parts (1b), (1c) and (1d), if $I(\widehat{\mathcal{Q}}_{i,j}) \cap I(\widehat{\mathcal{Q}}_{i',j'}) \neq \emptyset$ for $\widehat{\mathcal{Q}}_{i,j} \neq \widehat{\mathcal{Q}}_{i',j'}$, then there is a unique vertex $v \in I(\widehat{\mathcal{Q}}_{i,j}) \cap I(\widehat{\mathcal{Q}}_{i',j'})$. In that case, Part (1a) guarantees that the two possible images $\pi(v)$ coincide. Thus, π is well-defined and it certainly is an isomorphism.

We still need to translate Parts (1a)–(1d) into C-formulae. Since the subgraph $I(\mathcal{Q}_{i,j})$ of G is embedded in the disk $\boldsymbol{D}(\mathcal{Q}_{i,j})$, it is planar. Hence, by Theorem 9.17, there is a sentence $\mathsf{disk\text{-}iso}'_{i,j} \in \mathsf{C}^4$ which identifies $I(\mathcal{Q}_{i,j})_{(u^i_j, u^i_{j+1})}$. Let R^i_j and R^i_{j+1} be the relations that occur in $\mathsf{disk\text{-}iso}'_{i,j}$ for the colours of u^i_j and u^i_{j+1}, respectively.

To relativise $\mathsf{disk\text{-}iso}'_{i,j}$ to $I(\widehat{\mathcal{Q}}_{i,j})$, we replace every formula $\exists^{\geq p} x \varphi$ with the formula $\exists^{\geq p} x (\mathsf{int\text{-}vert}(x^i_j, x^i_{j+1}, x) \wedge \varphi)$ and every $E(x, y)$ with $\mathsf{int\text{-}edge}(x^i_j, x^i_{j+1}, x)$. Furthermore, we replace every $R^i_j(z, z)$ with the formula $z = x^i_j$ and every $R^i_{j+1}(z, z)$ with $z = x^i_{j+1}$. By Lemma 9.38, the resulting formula $\mathsf{disk\text{-}iso}_{i,j}(x^i_j, x^i_{j+1})$ is in $\mathsf{C}^7_{\mathsf{w}}$.

Again using Lemma 9.38, it is tedious but straightforward to construct a $\mathsf{C}^7_{\mathsf{w}}$-formula $\mathsf{disk\text{-}chain}(x^0, x^1, x^2)$ which checks if Parts (1b), (1c), and (1d) hold.

Now we can just let

$$\begin{aligned}\mathsf{inside\text{-}iso}(x^0, x^1, x^2) :=\ & \mathsf{nl\text{-}iso}(x^0, x^1, x^2) \wedge \mathsf{disk\text{-}chain}(x^0, x^1, x^2) \wedge \\ & \bigwedge_{i=0}^{2} \bigwedge_{j=1}^{n_i-1} \exists z \exists z' \big(\mathsf{csps\text{-}art}_j(x^i, x^{i+1}, z) \wedge \\ & \mathsf{csps\text{-}art}_{j+1}(x^i, x^{i+1}, z') \wedge \mathsf{disk\text{-}iso}_{i,j}(z, z')\big).\end{aligned}$$ ⌟

In the following, we assume without loss of generality that $\widehat{I} \cong I$.

Since $C(\mathcal{Q}_{i,j})$ is a cycle, the two sets of vertices of the segments on $C(\mathcal{Q}_{i,j})$ between u^i_j and u^i_{j+1} form blocks of the automorphism group of $C(\mathcal{Q}_{i,j})_{(u^i_j,u^i_{j+1})}$ and thus, by Corollary 9.11, also of the automorphism group of $G_{(u^i_j,u^i_{j+1})}$. To be more precise, every automorphism of G that fixes u^i_j and u^i_{j+1} either leaves each of the two segments invariant or "swaps sides", i.e. maps the two segments onto each other while preserving heights. Moreover, there is such an automorphism swapping sides in $I(\mathcal{Q}_{i,j})$ if and only if there is a vertex $v \in C(\mathcal{Q}_{i,j})$ with $v \notin \{u^i_j, u^i_{j+1}\}$ whose orbit of the automorphism group of $G_{(u^i_j,u^i_{j+1})}$ has size greater than 1. (In that case, it has size 2.)

Recall the definition of the cut graph $\mathrm{Cut}(\mathcal{B})$ from Definition 9.40. Also recall Lemma 9.43, which introduced two $\mathsf{C}^7_{\mathsf{w}}$-formulae $\mathsf{nl\text{-}cut\text{-}vert}(x^0, x^1, x^2, y)$ and $\mathsf{nl\text{-}cut\text{-}edge}(x^0, x^1, x^2, y_1, y_2)$ defining the vertex set and edge set of the cut graph. We define a *pseudo-cut graph* $\mathrm{Cut}(\widehat{\mathcal{B}})$ of $\widehat{\mathcal{B}}$ by letting

$$\begin{aligned} V\big(\mathrm{Cut}(\widehat{\mathcal{B}})\big) &:= \mathsf{nl\text{-}cut\text{-}vert}[\widehat{G}, \widehat{u}^0, \widehat{u}^1, \widehat{u}^2, y], \\ E\big(\mathrm{Cut}(\widehat{\mathcal{B}})\big) &:= \mathsf{nl\text{-}cut\text{-}edge}[\widehat{G}, \widehat{u}^0, \widehat{u}^1, \widehat{u}^2, y_1, y_2]. \end{aligned}$$

Let $C(\widehat{\mathcal{Q}}_{i,j})$ be the graph with vertex set $V\big(C(\widehat{\mathcal{Q}}_{i,j})\big) = \mathsf{bd\text{-}vert}[\widehat{G}, u^i_j, u^i_{j+1}, y]$ and edge set $E\big(C(\widehat{\mathcal{Q}}_{i,j})\big) = \mathsf{bd\text{-}edge}[\widehat{G}, u^i_j, u^i_{j+1}, y_1, y_2]$, where $\mathsf{bd\text{-}vert}(x, x', y)$ and $\mathsf{bd\text{-}edge}(x, x', y_1, y_2)$ are the $\mathsf{C}^7_{\mathsf{w}}$-formulae from Lemma 9.38. Also, let $I^*(\widehat{\mathcal{Q}}_{i,j})$ be the graph resulting from $I(\widehat{\mathcal{Q}}_{i,j})_{(u^i_j,u^i_{j+1})}$ by assigning to all vertices in $V\big(C(\widehat{\mathcal{Q}}_{i,j})\big)$ a common distinct colour and proceeding similarly with $E\big(C(\widehat{\mathcal{Q}}_{i,j})\big)$. Define the graph $I^*(\mathcal{Q}_{i,j})$ similarly.

Let $\mathrm{Cut}^*(\mathcal{B})$ be the (coloured) graph resulting from $\mathrm{Cut}(\mathcal{B})$ by adding colours corresponding to the following unary and binary relations:

(2a) for each set $J \subseteq [|\operatorname{art}(\mathcal{B})|]$ a relation R_J with $v \in R_J$ if and only if $J = \big\{i \mid \{v, w\} \in E(G) \text{ for a } w \in \mathsf{nl\text{-}art}_i[G, u^0, u^1, u^2, y]\big\}$, where $\mathsf{nl\text{-}art}_i(x^0, x^1, x^2, y)$ is the $\mathsf{C}^4_{\mathsf{w}}$-formula introduced in Corollary 9.33,

(2b) for each $i \in \{0, 1, 2\}$ and each $j \in [n_i - 1]$ a relation $R^1_{i,j}$ with $v \in R^1_{i,j}$ if and only if $v \in \mathsf{bd\text{-}vert}[G, u^i_j, u^i_{j+1}, y]$,

(2c) for each $i \in \{0, 1, 2\}$ and each $j \in [n_i - 1]$ a relation $R^2_{i,j}$ with $e \in R^2_{i,j}$ if and only if $e \in \mathsf{bd\text{-}edge}[G, u^i_j, u^i_{j+1}, y_1, y_2]$,

(2d) for every orbit O of the automorphism group of $I^*(\mathcal{Q}_{i,j})$ a relation R_O with $v \in R_O$ if and only if $v \in O$.

We show that all of the relations introduced in Parts (2a)–(2d) can be defined in $\mathsf{C}^{s+3}_{\mathsf{w}}$ by providing formulae that express containment in the relations. We omit the correctness proofs since they are straightforward.

(3a) For $R \coloneqq R_J$, let

$$\begin{aligned}\varphi_R(x^0,x^1,x^2,y) \coloneqq\ & \mathsf{nl\text{-}cut\text{-}vert}(x^0,x^1,x^2,y) \wedge \\ & \bigwedge_{i \in J} \exists z\big(\mathsf{nl\text{-}art}_i(x^0,x^1,x^2,z) \wedge E(y,z)\big) \wedge \\ & \bigwedge_{i \in [|\operatorname{art}(\mathcal{B})|] \setminus J} \neg\exists z\big(\mathsf{nl\text{-}art}_i(x^0,x^1,x^2,z) \wedge E(y,z)\big).\end{aligned}$$

(3b) For $R \coloneqq R^1_{i,j}$, let

$$\begin{aligned}\varphi_R(x^0,x^1,x^2,y) \coloneqq\ & \mathsf{nl\text{-}cut\text{-}vert}(x^0,x^1,x^2,y) \wedge \exists x^i_j \exists x^i_{j+1}\big(\mathsf{csps\text{-}art}_j(x^i,x^{i+1},x^i_j) \\ & \wedge \mathsf{csps\text{-}art}_{j+1}(x^i,x^{i+1},x^i_{j+1}) \wedge \mathsf{bd\text{-}vert}(x^i_j,x^i_{j+1},y)\big).\end{aligned}$$

(3c) For $R \coloneqq R^2_{i,j}$, let

$$\begin{aligned}\varphi_R(x^0,x^1,x^2,y_1,y_2) \coloneqq\ & \mathsf{nl\text{-}cut\text{-}edge}(x^0,x^1,x^2,y_1,y_2) \wedge \\ & \exists x^i_j \exists x^i_{j+1}\big(\mathsf{csps\text{-}art}_j(x^i,x^{i+1},x^i_j) \wedge \\ & \mathsf{csps\text{-}art}_{j+1}(x^i,x^{i+1},x^i_{j+1}) \wedge \mathsf{bd\text{-}edge}(x^i_j,x^i_{j+1},y_1,y_2)\big).\end{aligned}$$

(3d) Let $R \coloneqq R_O$. By Proposition 3.17 and the correspondence from Corollary 3.21, since C^4 identifies $I^*(\mathcal{Q}_{i,j})$ (and all vertex-coloured versions of it), the logic C^5 determines vertex orbits on $I^*(\mathcal{Q}_{i,j})$. Together with the identification, this implies that there is a $\mathsf{C}^5_{\mathsf{w}}$-formula $\varphi'_R(x)$ such that for every graph H, it holds that $H \models \varphi'_R(v)$ if and only if there is an isomorphism π^* from $I^*(\mathcal{Q}_{i,j})$ to H such that $\pi^*(w) = v$ for some $w \in O$.

We relativise $\varphi'_R(x)$ to $I(\mathcal{Q}_{i,j})$ by replacing every occurrence of the form $\exists^{\geq p} x \psi$ with

$$\exists^{\geq p} x \exists z \exists z' (\mathsf{csps\text{-}art}_j(x^i,x^{i+1},z) \wedge \mathsf{csps\text{-}art}_{j+1}(x^i,x^{i+1},z') \wedge \mathsf{int\text{-}vert}(z,z',x) \wedge \psi)$$

and proceeding similarly for the edges.

Let R^1_C and R^2_C be the colour relations corresponding to $V\big(C(\mathcal{Q}_{i,j})\big)$ and $E\big(C(\mathcal{Q}_{i,j})\big)$ in $I^*(\mathcal{Q}_{i,j})$, respectively. We replace $R^1_C(z,z)$ with

$$\exists y \exists y' (\mathsf{csps\text{-}art}_j(x^i,x^{i+1},y) \wedge \mathsf{csps\text{-}art}_{j+1}(x^i,x^{i+1},y') \wedge \mathsf{bd\text{-}vert}(y,y',z)$$

and proceed similarly with $R^2_C(z,z')$. Recall that R^i_j and R^i_{j+1} are the colour relations for u^i_j and u^i_{j+1}, respectively. We replace $R^i_j(z,z)$ with $\mathsf{csps\text{-}art}_j(x^i,x^{i+1},z)$ and do the analogous for $R^i_{j+1}(z,z)$. By Lemmas 9.31 and 9.38, the resulting formula $\varphi_R(x^0,x^1,x^2,x)$ is in $\mathsf{C}^7_{\mathsf{w}}$.

Our assumption $\widehat{\mathcal{B}} \cong \mathcal{B}$ implies that $|\operatorname{art}(\widehat{\mathcal{B}})| = |\operatorname{art}(\mathcal{B})|$. Define $\operatorname{Cut}^*(\widehat{\mathcal{B}})$ as the graph resulting from $\operatorname{Cut}(\widehat{\mathcal{B}})$ by interpreting each relation R from Parts (2a), (2b), (2d) as $\varphi_R[\widehat{G}, \widehat{u}^0, \widehat{u}^1, \widehat{u}^2, y]$ and each R from Part (2c) as $\varphi_R[\widehat{G}, \widehat{u}^0, \widehat{u}^1, \widehat{u}^2, y_1, y_2]$.

Claim 3. $\operatorname{Cut}^*(\mathcal{B}) \cong \operatorname{Cut}^*(\widehat{\mathcal{B}})$ if and only if $G_{u^0,u^1,u^2} \cong \widehat{G}_{\widehat{u}^0,\widehat{u}^1,\widehat{u}^2}$.

Proof of the claim: We begin by showing the backward direction of the claim. Assume that $G_{(u^0,u^1,u^2)} \cong \widehat{G}_{(\widehat{u}^0,\widehat{u}^1,\widehat{u}^2)}$, and let π be an isomorphism from G to $\widehat{G}$ mapping u^i to $\widehat{u}^i$. Since $\mathcal{B} \cong \widehat{\mathcal{B}}$, for all i, j, the isomorphism π maps $\mathcal{Q}_{i,j}$ to $\widehat{\mathcal{Q}}_{i,j}$. By Lemma 9.43, $V\big(\operatorname{Cut}(\mathcal{B})\big) = \mathsf{nl\text{-}cut\text{-}vert}[G, u^0, u^1, u^2, y]$ and $E\big(\operatorname{Cut}(\mathcal{B})\big) = \mathsf{nl\text{-}cut\text{-}edge}[G, u^0, u^1, u^2, y_1, y_2]$. Consequently, π must map $\operatorname{Cut}(\mathcal{B})$ to $\operatorname{Cut}(\widehat{\mathcal{B}})$. To see that π also induces an isomorphism between $\operatorname{Cut}^*(\mathcal{B})$ and $\operatorname{Cut}^*(\widehat{\mathcal{B}})$, consider a vertex $v \in V\big(\operatorname{Cut}^*(\mathcal{B})\big)$ and suppose $\pi(v)$ has a different colour (i.e. satisfies different colour relations) in $\operatorname{Cut}^*(\widehat{\mathcal{B}})$ than v in $\operatorname{Cut}^*(\mathcal{B})$.

Let R be one of the unary relations from Parts (2a)–(2d). Since π is an isomorphism which maps u^i to $\widehat{u}^i$ for $i \in \{0, 1, 2\}$, we have that

$$\begin{aligned} v \in R(G) &\iff v \in \varphi_R[G, u^0, u^1, u^2, y] \\ &\iff \pi(v) \in \varphi_R[\widehat{G}, \widehat{u}^0, \widehat{u}^1, \widehat{u}^2, y] \\ &\iff \pi(v) \in R(\widehat{G}). \end{aligned}$$

Similarly, we can show that every edge $e \in E\big(\operatorname{Cut}(\mathcal{B})\big)$ is mapped to an edge $\pi(e) \in E\big(\operatorname{Cut}(\widehat{\mathcal{B}})\big)$ contained in the same colour relations. Thus, π induces an isomorphism between $\operatorname{Cut}^*(\mathcal{B})$ and $\operatorname{Cut}^*(\widehat{\mathcal{B}})$, which concludes the backward direction of the proof.

For the forward direction, assume that $\operatorname{Cut}^*(\mathcal{B}) \cong \operatorname{Cut}^*(\widehat{\mathcal{B}})$ via an isomorphism π. Then since $|\operatorname{art}(\mathcal{B})| = |\operatorname{art}(\widehat{\mathcal{B}})|$, by Parts (2a) and (2b), the isomorphism π can be extended to an isomorphism π' from the graph with vertex set $V\big(\operatorname{Cut}(\mathcal{B})\big) \cup \operatorname{art}(\mathcal{B})$ whose edge set is the extension of $E\big(\operatorname{Cut}(\mathcal{B})\big)$ by all edges between $V\big(\operatorname{Cut}(\mathcal{B})\big) \cup \operatorname{art}(\mathcal{B})$ and $\operatorname{art}(\mathcal{B})$ to the corresponding subgraph of $\widehat{G}$, where π' maps every articulation vertex of $\mathcal{B}$ to one of equal height in $\widehat{\mathcal{B}}$.

Furthermore, by Parts (2b) and (2c), the mapping π induces an isomorphism from $C(\mathcal{Q}_{i,j})$ to $C(\widehat{\mathcal{Q}}_{i,j})$ which fixes u^i_j and u^i_{j+1} (and thus preserves heights). Let $L_{i,j}$ and $R_{i,j}$ as well as $\widehat{L}_{i,j}$ and $\widehat{R}_{i,j}$ be the two segments of $C(\mathcal{Q}_{i,j})$ and $C(\widehat{\mathcal{Q}}_{i,j})$ between u^i_j and u^i_{j+1}, respectively. Then for every pair (i, j), the isomorphism π either maps $L_{i,j}$ to $\widehat{L}_{i,j}$ and $R_{i,j}$ to $\widehat{R}_{i,j}$, or $L_{i,j}$ to $\widehat{R}_{i,j}$ and $R_{i,j}$ to $\widehat{L}_{i,j}$. Without loss of generality, assume the first.

If for every pair (i, j), the coloured graphs $I^*(\mathcal{Q}_{i,j})$ and $I^*(\widehat{\mathcal{Q}}_{i,j})$ are isomorphic via an isomorphism $\pi'_{i,j}$ mapping $L_{i,j}$ to $\widehat{L}_{i,j}$, then the collection of the $\pi_{i,j}$ clearly extends π to an isomorphism between G and $\widehat{G}$.

Thus, suppose there is a pair (i, j) such that every isomorphism from $I^*(\mathcal{Q}_{i,j})$ to $I^*(\widehat{\mathcal{Q}}_{i,j})$ swaps sides. Let $v \in C(\mathcal{Q}_{i,j})$ and let O be the orbit of v with respect to the automorphism group of $I^*(\mathcal{Q}_{i,j})$. Let $R := R_O$. Then, by Part (3d), it holds that $v \in \varphi_R[G, u^0, u^1, u^2, x]$, but $\pi(v) \notin \varphi_R[\widehat{G}, \widehat{u}^0, \widehat{u}^1, \widehat{u}^2, x]$. However, this is a contradiction, since π must respect all relations. ⌟

Thus, to check whether $G_{(u^0,u^1,u^2)} \cong \widehat{G}_{(\widehat{u}^0,\widehat{u}^1,\widehat{u}^2)}$ holds, it suffices to consider the (pseudo-)cut graphs of G and $\widehat{G}$.

Claim 4. There is a formula $\mathsf{cut\text{-}iso}_G(x^0, x^1, x^2) \in \mathsf{C}_\mathsf{w}^{s+3}$ (not depending on $\widehat{G}$) such that $\widehat{G} \models \mathsf{cut\text{-}iso}(\widehat{u}^0, \widehat{u}^1, \widehat{u}^2)$ if and only if $\mathrm{Cut}^*(\widehat{\mathcal{B}}) \cong \mathrm{Cut}^*(\mathcal{B})$.

Proof of the claim: By Lemma 9.41, every connected component of $\mathrm{Cut}(\mathcal{B})$ is in $\mathcal{E}_{g-1}$. Therefore, by Corollary 9.21 and by the induction assumption, we know that there is a sentence $\mathsf{cut\text{-}iso}'_G \in \mathsf{C}^s$ which identifies the graph $\mathrm{Cut}^*(\mathcal{B})$. By replacing every subformula $\exists^{\geq p} x \varphi$ with the formula $\exists^{\geq p} x(\mathsf{nl\text{-}cut\text{-}vert}(x^0, x^1, x^2, x) \land \varphi)$ and $E(x, y)$ with $\mathsf{nl\text{-}cut\text{-}edge}(x^0, x^1, x^2, x, y)$, we relativise $\mathsf{cut\text{-}iso}'_G$ to the (pseudo-)cut graph. To transform it into a formula working also on the uncoloured cut graph, for every relation R from Parts (2a), (2b), (2d), we replace each occurrence $R(z, z)$ with $\varphi_R(x^0, x^1, x^2, z)$ and proceed analogously for every R from Part (2c).

The resulting formula $\mathsf{cut\text{-}iso}_G(x^0, x^1, x^2)$ is in $\mathsf{C}_\mathsf{w}^{s+3}$ and it holds that

$$\widehat{G} \models \mathsf{cut\text{-}iso}_G(\widehat{u}^0, \widehat{u}^1, \widehat{u}^2) \iff \mathrm{Cut}^*(\widehat{\mathcal{B}}) \cong \mathrm{Cut}^*(\mathcal{B}). \quad ⌟$$

Now the formula

$$\mathsf{iso}_G := \exists x^0 \exists x^1 \exists x^2 \big(\mathsf{inside\text{-}iso}(x^0, x^1, x^2) \land \mathsf{cut\text{-}iso}_G(x^0, x^1, x^2)\big) \in \mathsf{C}_\mathsf{w}^{s+3}$$

identifies G. □

9.4.2 Case 2: Presence of Simplifying Patches

In this section, we still assume that G is polyhedrally embedded in the surface $\boldsymbol{S}$ of Euler genus g and that $n = |G|$ (Assumption 9.37), but we replace Assumption 9.39 with the following assumption.

Assumption 9.44. *G contains a simplifying patch.*

This case sounds simpler than the first one: we only need to remove a simplifying patch from our graph. The remaining pieces have smaller Euler genus and can thus be identified in the logic C_w^s. So all we need to do is colour the pieces in such a way that we can reconstruct the original graph. The problem with this line of reasoning is that simplifying patches have a much more complicated structure than non-simplifying patches. For example, we cannot define the internal graph of a simplifying patch in the same way as we did for non-simplifying patches in Lemma 9.38, since there is not necessarily a non-planar connected component which marks the "outside region" of the patch. Therefore, the idea of the proof is to remove the canonical sps and some interior parts of the corresponding patch, which are actually internal graphs of non-simplifying subpatches and thus definable by Lemma 21.A consequence is that there is no easy way to reconstruct the original graph from the graph obtained by removing a simplifying patch.

The first lemma handles trivial simplifying patches, so that afterwards we can focus on non-trivial ones.

Lemma 9.45. *If G has a trivial simplifying patch, then there is a sentence $\mathsf{iso}_G \in \mathsf{C}_{\mathsf{w}}^{s+2}$ that identifies G.*

Proof: If G has a trivial simplifying patch consisting of an edge $\{u, u'\}$, then each connected component of $G - \{u, u'\}$ is in $\mathcal{E}_{g-1}$ and can be identified by a sentence in $\mathsf{C}_{\mathsf{w}}^{s}$. From these sentences, we can construct a sentence in $\mathsf{C}_{\mathsf{w}}^{s+2}$ identifying G (arguing as in the proof of Lemma 9.27). □

From now on, we make the following assumption.

Assumption 9.46. *G contains no trivial simplifying patch.*

Recall the definition of a *segment* $\mathcal{Q}[v, v']$ of an sps $\mathcal{Q}$ from Section 9.2. A *subpatch* of a patch $\mathcal{Q}$ is a segment of $\mathcal{Q}$ which is a patch, i. e., which has no proper articulation vertices. A patch $\mathcal{Q}$ is a *minimal simplifying patch* if $\mathcal{Q}$ is simplifying and all proper subpatches of $\mathcal{Q}$ are non-simplifying. We are mostly interested in minimal simplifying patches.

The *internal region* of an sps $\mathcal{Q}$ is the set

$$\boldsymbol{R}(\mathcal{Q}) := \bigcup_{\boldsymbol{e} \in E(\mathcal{Q})} \boldsymbol{e} \quad \cup \bigcup_{\substack{\mathcal{Q}' \text{ non-trivial non-simplifying} \\ \text{subpatch of } \mathcal{Q}}} \boldsymbol{D}(\mathcal{Q}').$$

Note that if $\mathcal{Q}$ is a non-trivial patch, then $\boldsymbol{R}(\mathcal{Q}) \subseteq \boldsymbol{D}(\mathcal{Q})$, because $\boldsymbol{D}(\mathcal{Q}') \subseteq \boldsymbol{D}(\mathcal{Q})$ for every subpatch $\mathcal{Q}'$ of $\mathcal{Q}$. The *regional graph* of a patch $\mathcal{Q}$ is the graph $J := J(\mathcal{Q})$ with vertex set $V(J) := V(G) \cap \boldsymbol{R}(\mathcal{Q})$ and $E(J) := \{\boldsymbol{e} \in E(G) \mid \boldsymbol{e} \subseteq \boldsymbol{R}(\mathcal{Q})\}$. It follows from Lemma 9.10 that the graph J only depends on the abstract graph G and not on the embedding of G. Observe that if $\mathcal{Q}$ is a non-trivial and non-simplifying patch, then $\boldsymbol{R}(\mathcal{Q}) = \boldsymbol{D}(\mathcal{Q})$ and $J(\mathcal{Q}) = I(\mathcal{Q})$.

Our first lemma shows that the regional graph of a patch is definable in the logic $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$.

Lemma 9.47. *There are $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formulae*

$$\mathsf{J\text{-}vert}(x, x', y), \qquad \mathsf{J\text{-}edge}(x, x', y_1, y_2),$$

such that for all $u, u' \in V(G)$, the following holds. If the canonical sps $\mathcal{Q} := \mathcal{Q}^G(u,u')$ from u to u' is a patch, then for all $u, u' \in V(G)$,

$$\begin{aligned}\mathsf{J\text{-}vert}[G,u,u',y] &= V\big(J(\mathcal{Q})\big),\\ \mathsf{J\text{-}edge}[G,u,u',y_1,y_2] &= E\big(J(\mathcal{Q})\big).\end{aligned}$$

Proof: We first define a $\mathsf{C}^4_{\mathsf{w}}$-formula $\mathsf{csps\text{-}path}$ such that $G \models \mathsf{csps\text{-}path}(u,u',v,w)$ if and only if there is a path $Q \in \mathcal{Q}$ such that $v, w \in V(Q)$ and $\mathrm{ht}^{\mathcal{Q}}(v) \leq \mathrm{ht}^{\mathcal{Q}}(w)$. We set

$$\begin{aligned}\mathsf{csps\text{-}path}(x,x',z,z') := \mathsf{csps\text{-}vert}(x,x',z) \wedge \mathsf{csps\text{-}vert}(x,x',z') \wedge \bigvee_{i=0}^{n-1}\Bigg(\mathsf{dist}_{=i}(x,x') \wedge\\ \bigvee_{j\leq i}\Big(\mathsf{dist}_{=j}(x,z) \wedge \bigvee_{k\leq i-j}\big(\mathsf{dist}_{=k}(z,z') \wedge \mathsf{dist}_{=i-j-k}(z',x')\big)\Big)\Bigg).\end{aligned}$$

Now we can let

$$\begin{aligned}\mathsf{J\text{-}vert}(x,x',y) := {}& \mathsf{csps\text{-}vert}(x,x',y) \vee \exists z \exists z' \big(\mathsf{csps\text{-}path}(x,x',z,z') \wedge\\ & \neg\mathsf{csps\text{-}simplifying}(z,z') \wedge \neg E(z,z') \wedge \mathsf{int\text{-}vert}(z,z',y)\big).\end{aligned}$$

By Corollary 9.36 and Lemma 9.38, we have $\mathsf{J\text{-}vert} \in \mathsf{C}^{\max\{7,s+2\}}_{\mathsf{w}}$.

Similarly, we can define the $\mathsf{C}^{\max\{7,s+2\}}_{\mathsf{w}}$-formula

$$\begin{aligned}\mathsf{J\text{-}edge}(x,x',y_1,y_2) := {}& \mathsf{csps\text{-}edge}(x,x',y_1,y_2) \vee \exists z \exists z' \big(\mathsf{csps\text{-}path}(x,x',z,z') \wedge\\ & \neg\mathsf{csps\text{-}simplifying}(z,z') \wedge \mathsf{J\text{-}vert}(z,z',y_1) \wedge \mathsf{J\text{-}vert}(z,z',y_2) \wedge\\ & E(y_1,y_2)\big).\end{aligned}$$ □

For the remainder of this section, we fix a minimal non-trivial simplifying patch of G.

Assumption 9.48. *$\mathcal{Q} := \mathcal{Q}^G(u,u')$ is a minimal non-trivial simplifying patch of G. Furthermore, $\boldsymbol{D} := \boldsymbol{D}(\mathcal{Q})$, $\boldsymbol{R} := \boldsymbol{R}(\mathcal{Q})$, and $J := J(\mathcal{Q})$.*

By Lemma 9.47, the logic $\mathsf{C}^{\max\{7,s+2\}}$ distinguishes the regional graph J from the remainder of the graph. Furthermore, since $\mathcal{Q}$ is simplifying, every connected component of $G - J$ is contained in $\mathcal{E}_{g-1}$. We need to branch into two subcases once more.

Case 2.1: $G - J$ is connected

The proof in this case is similar to Case 1, but simpler. The key fact is that, in this case, we have $\boldsymbol{R} = \boldsymbol{D}$, by [64, Lemma 15.4.22(1)]. As remarked in Section 9.2 (after Definition 9.7), there are paths $Q, Q' \in \mathcal{Q}$ such that $C := Q \cup Q'$ is a cycle and

$\boldsymbol{bd}(\boldsymbol{D}) = \boldsymbol{C}$. It turns out that the cycle C only depends on u, u' and the abstract graph G (that is, we have analogues of Lemma 9.10 and Corollary 9.11). Indeed, our first step in this case will be to define the cycle C in the logic C^7_w.

Let $A_1, \ldots, A_m$ be the connected components of the graph $A^* := G - J$. Each A_i is embedded in $\boldsymbol{S} \setminus \boldsymbol{D}$, because each connected component of $G - V(\mathcal{Q})$ embedded in $\boldsymbol{D}$ belongs to J. Hence $N(A^*) \subseteq V(C)$. This implies that the graph G/A^* obtained from G by identifying all vertices in A^* is planar. Furthermore, by [64, Corollary 15.2.7], the graph G/A^* is 3-connected.

Lemma 9.49. *There are C^7_w-formulae*

$$\mathsf{bd\text{-}vert}(x, x', y), \qquad \mathsf{bd\text{-}edge}(x, x', y_1, y_2),$$

such that

$$\mathsf{bd\text{-}vert}[G, u, u', y] = V(C),$$
$$\mathsf{bd\text{-}edge}[G, u, u', y_1, y_2] = E(C).$$

Proof: The proof is completely analogous to the proof of Lemma 9.38, just redefining the formula astar:

$$\mathsf{astar}(x, x', y) := \neg\mathsf{J\text{-}vert}(x, x', y).$$

Note that in the proof of Lemma 9.38, we never use that A^* is connected. □

Now we fix one additional vertex: let u'' be the neighbour of u on the path Q. Let $k := \frac{|C|}{2} = |Q| - 1 = |Q'| - 1$. Using u'', we can enumerate the vertices of the cycle C in the cyclic order given by letting $u_0 := u$, $u_1 := u''$, then moving along Q to $u_k := u'$, and from there moving backwards along Q' to the neighbour u_{2k-1} of u on the path Q'.

Lemma 9.50. *For $i \in [0, 2k-1]$, there is a C^7_w-formula $\mathsf{bd\text{-}vert}_i(x, x', x'', y)$ such that*

$$\mathsf{bd\text{-}vert}_i[G, u, u', u'', y] = \{u_i\}.$$

Proof: We let $\mathsf{bd\text{-}vert}_0(x, x', x'', y) := (y = x)$, $\mathsf{bd\text{-}vert}_1(x, x', x'', y) := (y = x'')$, and

$$\mathsf{bd\text{-}vert}_i := \neg\mathsf{bd\text{-}vert}_{i-2}(x, x', x'', y) \wedge \exists y' \big(\mathsf{bd\text{-}vert}_{i-1}(x, x', x'', y) \wedge \mathsf{bd\text{-}edge}(x, x', y', y)\big)$$

for $i \in [2, 2k-1]$. □

We are ready to finish this subcase.

Proof of Lemma 9.29, Case 2.1: Let $\widehat{G}$ be an arbitrary graph. We shall prove that if there is no $\mathsf{C}^{s+3}_\mathsf{w}$-formula distinguishing G and $\widehat{G}$, then the two graphs are isomorphic.

So assume that there is no $\mathsf{C}_{\mathsf{w}}^{s+3}$-formula distinguishing G and $\widehat{G}$. Then there are vertices $\widehat{u}, \widehat{u}', \widehat{u}'' \in V(\widehat{G})$ such that, for all $\mathsf{C}_{\mathsf{w}}^{s+3}$-formulae $\varphi(x, x', x'')$, we have $G \models \varphi(u, u', u'') \iff \widehat{G} \models \varphi(\widehat{u}, \widehat{u}', \widehat{u}'')$. We fix such vertices $\widehat{u}, \widehat{u}', \widehat{u}''$. We shall prove that there is an isomorphism from G to $\widehat{G}$ mapping u to $\widehat{u}$, u' to $\widehat{u}'$, and u'' to $\widehat{u}''$.

Let $\widehat{J}$ be the subgraph of $\widehat{G}$ with vertex set $V(\widehat{J}) := \mathsf{J\text{-}vert}[\widehat{G}, \widehat{u}, \widehat{u}', y]$ and $E(\widehat{J}) := \mathsf{J\text{-}edge}[\widehat{G}, \widehat{u}, \widehat{u}', y_1, y_2]$. Similarly, let $\widehat{C}$ be the subgraph of $\widehat{G}$ with vertex set $V(\widehat{C}) := \mathsf{bd\text{-}vert}[\widehat{G}, \widehat{u}, \widehat{u}', y]$ and $E(\widehat{C}) := \mathsf{bd\text{-}edge}[\widehat{G}, \widehat{u}, \widehat{u}', y_1, y_2]$. Then $\widehat{C} \subseteq \widehat{J}$ and $\widehat{C}$ is a cycle. For every $i \in [0, 2k-1]$, let $\widehat{u}_i$ be the unique vertex such that $\widehat{G} \models \mathsf{bd\text{-}vert}_i(\widehat{u}, \widehat{u}', \widehat{u}'', \widehat{u}_i)$. Then $V(\widehat{C}) = \{u_0, \ldots, u_{2k-1}\}$, and the vertices $\widehat{u}_i$ appear on $\widehat{C}$ in cyclic ordering starting from $\widehat{u}_0 = \widehat{u}$ and $\widehat{u}_1 = \widehat{u}''$. Moreover, $\widehat{u}_k = \widehat{u}'$.

Now we individualise the vertices u_i in J and $\widehat{u}_i$ in $\widehat{J}$ using the same colour. More formally, for every i, we introduce a new colour relation R_i and define $R_i(J) := \{u_i\}$ and $R_i(\widehat{J}) := \{\widehat{u}_i\}$. We observe that the obtained coloured versions of J and $\widehat{J}$ satisfy the same $\mathsf{C}_{\mathsf{w}}^{4}$-sentences, because the vertices u_i and $\widehat{u}_i$ are defined in terms of u, u', u'' and $\widehat{u}, \widehat{u}', \widehat{u}''$, respectively, and for all $\mathsf{C}_{\mathsf{w}}^{7}$-formulae $\varphi(x, x', x'')$ we have $G \models \varphi(u, u', u'') \iff \widehat{G} \models \varphi(\widehat{u}, \widehat{u}', \widehat{u}'')$. Since J is planar and every planar graph is identified by a C^4-sentence, it follows that the coloured graphs are isomorphic. Hence, there is an isomorphism $f\colon V(J) \to V(\widehat{J})$ such that $f(u_i) = \widehat{u}_i$ for $i \in [0, 2k-1]$.

We shall extend f to an isomorphism from G to $\widehat{G}$. Let $\widehat{A}^* := \widehat{G} - \widehat{J}$. Note that $N^{\widehat{G}}(\widehat{A}^*) \subseteq V(\widehat{C})$, because G and thus also $\widehat{G}$ satisfies the $\mathsf{C}_{\mathsf{w}}^{7}$-formula

$$\forall y \forall z \big(\neg \mathsf{J\text{-}vert}(x, x', y) \wedge \mathsf{J\text{-}vert}(x, x', z) \wedge E(y, z) \to \mathsf{bd\text{-}vert}(x, x', z)\big).$$

We colour the graphs $A^* = G - J$ and $\widehat{A}^*$ using new colour relations R_I for $I \subseteq [0, 2k-1]$. We let $R_I(G)$ be the set of all $v \in V(A^*)$ such that $N^G(v) = \{u_i \mid i \in I\}$, and, similarly, we let $R_I(G)$ be the set of all $\widehat{v} \in V(\widehat{A}^*)$ such that $N^{\widehat{G}}(\widehat{v}) = \{\widehat{u}_i \mid i \in I\}$. Observe that there is a $\mathsf{C}_{\mathsf{w}}^{7}$-formula $\xi_I(x, x', x'', y)$ such that $\xi_I[G, u, u', u'', y] = R_I(G)$ and $\xi_I[\widehat{G}, \widehat{u}, \widehat{u}', \widehat{u}'', y] = R_I(\widehat{G})$. Thus, the coloured graphs A^* and $\widehat{A}^*$ satisfy the same $\mathsf{C}_{\mathsf{w}}^{s}$-sentences, because for all $\mathsf{C}_{\mathsf{w}}^{s+3}$-formulae $\varphi(x, x', x'')$, we have $G \models \varphi(u, u', u'') \iff \widehat{G} \models \varphi(\widehat{u}, \widehat{u}', \widehat{u}'')$,

As $\mathcal{Q}$ is simplifying, all connected components of A^* are in $\mathcal{E}_{g-1}$. Hence, by Assumption 9.25 and Corollary 9.21, there is a $\mathsf{C}_{\mathsf{w}}^{s}$-sentence iso_{A^*} that identifies A^*. Since A^* and $\widehat{A}^*$ satisfy the same $\mathsf{C}_{\mathsf{w}}^{s}$-sentences, there is an isomorphism g from A^* to $\widehat{A}^*$. The colour relations R_I guarantee that $f \cup g$ is an isomorphism from G to $\widehat{G}$. □

Case 2.2: $G - J$ is disconnected

In this case, we need to analyse the structure of the graph J in more detail. Let $\mathring{H}_1, \ldots, \mathring{H}_\ell$ be the connected components of $J - \{u, u'\}$. By the assumption of this

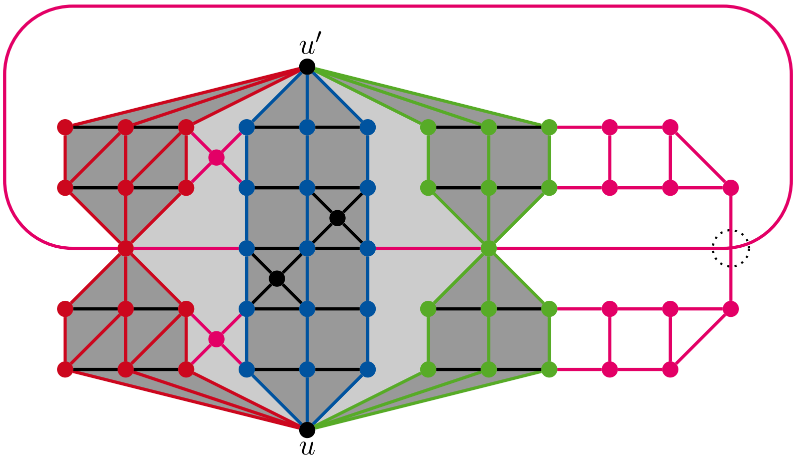


Figure 9.4: A simplifying patch in a graph of orientable genus 1.

case, we have $\ell \geq 2$. For every $i \in [\ell]$, let $H_i := J[V(\mathring{H}_i) \cup \{u, u'\}]$, and let $\mathcal{Q}_i$ be the set of all paths $Q \in \mathcal{Q}$ such that $Q \subseteq H_i$. Then $\mathcal{Q}_i$ is a shortest path system from u to u'. We call the $\mathcal{Q}_i$ the *fibres* of $\mathcal{Q}$. Note that $\boldsymbol{H_i} \subseteq \boldsymbol{D}$ for all i. Let $\boldsymbol{R}_i := \boldsymbol{R}(\mathcal{Q}_i)$. Then $\boldsymbol{R} = \bigcup_{i=1}^{\ell} \boldsymbol{R}_i$ and $\boldsymbol{R}_i \cap \boldsymbol{R}_j = \{u, u'\}$ for $i \neq j$. Let $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{\ell'}$ be the arcwise connected components of $\boldsymbol{D} \setminus \boldsymbol{R}$. By [64, Lemma 15.4.22(2)], we have $\ell' = \ell - 1$ and there is a permutation $\pi \in S_\ell$ such that $\boldsymbol{bd}(\boldsymbol{f}_i) \subseteq \boldsymbol{R}_{\pi(i)} \cup \boldsymbol{R}_{\pi(i+1)}$. Without loss of generality, we assume that π is the identity, that is, $\boldsymbol{bd}(\boldsymbol{f}_i) \subseteq \boldsymbol{R}_i \cup \boldsymbol{R}_{i+1}$. It is not hard to see (and shown in the proof of [64, Lemma 15.4.22]) that there are paths $Q'_i \in \mathcal{Q}_i$ and $Q_{i+1} \in \mathcal{Q}_{i+1}$ such that $C_i := Q'_i \cup Q_{i+1}$ is a cycle and $\boldsymbol{bd}(\boldsymbol{f}_i) = \boldsymbol{C}_i$. Let $\boldsymbol{f}_\ell := \boldsymbol{S} \setminus \boldsymbol{D}$. Then there are paths $Q_1 \in \mathcal{Q}_1$ and $Q'_\ell \in \mathcal{Q}_\ell$ such that $C_\ell := Q'_\ell \cup Q_1$ is a cycle and $\boldsymbol{C}_\ell = \boldsymbol{bd}(\boldsymbol{D}) = \boldsymbol{bd}(\boldsymbol{f}_\ell)$. For every i, we have $\boldsymbol{bd}(\boldsymbol{R}_i) = \boldsymbol{Q}_i \cup \boldsymbol{Q}'_i$. But note that $Q_i \cup Q'_i$ is not necessarily a cycle.

We use the notation introduced in this section so far (that is, $\boldsymbol{D}$, $\boldsymbol{R}$, $\boldsymbol{f}_i$, J, H_i, $\mathring{H}_i$, $\mathcal{Q}_i$) throughout the remainder of this subsection. Moreover, we always use indices modulo ℓ. For example, $H_{\ell+1}$ refers to H_1.

Example 9.51. *Consider the graph shown in Figure 9.4. The graph can be embedded into a torus in such a way that the red, blue, and green paths form a simplifying patch $\mathcal{Q} := \mathcal{Q}^G(u, u')$. The disk $\boldsymbol{D}(\mathcal{Q})$ is shown in grey, the region $\boldsymbol{R}(\mathcal{Q})$ within $\boldsymbol{D}(\mathcal{Q})$ in a darker grey. The patch has three fibres $\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3$ shown in red, blue, green, respectively. The boundary cycle of $\boldsymbol{D}(\mathcal{Q})$ consists of the leftmost red path and the rightmost green path from u to u'. The two areas in light grey are $\boldsymbol{f}_1$ (between red and blue) and $\boldsymbol{f}_2$ (between blue and green). The regional graph J consists of the red, blue, and green paths and all black edges and vertices.*

Observe that the graph has a second, different embedding into the torus in which $\mathcal{Q}$ is still a patch, but the boundary of its disk consists of a green and a blue path (and *therefore our numbering of the fibres would be different; the red fibre would be in the middle).*

Let $H \subseteq G$. For a connected component A of $G - H$, the vertices in $N(A) \subseteq V(H)$ are *vertices of attachment* of A. An *H-bridge* is a subgraph $B \subseteq \big(V(G), E(G) \setminus E(H)\big)$ such that either $B = \big(\{u,v\},\{uv\}\big)$ for some edge $\{u,v\} \in E(G)\setminus E(H)$ with $u, v \in V(H)$ or B is the union of a connected component A of $G - H$ together with all its vertices of attachment and all edges with at least one endvertex in $V(A)$. The set of *vertices of attachment* of an H-bridge B is the set $\text{at}(B) \coloneqq V(B) \cap V(H)$.

Informally speaking, for the regional graph J, a J-bridge is a connection between vertices in $V(H)$ that, in $\boldsymbol{G}$, reaches into the topological region which contains the embedding of $G - J$.

Let $B_1, \dots, B_m$ be a list of all J-bridges in G. If $|B_j| \geq 3$, let A_j be the connected component of $G - \boldsymbol{R}$ associated with B_j. If B_j is just a single edge, let A_j be the empty graph. In this case, we call B_j *trivial.* Note that for each $j \in [m]$, there is an $i \coloneqq i(j) \in [\ell]$ such that B_j is embedded in $\boldsymbol{f}_i$, or more precisely, in $\boldsymbol{cl}(\boldsymbol{f}_i)$. We say that$B_j$ is *attached to H_i* if it has a vertex of attachment in $V(H_i)$. We say that B_j *connects* H_i and $H_{i'}$ if it is attached to both H_i and $H_{i'}$. If B_j is attached to H_i, then it is embedded in $\boldsymbol{f}_i$ or in $\boldsymbol{f}_{i-1}$ (indices modulo m). Thus, if B_j connects H_i and $H_{i'}$, then either $i' = i + 1$ and B_j is embedded in $\boldsymbol{f}_i$, or $i' = i - 1$ and B_j is embedded in $\boldsymbol{f}_{i-1}$.

Example 9.52. *Consider again the graph shown in Figure 9.4 with the simplifying patch $\mathcal{Q}$ as in Example 9.51. The graph J (consisting of all red, green, blue, and black vertices and edges) has six bridges, all shown in pink. Three of these bridges are trivial. Note that in this example, all six bridges are planar; the non-planarity of the entire graph is a result of combining the bridges.* ⌟

Observe that there is at most one $i \in [\ell]$ such that there is no J-bridge connecting H_i and H_{i+1}. To see this, towards a contradiction, suppose that there are i and i' with $i < i'$ such that there is no bridge connecting H_i and H_{i+1} and no bridge connecting $H_{i'}$ and $H_{i'+1}$. Then $\{u, u'\}$ separates $\mathring{H}_i$ from $\mathring{H}_{i+1}$, which is impossible, since G is 3-connected. If there is no bridge connecting H_i and H_{i+1}, then we call $\mathcal{Q}_i$ and $\mathcal{Q}_{i+1}$ *dangling fibres.*

We say that two fibres $\mathcal{Q}_i$ and $\mathcal{Q}_{i'}$ are *adjacent* if $|i - i'| = 1$ or $\{i, i'\} = \{1, \ell\}$. Note that $\mathcal{Q}_i, \mathcal{Q}_{i'}$ are adjacent if $i \neq i'$ and either there is a J-bridge that connects H_i and $H_{i'}$ or both $\mathcal{Q}_i$ and $\mathcal{Q}_{i'}$ are dangling fibres. This means that we can detect the cyclic adjacency structure on the fibres $\mathcal{Q}_i$ just by looking at the bridges connecting them. It follows that the cyclic ordering of the fibres only depends on the abstract graph G and not on its embedding.

Lemma 9.53. *There are $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formulae*

$$\textsf{same-H}(x, x', y_1, y_2), \qquad \textsf{bconn-H}(x, x', y_1, y_2), \qquad \textsf{adj-H}(x, x', y_1, y_2),$$

such that for all $w_1, w_2 \in V(G)$, we have:

$$\begin{aligned} G \models \mathsf{same\text{-}H}(u,u',w_1,w_2) &\iff \textit{there is an } i \in [\ell] \textit{ such that } w_1, w_2 \in V(H_i),\\ G \models \mathsf{bconn\text{-}H}(u,u',w_1,w_2) &\iff \textit{there are distinct } i,i' \in [\ell] \textit{ and } j \in [m] \textit{ such that}\\ &\qquad w_1 \in V(\mathring{H}_i) \textit{ and } w_2 \in V(\mathring{H}_{i'}) \textit{ and } B_j \textit{ connects}\\ &\qquad \mathcal{Q}_i \textit{ and } \mathcal{Q}_{i'},\\ G \models \mathsf{adj\text{-}H}(u,u',w_1,w_2) &\iff \textit{there is an } i \in [\ell] \textit{ such that } w_1 \in V(\mathring{H}_i) \textit{ and}\\ &\qquad w_2 \in V(\mathring{H}_{i-1}) \cup V(\mathring{H}_{i+1}).\end{aligned}$$

Proof: By definition of H_i, there is an $i \in [\ell]$ such that $w_1, w_2 \in V(H_i)$ if and only if $w_1, w_2 \in V(J)$ and either $\{w_1, w_2\} \cap \{u, u'\} \neq \emptyset$ or w_1 and w_2 belong to the same $\mathring{H}_i$. Thus, we can let

$$\begin{aligned}\mathsf{same\text{-}H}(x,x',y_1,y_2) \coloneqq{}& \mathsf{J\text{-}vert}(x,x',y_1) \wedge \mathsf{J\text{-}vert}(x,x',y_2) \wedge\\ &\big(y_1 = x \vee y_1 = x' \vee y_2 = x \vee y_2 = x' \vee \varphi(x,x',y_1,y_2)\big),\end{aligned}$$

where $\varphi(x,x',y_1,y_2)$ is a $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formula stating that y_1, y_2 belong to the same connected component of $J - \{u, u'\}$. Using $\mathsf{J\text{-}vert}(x,x',y)$ and $\mathsf{J\text{-}edge}(x,x',y_1,y_2)$ as building blocks, it is easy to define such a formula.

There is a J-bridge that connects fibres $\mathcal{Q}_i$ and $\mathcal{Q}_{i'}$ if and only if there is a path $P \subseteq G - \{u,u'\}$ from a vertex $w_i \in V(\mathring{H}_i)$ to a vertex $w_{i'} \in V(\mathring{H}_{i'})$ with all internal vertices in $G - J$. Let $\psi(x,x',z_1,z_2)$ be a $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formula such that $G \models \psi(u,u',w_1,w_2)$ if and only $w_1, w_2 \in V(J) \setminus \{x, x'\}$ and there is a path from w_1 to w_2 with all internal vertices in $V(G - J)$. We can easily construct such a formula using $\mathsf{J\text{-}vert}(x,x',y)$ and $\mathsf{J\text{-}edge}(x,x',y_1,y_2)$ as building blocks. Now we let

$$\begin{aligned}\mathsf{bconn\text{-}H}(x,x',y_1,y_2) \coloneqq{}& \mathsf{J\text{-}vert}(x,x',y_1) \wedge \mathsf{J\text{-}vert}(x,x',y_2) \wedge \neg\mathsf{same\text{-}H}(x,x',y_1,y_2)\\ &\wedge \exists z_1 \exists z_2 \left(\bigwedge_{i=1}^{2} \mathsf{same\text{-}H}(x,x',y_i,z_i) \wedge \psi(x,x',z_1,z_2)\right).\end{aligned}$$

Recall that two fibres are adjacent if and only if either there is a J-bridge that connects them or both fibres are dangling. To define dangling fibres, we use the following formula:

$$\begin{aligned}\mathsf{dangling}(x,x',y) \coloneqq{}& \forall y' \forall y'' \big(\mathsf{bconn\text{-}H}(x,x',y,y') \wedge \mathsf{bconn\text{-}H}(x,x',y,y'')\\ &\to \mathsf{same\text{-}H}(x,x',y',y'')\big).\end{aligned}$$

Then $G \models \mathsf{dangling}(u,u',v)$ if and only if v belongs to a dangling fibre.

We let

$$\begin{aligned}\mathsf{adj\text{-}H}(x,x',y_1,y_2) \coloneqq{}& \mathsf{bconn\text{-}H}(x,x',y_1,y_2) \vee\\ &\left(\neg\mathsf{same\text{-}H}(x,x',y_1,y_2) \wedge \bigwedge_{i=1}^{2} \mathsf{dangling}(x,x',y_i)\right). \qquad \square\end{aligned}$$

Lemma 9.54. *There is a vertex $u'' \in V(J)$ and for every $i \in [\ell]$ a $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formula $\mathsf{H\text{-}vert}_i(x, x', x'', y)$ such that*

$$\mathsf{H\text{-}vert}_i[G, u, u', u'', y] = V(H_i).$$

Before we prove the lemma, let us remark that H_i is an induced subgraph of G. Therefore, there is no need for a formula $\mathsf{H\text{-}edge}$ defining $E(H_i)$.

Proof of Lemma 9.54: It will be easier to define formulae $\mathsf{H\text{-}vert}_i^\circ(x, x', x'', y)$ such that $\mathsf{H\text{-}vert}_i^\circ[G, u, u', u'', y] = V(\overset{\circ}{H}_i)$. Then we let

$$\mathsf{H\text{-}vert}_i(x, x', x'', y) := \mathsf{H\text{-}vert}_i^\circ(x, x', x'', y)' \vee y = x \vee y = x'.$$

We let

$$\mathsf{same\text{-}H}^\circ(x, x', y_1, y_2) := \mathsf{same\text{-}H}(x, x', y_1, y_2) \wedge \bigwedge_{i=1}^{2}(y_i \neq x \wedge y_i \neq x').$$

If $\ell = 2$, we choose an arbitrary $u'' \in V(\overset{\circ}{H}_1)$, and we let

$$\begin{aligned}\mathsf{H\text{-}vert}_1^\circ(x, x', x'', y) &:= \mathsf{same\text{-}H}^\circ(x, x', x'', y),\\ \mathsf{H\text{-}vert}_2^\circ(x, x', x'', y) &:= \mathsf{J\text{-}vert}(x, x', y) \wedge \neg\mathsf{same\text{-}H}(x, x', x'', y).\end{aligned}$$

In the following, we assume $\ell \geq 3$. If there are dangling fibres, we proceed as follows. Suppose $\mathcal{Q}_{i-1}$ and $\mathcal{Q}_i$ are dangling. We choose an arbitrary $u'' \in V(\overset{\circ}{H}_i)$. We let

$$\begin{aligned}\mathsf{H\text{-}vert}_i^\circ(x, x', x'', y) &:= \mathsf{same\text{-}H}^\circ(x, x', x'', y),\\ \mathsf{H\text{-}vert}_{i+1}^\circ(x, x', x'', y) &:= \mathsf{J\text{-}vert}(x, x', y) \wedge \mathsf{bconn\text{-}H}(x, x', x'', y),\end{aligned}$$

and for $j \in [2, \ell - 1]$

$$\begin{aligned}\mathsf{H\text{-}vert}_{i+j}^\circ(x, x', x'', y) := {}& \mathsf{J\text{-}vert}(x, x', y) \wedge \neg\mathsf{H\text{-}vert}_{i+j-2}^\circ(x, x', x'', y) \wedge \\ & \exists y'\big(\mathsf{H\text{-}vert}_{i+j-1}^\circ(x, x', x'', y') \wedge \mathsf{bconn\text{-}H}(x, x', y', y)\big).\end{aligned}$$

In the following, we assume that there are no dangling fibres. For $i \in [\ell]$, denote by $(i, i+1)$-*bridge* a J-bridge that connects $\mathcal{Q}_i$ and $\mathcal{Q}_{i+1}$. Since there is no dangling fibre, for all $i \in [\ell]$, there is at least one $(i, i+1)$-bridge.

Suppose that, for some i, there is a vertex $v \in V(\overset{\circ}{H}_i)$ that is a vertex of attachment of an $(i, i+1)$-bridge, but not of an $(i-1, i)$-bridge. Then we let $u'' := v$. As before, $\mathsf{H\text{-}vert}_i^\circ(x, x', x'', y) := \mathsf{same\text{-}H}^\circ(x, x', x'', y)$. To define $\mathsf{H\text{-}vert}_{i+1}^\circ(x, x', x'', y)$, we let $\psi(x, x', z_1, z_2)$ be a $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formula such that $G \models \psi(u, u', w_1, w_2)$ if and only $w_1, w_2 \in V(J) \setminus \{x, x'\}$ and there is a path from w_1 to w_2 with all internal vertices in $V(G - J)$. Then we let

$$\begin{aligned}\mathsf{H\text{-}vert}_{i+1}^\circ(x, x', x'', y) := {}& \neg\mathsf{H\text{-}vert}_i^\circ(x, x', x'', y) \wedge \\ & \exists y'\big(\psi(x, x', x'', y') \wedge \mathsf{same\text{-}H}^\circ(x, x', y, y')\big).\end{aligned}$$

For $j \in [2, \ell-1]$, we define $\mathsf{H\text{-}vert}^{\circ}_{i+j}(x, x', x'', y)$ as above:

$$\begin{aligned}\mathsf{H\text{-}vert}^{\circ}_{i+j}(x, x', x'', y) \coloneqq{}& \mathsf{J\text{-}vert}(x, x', y) \land \neg\mathsf{H\text{-}vert}^{\circ}_{i+j-2}(x, x', x'', y) \land \\ & \exists y' \big(\mathsf{H\text{-}vert}^{\circ}_{i+j-1}(x, x', x'', y') \land \mathsf{bconn\text{-}H}(x, x', y', y)\big).\end{aligned}$$

In the following, we assume that for every $i \in [\ell]$ and every $v \in V(\mathring{H}_i)$, either v is a vertex of attachment of both an $(i-1,i)$-bridge and an $(i,i+1)$-bridge (we call v *doubly-attached*) or it is neither a vertex of attachment of an $(i-1,i)$-bridge nor of an $(i,i+1)$-bridge (we call v *unattached*). Observe that if v is doubly-attached, then it is an articulation vertex of the sps $\mathcal{Q}_i$.

Claim 1. Let $i \in [2, \ell-1]$. Then no vertex $v \in V(\mathring{H}_i)$ is unattached.

Proof of the claim: Suppose, towards a contradiction, that $v \in V(\mathring{H}_i)$ is unattached. Suppose first that $v \in V(Q)$ for some path $Q \in \mathcal{Q}_i$ from u to u'. As we have no dangling fibres, there is at least one doubly-attached vertex in $V(\mathring{H}_i)$. Since all doubly-attached vertices are articulation vertices, every doubly-attached vertex in $V(\mathring{H}_i)$ appears on the path Q. Let w be the last doubly-attached vertex on Q before v, or, if no such vertex exists, let $w \coloneqq u$. Let w' be the first doubly-attached vertex on Q after v, or, if no such vertex exists, let $w' \coloneqq u'$. Then, by our assumption, we have $w \neq u$ or $w' \neq u'$. Say, $w' \neq u'$. Now $\{w, w'\}$ separates v from u'. This is an easy consequence of the fact that the graph that is the union of H_{i-1}, H_i, H_{i+1}, all $(i-1,i)$-bridges, all $(i,i+1)$-bridges, and all J-bridges that have all their vertices of attachment in H_i is embedded in the disk $\boldsymbol{D}$. However, this contradicts G being 3-connected.

It remains to consider the case that $v \in V(J) \setminus V(\mathcal{Q}_i)$. Then $v \in V\big(I(\mathcal{Q}')\big)$ for some non-trivial non-simplifying subpatch $\mathcal{Q}' = \mathcal{Q}_i[v_1, v_2]$ of $\mathcal{Q}_i$. Note that $\mathcal{Q}'$ contains no articulation vertices of $\mathcal{Q}_i$ except for (possibly) v_1 and v_2; otherwise it would not be a patch. Thus, every vertex $v' \in V(\mathcal{Q}') \setminus \{v_1, v_2\}$ is also unattached. We pick such a v'. It is contained in a path $Q \in \mathcal{Q}_i$. Therefore, we can apply the same argument as above to v' instead of v and again obtain a contradiction. ⌟

Let $\mathsf{unattached}(x, x', y)$ be a $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formula such that $G \models \mathsf{unattached}(u, u', v)$ if and only if $v \in V(J) \setminus \{u, u'\}$ and v is unattached. It is straightforward to construct such a formula. Suppose next that there is an unattached vertex v. Then $v \in V(\mathring{H}_1)$ or $v \in V(\mathring{H}_\ell)$. Say, $v \in V(\mathring{H}_\ell)$. Let u'' be an arbitrary vertex in $V(\mathring{H}_1)$. We let

$$\begin{aligned}\mathsf{H\text{-}vert}^{\circ}_{1}(x, x', x'', y) \coloneqq{}& \mathsf{same\text{-}H}^{\circ}(x, x', x'', y), \\ \mathsf{H\text{-}vert}^{\circ}_{\ell}(x, x', x'', y) \coloneqq{}& \mathsf{J\text{-}vert}(x, x', y) \land \neg\mathsf{H\text{-}vert}^{\circ}_{1}(x, x', x'', y) \land \\ & \exists y' \big(\mathsf{same\text{-}H}^{\circ}(x, x', y, y') \land \mathsf{unattached}(x, x', y')\big)\end{aligned}$$

and for $i \in [2, \ell-1]$

$$\begin{aligned}\mathsf{H\text{-}vert}^{\circ}_{i}(x, x', x'', y) \coloneqq{}& \mathsf{J\text{-}vert}(x, x', y) \land \neg\mathsf{H\text{-}vert}^{\circ}_{i-2}(x, x', x'', y) \land \\ & \exists y' \big(\mathsf{H\text{-}vert}^{\circ}_{i-1}(x, x', x'', y') \land \mathsf{bconn\text{-}H}(x, x', y', y)\big).\end{aligned}$$

In the following, we assume that there is no unattached vertex. This implies that every $\mathcal{Q}_i$ consists of just one path Q_i. For every i, let L_i be the graph that is the union of the paths Q_i and Q_{i+1} and all $(i, i+1)$-bridges. Observe that $L_1, \dots L_{\ell-1}$ are planar, because they are embedded in the disk $\boldsymbol{D}$. In fact, all these L_i have a planar embedding where $Q_i \cup Q_{i+1}$ is a facial cycle. Note that if L_ℓ also has such a planar embedding, then the graph $\bigcup_{i=1}^{\ell} L_\ell$ is planar.

Suppose L_ℓ does not have a planar embedding in which $Q_\ell \cup Q_1$ is a facial cycle. Using the fact that planarity is expressible in C^4, we can construct a $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formula $\mathsf{planar\text{-}L}(x, x', y, y')$ such that $G \models \mathsf{planar\text{-}L}(u, u', v, v')$ if and only if for some $i \in [\ell]$ the following conditions are satisfied:

- either $v \in V(\mathring{H}_i)$ and $v' \in V(\mathring{H}_{i+1})$, or $v' \in V(\mathring{H}_i)$ and $v \in V(\mathring{H}_{i+1})$;
- L_i has a planar embedding where $Q_i \cup Q_{i+1}$ is a facial cycle.

We choose $u'' \in V(\mathring{H}_1)$, and we let

$$\begin{aligned} \mathsf{H\text{-}vert}^\circ_1(x, x', x'', y) &:= \mathsf{same\text{-}H}^\circ(x, x', x'', y), \\ \mathsf{H\text{-}vert}^\circ_\ell(x, x', x'', y) &:= \mathsf{J\text{-}vert}(x, x', y) \wedge \exists y' \big(\mathsf{bconn}(x, x', y, y') \wedge \\ &\qquad \neg\mathsf{planar\text{-}L}(x, x', y, y') \wedge \mathsf{same\text{-}H}^\circ(x, x', x'', y')\big) \end{aligned}$$

and for $i \in [2, \ell - 1]$, as before,

$$\begin{aligned} \mathsf{H\text{-}vert}^\circ_i(x, x', x'', y) &:= \mathsf{J\text{-}vert}(x, x', y) \wedge \neg\mathsf{H\text{-}vert}^\circ_{i-2}(x, x', x'', y) \wedge \\ &\qquad \exists y' \big(\mathsf{H\text{-}vert}^\circ_{i-1}(x, x', x'', y') \wedge \mathsf{bconn\text{-}H}(x, x', y', y)\big). \end{aligned}$$

In the following, we assume that L_ℓ has a planar embedding where $Q_\ell \cup Q_1$ is a facial cycle. Then the graph $L := \bigcup_{i=1}^{\ell} L_\ell$ is planar. However, G is not planar. The graphs G and L differ only in the J-bridges that are attached to just a single fibre. Let us call a J-bridge whose vertices of attachment are in the fibre $\mathcal{Q}_i$ (and thus on the path Q_i) an *i-bridge*. Due to the 3-connectedness of G, every i-bridge has at least 3 vertices of attachment in $V(Q_i)$. Furthermore, since all vertices of Q_i are doubly-attached (i. e., they are vertices of attachment of both an $(i-1, i)$-bridge and an $(i, i+1)$-bridge), there is no way to attach an i-bridge for some $i \in [2, \ell - 1]$ without destroying the embedding in the disk $\boldsymbol{D}$. This is easy to see considering the fact that an i-bridge embedded in, say, $\boldsymbol{f}_i$ has two vertices v, v' of attachment of distance at least two in $\mathcal{Q}_i$ and it thus “blocks” the vertex between v and v' in $V(\mathcal{Q}_i)$ from being attached to a vertex in $\mathcal{Q}_{i+1}$ (cf. [64, Corollary 9.1.2]). Hence there can only be i-bridges for $i = 1$ and $i = \ell$, and there must be at least one such bridge, because otherwise $G = L$ would be planar. Say, there is an ℓ-bridge. We can easily construct a $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formula $\mathsf{self\text{-}bridge}(x, x', y)$ such that $G \models \mathsf{self\text{-}bridge}(u, u', v)$ if and only if $v \in V(\mathring{H}_i)$ for some i such that there is an i-bridge. We choose $u'' \in V(\mathring{H}_1)$ and let

$$\begin{aligned} \mathsf{H\text{-}vert}^\circ_1(x, x', x'', y) &:= \mathsf{same\text{-}H}^\circ(x, x', x'', y), \\ \mathsf{H\text{-}vert}^\circ_\ell(x, x', x'', y) &:= \mathsf{self\text{-}bridge}(x, x', y) \wedge \neg\mathsf{same\text{-}H}(x, x', x'', y) \end{aligned}$$

and for $i \in [2, \ell - 1]$, as before,

$$\begin{aligned}\mathsf{H\text{-}vert}^{\circ}_{i}(x, x', x'', y) \coloneqq{}& \mathsf{J\text{-}vert}(x, x', y) \wedge \neg\mathsf{H\text{-}vert}^{\circ}_{i-2}(x, x', x'', y) \wedge \\ & \exists y' \big(\mathsf{H\text{-}vert}^{\circ}_{i-1}(x, x', x'', y') \wedge \mathsf{bconn\text{-}H}(x, x', y', y)\big).\end{aligned}$$

This completes the proof. □

In the following, we fix a vertex u'' that is chosen according to Lemma 9.54.

Assumption 9.55. *$u'' \in V(J)$ is a fixed vertex such that $\mathsf{H\text{-}vert}_i[G, u, u', u'', y] = V(H_i)$ for every $i \in [\ell]$.*

A J-bridge B is an *inner bridge* if it has at least one vertex of attachment in $\bigcup_{i=2}^{\ell-1} V(\mathring{H}_i)$. Note that all inner bridges are embedded in the disk $\boldsymbol{D}$. Let K be the union of J with all inner bridges. Then K is a planar graph embedded in $\boldsymbol{D}$. Using Lemma 9.18 for $\varphi \coloneqq \mathsf{J\text{-}vert}(x, x', y)$, we can construct $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formulae that define membership in K.

Corollary 9.56. *There are $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formulae*

$$\mathsf{K\text{-}vert}(x, x', x'', y), \qquad \mathsf{K\text{-}edge}(x, x', x'', y_1, y_2)$$

such that

$$\begin{aligned}\mathsf{K\text{-}vert}[G, u, u', u'', y] &= V(K), \\ \mathsf{K\text{-}edge}[G, u, u', u'', y_1, y_2] &= E(K).\end{aligned}$$

Finally, we are ready to complete the proof of Lemma 9.29.

Proof of Lemma 9.29, Case 2.2: Let us briefly recall our main assumptions for this case:

- G is a graph of order $n = |G|$ polyhedrally embedded in a surface $\boldsymbol{S}$ of Euler genus $g \geq 1$.
- $\mathcal{Q} = \mathcal{Q}^G(u, u')$ is a non-trivial simplifying patch in G with $\ell \geq 2$ fibres.
- u'' is a vertex that allows to identify the fibres of $\mathcal{Q}$ via the formulae of Lemma 9.54.

We continue to use the notation introduced in this section, such as $\boldsymbol{D}$, J, $\mathcal{Q}_i$, H_i and $\mathring{H}_i$, etc.

Moreover, we define

$$h \coloneqq \operatorname{dist}(u, u').$$

Let $\widehat{G}$ be an arbitrary graph. We shall prove that if there is no $\mathsf{C}_{\mathsf{w}}^{s+3}$-formula distinguishing G and $\widehat{G}$, then the two graphs are isomorphic.

So assume that there is no $\mathsf{C}_\mathsf{w}^{s+3}$-formula distinguishing G and $\widehat{G}$. Then $|\widehat{G}| = n$ and $\widehat{G} \notin \mathcal{E}_{g-1}$. Furthermore, there are vertices $\widehat{u}, \widehat{u}', \widehat{u}'' \in V(\widehat{G})$ such that for all $\mathsf{C}_\mathsf{w}^{s+3}$-formulae $\varphi(x, x', x'')$ we have $G \models \varphi(u, u', u'') \iff \widehat{G} \models \varphi(\widehat{u}, \widehat{u}', \widehat{u}'')$. We fix such vertices $\widehat{u}, \widehat{u}', \widehat{u}''$. We shall prove that there is an isomorphism from G to $\widehat{G}$ mapping u to $\widehat{u}$, u' to $\widehat{u}'$, and u'' to $\widehat{u}''$.

Let $\widehat{\mathcal{Q}} := \mathcal{Q}^{\widehat{G}}(\widehat{u}, \widehat{u}')$. We say $\mathcal{Q}$ and $\widehat{\mathcal{Q}}$ are *isomorphic*, and write $\mathcal{Q} \cong \widehat{\mathcal{Q}}$, if there is an isomorphism from $G(\mathcal{Q})$ to $\widehat{G}(\widehat{\mathcal{Q}})$ mapping u to $\widehat{u}$ and u' to $\widehat{u}'$. Thus, in the following, we always regard u, u', u'' and the corresponding $\widehat{u}, \widehat{u}', \widehat{u}''$ as distinguished vertices that isomorphisms need to respect.

Just as in the proof of Claim 1 of Case 1, we have a formula $\mathsf{sps\text{-}iso}(x, x') \in \mathsf{C}_\mathsf{w}^6$ (not depending on $\widehat{G}$) such that $\widehat{G} \models \mathsf{sps\text{-}iso}(\widehat{u}, \widehat{u}')$ if and only if $\widehat{\mathcal{Q}}$ is a pseudo-patch with $\widehat{\mathcal{Q}} \cong \mathcal{Q}$.

Hence $\widehat{\mathcal{Q}}$ is a non-trivial pseudo-patch in $\widehat{G}$. Let $\widehat{J}$ be the graph with vertex set $V(\widehat{J}) := \mathsf{J\text{-}vert}[\widehat{G}, \widehat{u}, \widehat{u}', y]$ and edge set $E(\widehat{J}) := \mathsf{J\text{-}edge}[\widehat{G}, \widehat{u}, \widehat{u}', y_1, y_2]$. Note that $\widehat{u}'' \in V(\widehat{J})$, because $u'' \in V(J) = \mathsf{J\text{-}vert}[G, u, u', y]$. Therefore, we call J and $\widehat{J}$ *isomorphic*, and write $J \cong \widehat{J}$, if $J_{(u,u',u'')} \cong \widehat{J}_{(\widehat{u},\widehat{u}',\widehat{u}'')}$, that is, there is an isomorphism from J to $\widehat{J}$ that maps u to $\widehat{u}$, u' to $\widehat{u}'$, and u'' to $\widehat{u}''$. Note that every such isomorphism induces an isomorphism from $G(\mathcal{Q})$ to $G(\widehat{\mathcal{Q}})$.

Let $T \subseteq V(J)$ be the set of vertices of attachment of all J-bridges in G, and, similarly, let $\widehat{T}$ be the set of vertices of attachment of all $\widehat{J}$-bridges in $\widehat{G}$.

Claim 1. There is a formula $\mathsf{J\text{-}iso}(x, x', x'') \in \mathsf{C}_\mathsf{w}^{\max\{7,s+2\}}$ such that we have $\widehat{G} \models \mathsf{J\text{-}iso}(\widehat{u}, \widehat{u}', \widehat{u}'')$ if and only if $J \cong \widehat{J}$ via an isomorphism that maps T to $\widehat{T}$.

Proof of the claim: Let J^* be the graph resulting from $J_{(u,u',u'')}$ by assigning to all vertices in T a common distinct colour (however maintaining the individual colours for u, u', u''). Since J is planar, the claim follows by relativising a sentence $\mathsf{J\text{-}iso}' \in \mathsf{C}_\mathsf{w}^4$ which identifies J^* to the subgraph whose vertex and edge set is defined by the formula $\mathsf{J\text{-}vert}$ and $\mathsf{J\text{-}edge}$, respectively, and by replacing the colour relations for u, u', and u'' with equations of the form $z = x$, $z = x'$, $z = x''$ and the colour relation for T with $\exists z' \big(E(z, z') \wedge \neg \mathsf{J\text{-}edge}(x, x', z, z')\big)$. By Lemma 9.47, the formula has width $\max\{7, s + 2\}$. ⌟

In the following, we assume without loss of generality that $\widehat{J} \cong J$ and we only consider isomorphisms which preserve the property of being a vertex of attachment.

We intend to equip certain supergraphs of J and $\widehat{J}$ with colour relations such that the coloured graphs are isomorphic if and only if G and $\widehat{G}$ are isomorphic via an isomorphism mapping u to $\widehat{u}$, u' to $\widehat{u}'$, and u'' to $\widehat{u}''$. Then we show that the coloured supergraph of J can be identified in $\mathsf{C}_\mathsf{w}^{s+3}$.

First recall that, for every J-bridge, its vertices of attachment lie on a shortest path from u to u'. Thus, by Claim 1, we can assume the same for $\widehat{G}$, since we can express the sps-containment of a vertex of attachment. Hence, each element in $V(J)$

and $V(\widehat{J})$ which is a vertex of attachment of a bridge has a well-defined height in $\mathcal{Q}$ and $\widehat{\mathcal{Q}}$, respectively.

For every $i \in [\ell]$, we let $\widehat{H}_i$ be the induced subgraph of $\widehat{G}$ whose vertex set is $\mathsf{H\text{-}vert}_i[\widehat{G}, \widehat{u}, \widehat{u}', \widehat{u}'', y]$. Then $\widehat{J} = \bigcup_{i=1}^{\ell} \widehat{H}_i$, because there is a $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$-formula which expresses that $J = \bigcup_{i=1}^{\ell} H_i$.

We now colour vertices in $V(G-J)$ by their "attachment pattern" in J. For every $v \in V(G-J)$, we let

$$S(v) := \{\!\!\{ (i,j) \mid w \in N(v) \cap V(H_i),\ w \text{ has height } j \text{ in } \mathcal{Q} \}\!\!\}. \tag{9.7}$$

That is, for each $w \in N(v) \cap V(H_i)$ of height j, the set $S(v)$ contains one separate copy of (i,j). Similarly, for $\widehat{v} \in V(\widehat{G} - \widehat{J})$ we let

$$\widehat{S}(\widehat{v}) := \Big\{\!\!\Big\{ (i,j) \,\Big|\, \widehat{w} \in N(\widehat{v}) \cap V(\widehat{H}_i),\ \widehat{w} \text{ has height } j \text{ in } \widehat{\mathcal{Q}} \Big\}\!\!\Big\}.$$

Let A be a connected component of $G-J$. We view A as a coloured graph where (in addition to colours that may have already been present in G) each vertex v is coloured with the multiset $S(v)$. Since $\mathcal{Q}$ is simplifying, $\operatorname{eg}(A) \leq g-1$, and thus there is a $\mathsf{C}_{\mathsf{w}}^{s}$-sentence $\mathsf{bridge\text{-}iso}'_A$ that identifies A. We shall transform it into a $\mathsf{C}_{\mathsf{w}}^{s+3}$-formula $\mathsf{bridge\text{-}iso}_A(x, x', x'', y)$ such that $\widehat{G} \models \mathsf{bridge\text{-}iso}_A(\widehat{u}, \widehat{u}', \widehat{u}'', \widehat{v})$ if and only if A is isomorphic to the connected component of $\widehat{v}$ in $\widehat{G} - \widehat{J}$ via an isomorphism π that preserves the attachment patterns, that is, $S(v) = \widehat{S}\big(\pi(v)\big)$ for all $v \in V(A)$.

Note that if a bridge in G is attached to two vertices w and w' with the same label pair (i,j), then it must hold that $w = w'$. Thus, for every vertex v in G, the multiset $S(v)$ is actually a set, i.e. each tuple occurring in the multiset has multiplicity 1. However, in $\widehat{G}$, this might not be the case.

To relativise $\mathsf{bridge\text{-}iso}'_A$ to the connected component of a vertex v in $G-J$, we use the formula comp_φ from Lemma 9.18 for $\varphi := \mathsf{J\text{-}vert}(x, x', y)$ and replace every $\exists z \psi$ with $\exists z(\mathsf{comp}_{\mathsf{J\text{-}vert}}(x, x', y, z) \wedge \psi)$. Since $\mathsf{J\text{-}vert}(x, x', y)$ has width $\max\{7, s+2\}$, Lemma 9.18 yields that $\mathsf{comp}_{\mathsf{J\text{-}vert}}(x, x', y, z) \in \mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$.

To account for the colours, we define for each multiset S of label pairs (i,j) a relation R_S with $v \in R_S$ if and only if $S(v) = S$. Note that all label pairs that can occur are contained in $[m] \times [0,h]$. Let us denote the multiplicity of a pair (i,j) in a multiset S by $\operatorname{mult}_S(i,j)$. We let

$$\mathsf{att\text{-}pat}_S(x, x', x'', y) := \bigwedge_{\substack{(i,j) \in [m] \times [0,h] \\ \operatorname{mult}_S(i,j) = p}} \exists^{=p} z \big(\mathsf{H\text{-}vert}_i(x, x', x'', z) \wedge \mathsf{csps\text{-}height}_j(x, x', z) \wedge E(y,z)\big).$$

Then $G \models \mathsf{att\text{-}pat}_S(u, u', u'', v) \iff S(v) = S$. Note that, by Lemma 9.54, we have $\mathsf{att\text{-}pat}_S \in \mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$. We replace every $R_S(z,z)$ in $\mathsf{bridge\text{-}iso}'_A$ with

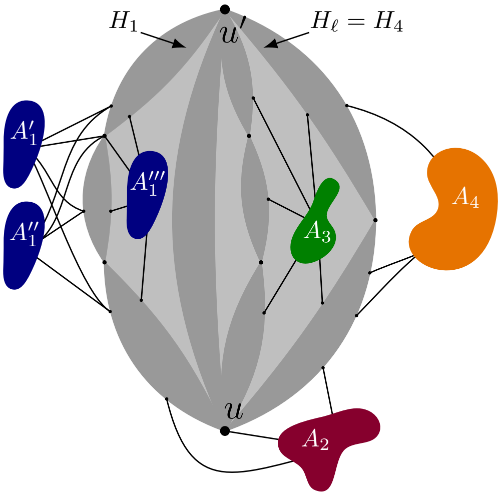


Figure 9.5: A simplifying patch with an inner bridge (green), multiple critical bridges (blue, orange, red) and a super-critical bridge (blue). Note that without A_1'', the graph would have an opposite pair $\{A_1', A_1'''\}$.

$\mathsf{att\text{-}pat}_S(x, x', x'', z)$ and obtain the desired formula $\mathsf{bridge\text{-}iso}_A(x, x', x'', y)$, which has width $s + 3$.

In the following, we only consider J-bridges and $\widehat{J}$-bridges. If the reference to J or $\widehat{J}$ is clear from the context, we often do not mention it explicitly and simply use the term “bridge”.

Recall that every J-bridge is either an *i-bridge* with all vertices of attachment in a single fibre H_i or an *$(i, i+1)$-bridge* with vertices of attachment in two adjacent fibres H_i and H_{i+1} for some $i \in [\ell]$.

Recall (from the paragraph preceding Corollary 9.56) that an *inner bridge* is a J-bridge which has at least one vertex of attachment in $\bigcup_{i=2}^{\ell-1} V(\mathring{H}_i)$ and that the graph K is the union of J with all inner bridges. K is a planar graph embedded in $\boldsymbol{D}$. By Corollary 9.56, we have $\mathsf{C}_{\mathsf{w}}^{s+3}$-formulae $\mathsf{K\text{-}vert}(x, x', x'', y)$ and $\mathsf{K\text{-}edge}(x, x', x'', y_1, y_2)$ such that $V(K) = \mathsf{K\text{-}vert}[G, u, u', u'', y]$ and $E(K) = \mathsf{K\text{-}edge}[G, u, u', u'', y_1, y_2]$. We let $\widehat{K}$ be the subgraph of $\widehat{G}$ with vertex set $V(\widehat{K}) = \mathsf{K\text{-}vert}[\widehat{G}, \widehat{u}, \widehat{u}', \widehat{u}'', y]$ and edge set $E(\widehat{K}) = \mathsf{K\text{-}edge}[\widehat{G}, \widehat{u}, \widehat{u}', \widehat{u}'', y_1, y_2]$.

A bridge is *critical* if it is not an inner bridge (see Figure 9.5). Observe that a bridge is critical if it is either an ℓ-bridge or a 1-bridge or an $(\ell, 1)$-bridge. Let $\mathcal{B}_{\text{crit}}$ denote the set of all critical J-bridges. Similarly, let $\widehat{\mathcal{B}}_{\text{crit}}$ be the set of all $\widehat{J}$-bridges in $\widehat{G}$ whose vertices of attachment are contained in $V(\widehat{\mathcal{Q}}_\ell) \cup V(\widehat{\mathcal{Q}}_1)$ (where $\widehat{\mathcal{Q}}_i$ is the set of all paths $\widehat{Q} \in \widehat{\mathcal{Q}}$ such that $\widehat{Q} \subseteq \widehat{H}_i$). With each J-bridge $B \in \mathcal{B}_{\text{crit}}$, we associate a *type* $\theta(B)$ as follows.

- If B consists of a single edge $\{v, v'\}$, then $\theta(B) = \big\{(i,j),(i',j')\big\}$, where $(i,j) = (0,0)$ if $v = u$, $(i,j) = (0,h)$ if $v = u'$, and otherwise $v \in V(\mathring{H}_i)$ and $j = \mathrm{dist}(u,v)$, and similarly $(i',j') = (0,0)$ if $v' = u$, $(i',j') = (0,h)$ if $v' = u'$, and otherwise $v' \in V(\mathring{H}_{i'})$ and $j' = \mathrm{dist}(u,v')$.
- If $|B| \geq 3$, let $A \coloneqq B - V(J)$ be the connected component of $G - J$ associated with B. We view A as a coloured graph (with colours representing the attachment patterns $S(v)$ as above) and choose a label θ_A for the isomorphism type of A (in such a way that $\theta_A = \theta_{A'} \iff A \cong A'$). We let $\theta(B) \coloneqq \{\theta_A\}$.

We can define the type $\widehat{\theta}(B)$ of a $\widehat{J}$-bridge $\widehat{B} \in \widehat{\mathcal{B}}_{\mathrm{crit}}$ similarly. Observe that there is a bijection $\beta\colon \mathcal{B}_{\mathrm{crit}} \to \widehat{\mathcal{B}}_{\mathrm{crit}}$ such that $\theta(B) = \widehat{\theta}\big(\beta(B)\big)$ for all $B \in \mathcal{B}_{\mathrm{crit}}$. To see this, note that we can use the formulae $\mathsf{bridge\text{-}iso}_A$ to construct for every type θ a $\mathsf{C}_{\mathsf{w}}^{s+3}$-formula that encodes the number of bridges of type θ.

Observe that if two critical bridges have the same type, then either both are $(\ell, 1)$-bridges or both are 1-bridges or both are ℓ-bridges.

Recall that $\mathrm{at}(B)$ denotes the set of vertices of attachment of a bridge B. Let us call bridges B, B' *aligned* if $\mathrm{at}(B) = \mathrm{at}(B')$. We show that being aligned can be defined in $\mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$. Let

$$\mathsf{att\text{-}vert}(x,x',y,z) \coloneqq \mathsf{J\text{-}vert}(x,x',z) \wedge \exists z'\big(E(z,z') \wedge \mathsf{comp}_{\mathsf{J\text{-}vert}}(x,x',z',y)\big).$$

Then if $v \notin J$, it holds that $G \models \mathsf{att\text{-}vert}(u,u',v,w)$ if and only if w is a vertex of attachment of some J-bridge that contains v. By Lemmas 9.18 and 9.47, the formula $\mathsf{att\text{-}vert}$ has width $\max\{7, s+2\}$. Now we can define

$$\begin{aligned}\mathsf{aligned}(x,x',y,y') \coloneqq{}& \neg\mathsf{comp}_{\mathsf{J\text{-}vert}}(x,x',y,y') \wedge \\ & \forall z\big(\mathsf{att\text{-}vert}(x,x',y,z) \leftrightarrow \mathsf{att\text{-}vert}(x,x',y',z)\big).\end{aligned}$$

Then if $v, v' \notin J$, it holds that $G \models \mathsf{aligned}(u,u',v,v')$ if and only if v and v' are contained in distinct J-bridges B and B', respectively, and B and B' are aligned. Furthermore, $\mathsf{aligned}$ has width $\max\{7, s+2\}$ since $\mathsf{att\text{-}vert}$ has width $\max\{7, s+2\}$. The two formulae can easily be modified to also capture the case that v or v' itself is a vertex of attachment (and the case of trivial bridges, but we do not need this for our purposes).

Recall that for all critical bridges $B \in \mathcal{B}_{\mathrm{crit}}$, we have $\mathrm{at}(B) \subseteq V(\mathcal{Q}_1) \cup V(\mathcal{Q}_\ell)$. Let

$$\begin{aligned} Z &\coloneqq \mathrm{art}(\mathcal{Q}_1) \cup \mathrm{art}(\mathcal{Q}_\ell), \\ \widehat{Z} &\coloneqq \mathrm{art}(\widehat{\mathcal{Q}}_1) \cup \mathrm{art}(\widehat{\mathcal{Q}}_\ell). \end{aligned}$$

Claim 2. Let $B, B' \in \mathcal{B}_{\mathrm{crit}}$ such that $\theta(B) = \theta(B')$.

1. If B and B' are not aligned, then
$$\mathrm{at}(B) \cap \mathrm{at}(B') = \mathrm{at}(B) \cap Z = \mathrm{at}(B') \cap Z,$$
and B or B' is embedded in $\boldsymbol{D}$.
2. At most one of B, B' is embedded in $\boldsymbol{D}$.

Proof of the claim: Let $\theta := \theta(B) = \theta(B')$. Suppose that $\mathrm{at}(B) = \{v_1, \ldots, v_r\}$ and $\mathrm{at}(B') = \{v'_1, \ldots, v'_{r'}\}$. Since the type of a bridge contains information about the number of vertices of attachment, their fibres, and their height, we have $s = s'$, and without loss of generality we may assume that, for every i, the vertices v_i and v'_i belong to the same fibre and have the same height in this fibre. Thus, v_i is an articulation vertex of its fibre if and only if v'_i is one (and in this case, they are equal). Hence $\mathrm{at}(B) \cap Z = \mathrm{at}(B') \cap Z$ and therefore, we have
$$\mathrm{at}(B) \cap Z = \mathrm{at}(B) \cap \mathrm{at}(B') \cap Z \subseteq \mathrm{at}(B) \cap \mathrm{at}(B').$$

Furthermore, for every fibre i and every height j, there are at most two vertices $v, v' \in V(H_i)$ of height j that may be vertices of attachment of a bridge, one on the path Q_i and one on the path Q'_i. It follows that every vertex in $\mathrm{at}(B) \cap \mathrm{at}(B')$ lies in $V(Q_i) \cap V(Q'_i)$. Hence, every path from u to u' in $\mathcal{Q}_i$ passes through v and therefore, v is an articulation vertex of $\mathcal{Q}_i$. This proves $\mathrm{at}(B) \cap \mathrm{at}(B') \subseteq \mathrm{at}(B) \cap Z$ and, hence, equality.

This means that if B and B' are not aligned, one of them has some vertices of attachment in $V(Q_i) \setminus V(Q'_i)$ and the other has some vertices of attachment in $V(Q'_i) \setminus V(Q_i)$. Thus, one must be embedded in $\boldsymbol{f}_{i-1}$ and one in $\boldsymbol{f}_i$. At least one of these sets is a subset of $\boldsymbol{D}$.

Since G is 3-connected, we have $r \geq 3$, and this means we cannot embed both B and B' into $\boldsymbol{D}$, because this would violate planarity (reasoning via a $K_{3,3}$-minor). ⌟

Note that for a connected component of $G - J$ or $\widehat{G} - \widehat{J}$, there is only a bounded number of possible isomorphism types θ_A. Thus, we can check whether two bridges have the same isomorphism type using the formulae $\mathsf{bridge\text{-}iso}_A$. Hence, employing the formulae $\mathsf{bridge\text{-}iso}_A$, $\mathsf{aligned}$, $\mathsf{att\text{-}vert}$ and requiring the variable z in the definition of $\mathsf{csps\text{-}art}$ (cf. Lemma 9.31) additionally to be in $V(H_1)$ and $V(H_\ell)$, respectively, we can show that all restrictions the claim imposes on G are definable in $\mathsf{C}_{\mathsf{W}}^{s+3}$. Thus, for all $\widehat{B}, \widehat{B}' \in \widehat{\mathcal{B}}_{\mathrm{crit}}$, if $\widehat{B}$ and $\widehat{B}'$ are not aligned, then $\mathrm{at}(\widehat{B}) \cap \mathrm{at}(\widehat{B}') = \mathrm{at}(\widehat{B}) \cap \widehat{Z} = \mathrm{at}(\widehat{B}') \cap \widehat{Z}$.

There is an interesting special case of pairs of critical bridges that we need to deal with separately. Consider a type θ such that there are exactly two bridges $B, B' \in \mathcal{B}_{\mathrm{crit}}$ of type θ, and these two bridges are not aligned. Then, by Claim 2, either both B and B' are ℓ-bridges or both are 1-bridges. Moreover, the claim implies that exactly one of them is embedded in $\boldsymbol{f}_1 \cup \boldsymbol{f}_{\ell-1} \subseteq \boldsymbol{D}$. Say, B is embedded in $\boldsymbol{f}_1 \cup \boldsymbol{f}_{\ell-1}$. We call $\{B, B'\}$ an *opposite pair*. That is, an opposite pair in G is an unordered pair $\{B, B'\}$ of bridges $B, B' \in \mathcal{B}_{\text{crit}}$ such that $\theta(B) = \theta(B')$, there is no $B'' \in \mathcal{B}_{\text{crit}} \setminus \{B, B'\}$ such that $\theta(B'') = \theta(B)$, and B, B' are not aligned.

Claim 3. Let $\{B_1, B_1'\}, \ldots, \{B_p, B_p'\}$ be a list of all opposite pairs of G. Then the graph

$$K^+ := K \cup \bigcup_{i=1}^{p} (B_i \cup B_i')$$

is planar.

Proof of the claim: Recall that, for every i, either B_i or B_i' is embedded in $\boldsymbol{D}$. Without loss of generality, we assume that, for all i, the bridge B_i is embedded in $\boldsymbol{D}$. Then $K \cup \bigcup_{i=1}^{p} B_i$ is planar, because it is embedded in $\boldsymbol{D}$. Moreover, for every i, the bridges B_i and B_i' are isomorphic and have vertices of attachment in the same fibres and of the same height. We can just copy the embedding of B_i and embed B_i' in the same way outside of $\boldsymbol{D}$. ⌟

An *opposite pair* in $\widehat{G}$ is an unordered pair $\{\widehat{B}, \widehat{B}'\}$ of bridges $\widehat{B}, \widehat{B}' \in \widehat{\mathcal{B}}_{\text{crit}}$ such that $\theta(\widehat{B}) = \theta(\widehat{B}')$, there is no $\widehat{B}'' \in \mathcal{B}_{\text{crit}} \setminus \{\widehat{B}, \widehat{B}'\}$ such that $\theta(\widehat{B}'') = \theta(\widehat{B})$, and $\widehat{B}$, $\widehat{B}'$ are not aligned. Let $\{\widehat{B}_1, \widehat{B}_1'\}, \ldots, \{\widehat{B}_{p'}, \widehat{B}_{p'}'\}$ be a list of all opposite pairs in $\widehat{G}$. It is easy to see that $p = p'$. We let

$$\widehat{K}^+ := \widehat{K} \cup \bigcup_{i=1}^{p} (\widehat{B}_i \cup \widehat{B}_i').$$

It is easy to construct $\mathsf{C}_{\mathsf{w}}^{s+3}$-formulae

$$\mathsf{K\text{-}plus\text{-}vert}(x, x', x'', y), \qquad \mathsf{K\text{-}plus\text{-}edge}(x, x', x'', y_1, y_2),$$

such that

$$\begin{aligned} V(K^+) &= \mathsf{K\text{-}plus\text{-}vert}[G, u, u', u'', y], & E(K^+) &= \mathsf{K\text{-}plus\text{-}edge}[G, u, u', u'', y_1, y_2], \\ V(\widehat{K}^+) &= \mathsf{K\text{-}plus\text{-}vert}[\widehat{G}, \widehat{u}, \widehat{u}', \widehat{u}'', y], & E(\widehat{K}^+) &= \mathsf{K\text{-}plus\text{-}edge}[\widehat{G}, \widehat{u}, \widehat{u}', \widehat{u}'', y_1, y_2]. \end{aligned}$$

Let us call a J-bridge B *super-critical* if it is critical, but not contained in an opposite pair. Let $\mathcal{B}_{\text{sc}}$ be the set of all super-critical J-bridges (see Figure 9.5). Similarly, we call a $\widehat{J}$-bridge $\widehat{B}$ *super-critical* if it is critical, but not contained in an opposite pair, and we let $\widehat{\mathcal{B}}_{\text{sc}}$ be the set of all super-critical $\widehat{J}$-bridges.

Observe that the bijection β between $\mathcal{B}_{\text{crit}}$ and $\widehat{\mathcal{B}}_{\text{crit}}$ defined above induces a bijection between $\mathcal{B}_{\text{sc}}$ and $\widehat{\mathcal{B}}_{\text{sc}}$. Moreover, we have $G = K^+ \cup \bigcup_{B \in \mathcal{B}_{\text{sc}}} B$, and this implies $\widehat{G} = \widehat{K}^+ \cup \bigcup_{\widehat{B} \in \widehat{\mathcal{B}}_{\text{sc}}} \widehat{B}$.

Next, we expand K^+ and $\widehat{K}^+$ by new colours that encode the information about which bridges are attached to which vertices. For every $v \in V(K^+)$, we let

$$\Theta(v) := \{\!\!\{ \theta(B) \mid B \in \mathcal{B}_{\text{crit}}, v \in \text{at}(B) \}\!\!\}$$

Moreover, we let

$$\Phi(v) := \begin{cases} x & \text{if } v = u, \\ x' & \text{if } v = u', \\ x'' & \text{if } v = u'', \\ i & \text{if } v \in V(\mathring{H}_i) \setminus \{u''\} \text{ for some } i \in \ell, \\ \bot & \text{if } v \in V(K) \setminus V(J). \end{cases}$$

We view $\Theta(v)$ and $\Phi(v)$ as additional colours of the vertices of K^+ and consider K^+ in the following as a coloured graph where these new colours are incorporated. We note that, for every colour $c \in \text{rg}(\Phi)$, we have a $\mathsf{C}_{\mathsf{w}}^{s+3}$-formula $\mathsf{Phi}_c(x, x', x'', y)$ such that $G \models \mathsf{Phi}_c(u, u', u'', v) \iff \Phi(v) = c$. Similarly, using the formulae $\mathsf{bridge\text{-}iso}_A$ defined above, for every colour $c \in \text{rg}(\Theta)$, we can construct a $\mathsf{C}_{\mathsf{w}}^{s+3}$-formula $\mathsf{Theta}_c(x, x', x'', y)$ such that $G \models \mathsf{Theta}_c(u, u', u'', v) \iff \Theta(v) = c$.

We can use these formulae to transfer the colouring to the graph $\widehat{K}^+$: for $\widehat{v} \in V(\widehat{K}^+)$, we let $\widehat{\Phi}(\widehat{v})$ be the unique $c \in \text{rg}(\Phi)$ such that $\widehat{G} \models \mathsf{Phi}_c(\widehat{u}, \widehat{u}', \widehat{u}'', \widehat{v})$. If there is more than one or no such c, then the graphs can be distinguished by a $\mathsf{C}_{\mathsf{w}}^{s+3}$-formula. Similarly, for $\widehat{v} \in V(\widehat{K}^+)$, we let $\widehat{\Theta}(\widehat{v})$ be the unique $c \in \text{rg}(\Theta)$ such that $\widehat{G} \models \mathsf{Theta}_c(\widehat{u}, \widehat{u}', \widehat{u}'', \widehat{v})$. In the following, we regard K^+ and $\widehat{K}^+$ as coloured graphs with these colours, in addition to the colours inherited from G and $\widehat{G}$.

Claim 4.

$$K^+ \cong \widehat{K}^+.$$

Proof of the claim: The graphs K^+ and $\widehat{K}^+$ with all colours are definable in G by $\mathsf{C}_{\mathsf{w}}^{s+3}$-formulae using the three parameters u, u', u''. Moreover, K^+ is a planar graph, and thus, there is a $\mathsf{C}_{\mathsf{w}}^{4}$-formula that identifies it. From this formula and the formulae defining membership in the subgraphs K^+ and $\widehat{K}^+$, we can construct a $\mathsf{C}_{\mathsf{w}}^{s+3}$-formula that would distinguish G and $\widehat{G}$ if K^+ and $\widehat{K}^+$ were non-isomorphic. ⌟

In the following, we let π be an isomorphism from K^+ to $\widehat{K}^+$. It is our goal to extend π to an isomorphism from G to $\widehat{G}$. For this, we need to extend π to all super-critical bridges. We process the bridges by types. So let θ be a type. Let $\mathcal{B}_\theta$ be the set of all $B \in \mathcal{B}_{\text{sc}}$ with $\theta(B) = \theta$, and similarly, let $\widehat{\mathcal{B}}_\theta$ be the set of all $\widehat{B} \in \widehat{\mathcal{B}}_{\text{sc}}$ with $\widehat{\theta}(B) = \theta$. Then the bijection β between $\mathcal{B}_{\text{crit}}$ and $\widehat{\mathcal{B}}_{\text{crit}}$ defined above induces a bijection between $\mathcal{B}_\theta$ and $\widehat{\mathcal{B}}_\theta$.

We shall construct an extension π_θ of π that is an isomorphism from $K^+ \cup \bigcup_{B \in \mathcal{B}_\theta} B$ to $\widehat{K}^+ \cup \bigcup_{\widehat{B} \in \widehat{\mathcal{B}}_\theta} \widehat{B}$. We can easily combine all the π_θ to one isomorphism from G to $\widehat{G}$, because they all coincide on K^+ and the intersection between any two bridges is in K^+ and $\widehat{K}^+$, respectively.

Suppose first that all $B, B' \in \mathcal{B}_\theta$ are aligned. Then, for all $\widehat{B}, \widehat{B}' \in \widehat{\mathcal{B}}_\theta$, we have $\text{at}(\widehat{B}) = \text{at}(\widehat{B}')$, since $\mathsf{aligned} \in \mathsf{C}_{\mathsf{w}}^{\max\{7,s+2\}}$. Note that β induces an isomorphism from $\bigcup_{B\in\mathcal{B}_\theta} B$ to $\bigcup_{\widehat{B}\in\widehat{\mathcal{B}}_\theta} \widehat{B}$. We can easily extend this isomorphism to an isomorphism from $K^+ \cup \bigcup_{B\in\mathcal{B}_\theta} B$ to $\widehat{K}^+ \cup \bigcup_{\widehat{B}\in\widehat{\mathcal{B}}_\theta} \widehat{B}$, because the attachment pattern is encoded in the colouring of the bridges.

Suppose next that there are $B_1, B_2 \in \mathcal{B}_\theta$ that are not aligned. Then $\mathcal{B}_\theta \geq 3$, because otherwise $\{B_1, B_2\}$ would be an opposite pair. Say, $\mathcal{B}_\theta = \{B_1, B_2, \ldots, B_m\}$. Without loss of generality, we assume that every B_i is a 1-bridge. The case that every B_i is an ℓ-bridge or every B_i is an $(\ell, 1)$-bridge can be dealt with in the same way. By Part 1 of Claim 2, one of B_1 and B_2, say B_1, is embedded in $\boldsymbol{f}_1$. But then, by Part 2 of Claim 2, the bridges $B_2, \ldots, B_p$ are not embedded in $\boldsymbol{D}$. By Part 1 again, $B_2, \ldots, B_p$ are aligned. For every $i \in [p]$, let $X_i \coloneqq \mathrm{at}(B_i) \cap \mathrm{art}(\mathcal{Q}_1)$ and $Y_i \coloneqq \mathrm{at}(B_i) \setminus X_i$. Then, by Part 1, we know that $X_1 = X_2 = \cdots = X_p$ and $Y_2 = \cdots = Y_p$ and $Y_1 \cap Y_i = \emptyset$ for $i \geq 2$. Now the key observation is that the vertices in Y_1 have a different colour than the vertices in Y_2, because they are attached to a different number of bridges of type θ. The isomorphism π maps Y_1 to a set $\widehat{Y_1}$ of vertices that are attached to exactly one bridge of type θ, and it maps Y_2 to a set $\widehat{Y_2}$ of vertices that are attached to $p-1$ bridges of type θ. Moreover, it maps $X \coloneqq X_1$ to a set $\widehat{X}$ of vertices that are attached to p bridges of type θ. We can now extend the isomorphism π by mapping B_1 to the unique bridge of type θ that is attached to the vertices in $\widehat{Y_1}$ and by mapping $B_2, \ldots, B_p$ to the $p-1$ bridges of type θ that are attached to $\widehat{Y_2}$. □

This completes the proof of Lemma 9.29 and thus also the proof of Theorem 9.24. We finally prove the bound $2g+3$ if the surface $\boldsymbol{S}$ that G is embedded into is orientable.

Corollary 9.57. *The Weisfeiler-Leman dimension of a graph embeddable in an orientable surface of Euler genus g is at most $2g+3$.*

Proof: The Euler genus of an orientable surface is always even. Suppose G is a graph embeddable in an orientable surface of Euler genus g. Since the subgraphs obtained by cutting through the beads are also embeddable in orientable surfaces of smaller Euler genus, this procedure reduces the Euler genus by at least 2. Therefore, inductively proceeding as described in the previous section, redefining s to be the number of variables needed for graphs embeddable in orientable surfaces of Euler genus at most $g-2$, we can improve our bound from Theorem 9.24 to $2g+3$. □

9.5 Discussion

We have proved an upper bound of $4g+3$ for the Weisfeiler-Leman dimension of graphs of Euler genus g and showed that if G is known to be embeddable on an orientable surface of Euler genus g, the bound improves to $2g+3$. The immediate question that remains is how tight our bound is.

For every $g \in \mathbb{N}$, the class of graphs of Euler genus at most g contains the class of planar graphs since those are precisely the graphs of Euler genus 0. Therefore, the results presented in this chapter are a generalisation of the ones from Chapter 8. We believe that, by refining our arguments in some places, it might be possible to reduce the bound from Theorem 9.24 to $3g + 3$ or even $2g + 3$; any further improvement seems to require substantial additional ideas. As indicated in Chapter 8, it is conceivable that the Weisfeiler-Leman dimension of planar graphs is 2. If this is the case, the additive term in our bound would automatically drop to 2.

Towards an approach from below, using the Cai, Fürer, Immerman construction (see Section 6.4 and [30]), it is easy to prove a linear lower bound of $\epsilon \cdot g$ for the Weisfeiler-Leman dimension of graphs of Euler genus g, albeit with a rather small constant $\epsilon > 0$. To close the gap between upper and lower bound, it may be worthwhile to spend some effort on improving the lower bound.

By the graph minor theory due to Robertson and Seymour, for every surface, the graphs embeddable on the surface can be completely characterised via a certain list of excluded minors [145]. For example, recall Theorem 2.2, which says that a graph is planar if and only if it excludes both the complete graph K_5 and the complete bipartite graph $K_{3,3}$ as minors. Archdeacon determined the 35 excluded minors that characterise the graphs embeddable in the projective plane [2]. Unfortunately, apart from the sphere and the projective plane, further explicit excluded-minor characterisations are not known. Still, a natural and interesting target for further study of the Weisfeiler-Leman dimension of a graph class would be the one characterised by the excluded minor K_ℓ. We know that the Weisfeiler-Leman dimension of this class is bounded [64]. But even an exponential bound of the Weisfeiler-Leman dimension in terms of ℓ would be major progress.

10 Conclusion

This thesis introduces the reader to the realm of the Weisfeiler-Leman algorithm, a natural combinatorial approach to grasping the structure of a graph. The presented results lay the foundation for further study and a deeper comprehension of the algorithm. In particular, in the context of the graph isomorphism problem, the power and the limits of the algorithm are of crucial importance for a precise understanding of how and where it can be used. Additionally, as previous research has shown, by understanding aspects of the algorithm in detail, more and more connections to seemingly unrelated areas can be uncovered. Thus, research on the Weisfeiler-Leman algorithm can have interdisciplinary impact. Vice versa, the known links to other fields may enable entirely new approaches to further study of the algorithm.

We discussed the iteration number and the dimension as two central parameters of the algorithm. The iteration number yields an upper bound on the quantifier depth of a formula that distinguishes the given graph from every other graph, whereas the dimension provides information about the number of variables needed to define the graph in the logic C.

In the following, we summarise the contribution of this thesis and possible future projects by recapitulating what we know and what we do not know yet about certain dimensions of the Weisfeiler-Leman algorithm.

Chapters 4 and 6 provide a very precise understanding of the mechanics of the 1-dimensional Weisfeiler-Leman algorithm. By the results in Chapter 6, we can completely characterise the graphs and even the finite relational structures that are identified by the 1-dimensional Weisfeiler-Leman algorithm. We also discovered properties that are particular about dimension $k = 1$ of the algorithm. For example, if a graph is identified by the 1-dimensional Weisfeiler-Leman algorithm, then every vertex-coloured version of it is also identified by it. A second result is that, once we are given a graph, there is an efficient procedure that decides whether the graph is identified by the 1-dimensional Weisfeiler-Leman algorithm. Moreover, investigating the complexity of the algorithm, by the results in Chapter 4, there are infinitely many graphs on which it actually takes the maximum theoretically possible number of iterations to stabilise. So, in a sense, the spreading of information about the structure of these graphs is as slow as it can possibly be. This bound is, up to an additive constant of 1, even tight for every graph order $n \in \mathbb{N}_{\geq 10}$.

Due to their low vertex degrees, the infinite families of long-refinement graphs that we have found have a particularly nice and comprehensible structure. However, it is not clear how to extract from them two graphs of equal size which Colour Refinement does not distinguish before having computed the stable partition on each of them. The best lower bound on the number of iterations of the 1-dimensional algorithm needed to distinguish two input graphs of order n that are not equivalent with respect to the algorithm is still $n - O(\sqrt{n})$ [113]. It remains an open problem to improve on this bound.

Concerning higher dimensions of the algorithm, there are many more big challenges for follow-up research projects. We were the first to improve on the trivial upper bound on the number of iterations of the 2-dimensional Weisfeiler-Leman algorithm. For the non-counting version of the algorithm, our bound from Chapter 5 remains the best known. The current best upper bound on the number of iterations of the classical 2-dimensional Weisfeiler-Leman algorithm is $O(n \log n)$ [120], but it is still open whether this bound is tight. The best known lower bound on the iteration number on n-vertex graphs is $\Omega(n)$ [48]. New explicit lower bounds, e.g. by constructions similar to the ones we presented in Chapter 4, could help close gaps. Unfortunately, as the techniques in Chapter 5 as well as in [120] show, it is highly non-trivial to track how the 2-dimensional Weisfeiler-Leman algorithm processes a graph. It will become even more involved for higher dimensions.

As discussed in Chapter 5, there is not much hope to find a complete characterisation of the graphs identified by the 2-dimensional Weisfeiler-Leman algorithm. All graphs of colour class size at most 3 are identified by the algorithm, but by the Cai, Fürer, Immerman construction, there are graphs of maximum colour class size 4 which are not identified by the 2-dimensional Weisfeiler-Leman algorithm [30] (see also Section 6.4). Towards this direction of research, [46] describes a procedure to decide whether a graph of colour class size at most 4 has Weisfeiler-Leman dimension at most 2. Still, a major obstacle for a full understanding of all graphs of Weisfeiler-Leman dimension 2 is the lack of a characterisation of the strongly regular graphs that are identified by the algorithm or, equivalently, determined up to isomorphism by their parameters. Also a characterisation of the graphs identified by the k-dimensional Weisfeiler-Leman algorithm for any $k \geq 2$ is not in sight.

Instead of analysing what a fixed dimension of the Weisfeiler-Leman algorithm is capable to detect, we slightly changed perspectives and investigated which dimension is necessary or sufficient to detect certain graphs or graph properties in Chapters 7–9. This usually amounts to narrowing the gap between lower and upper bounds instead of closing it. For most natural graph classes, the Weisfeiler-Leman dimension remains open, even asymptotically. In Chapters 8 and 9, we presented upper bounds on the Weisfeiler-Leman dimension of the class of planar graphs and more generally, the class of graphs of Euler genus $g \in \mathbb{N}_0$. We determined the precise Weisfeiler-Leman dimension of the class of planar graphs up to an additive constant of 1. Knowing that it is 2 or 3, we developed an explicit linear parametrisation by the Euler genus of the input graph as an upper bound on its Weisfeiler-Leman dimension.

The first natural follow-up task is to determine the precise Weisfeiler-Leman dimension of the class of planar graphs. As demonstrated in Chapter 7, for the reduction to 3-connected planar graphs, it suffices to employ the 2-dimensional algorithm. Still, our approach relies on the determination of vertex orbits in those graphs. Thus, one way to bring the dimension for the identification of planar graphs down to 2 would be showing that the 2-dimensional Weisfeiler-Leman algorithm determines vertex orbits on the class of all 3-connected planar graphs and identifies every such graph.

Also, the precise influence of the Euler genus on the dimension remains an open problem. A first step in this direction would be good lower bounds. Beyond graphs of bounded Euler genus, one can also try to tie the Weisfeiler-Leman dimension to other graph invariants. As roughly sketched in Section 7.5, we have obtained improved upper and lower bounds on the parametrisation of the Weisfeiler-Leman dimension of a graph by its treewidth. In [69], the authors provide new bounds on the Weisfeiler-Leman dimension of graphs of clique width k and of graphs of rank width k', but also there, it is open how tight those bounds are. A challenging problem related to the quest for a parametrisation of the Weisfeiler-Leman dimension by the Euler genus is the study of the Weisfeiler-Leman dimension of a graph class that excludes certain minors, e.g. the complete graph K_ℓ.

Bibliography

[1] Babak Ahmadi, Kristian Kersting, Martin Mladenov, and Sriraam Natarajan. Exploiting symmetries for scaling loopy belief propagation and relational training. *Machine Learning Journal*, 92(1):91–132, 2013.

[2] Dan Archdeacon. A Kuratowski theorem for the projective plane. *Journal of Graph Theory*, 5(3):243–246, 1981.

[3] Vikraman Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. On Weisfeiler-Leman invariance: subgraph counts and related graph properties. In *Proceedings of the Twenty-Second International Symposium on Fundamentals of Computation Theory*, pages 111–125, Copenhagen, Denmark, August 2019.

[4] Vikraman Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. On the power of color refinement. In *Proceedings of the Twentieth International Symposium on Fundamentals of Computation Theory*, pages 339–350, August 2015.

[5] Vikraman Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. On Tinhofer's linear programming approach to isomorphism testing. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sannella, editors, *Proceedings of the Fortieth International Symposium on Mathematical Foundations of Computer Science*, volume 9235 of *Lecture Notes in Computer Science*, pages 26–37. Springer Verlag, 2015.

[6] Vikraman Arvind, Johannes Köbler, Gaurav Rattan, and Oleg Verbitsky. Graph isomorphism, color refinement, and compactness. *Computational Complexity*, 26(3):627–685, 2017.

[7] Albert Atserias and Elitza N. Maneva. Sherali-Adams relaxations and indistinguishability in counting logics. *SIAM Journal on Computing*, 42(1):112–137, 2013.

[8] Albert Atserias and Joanna Ochremiak. Definable ellipsoid method, sums-of-squares proofs, and the isomorphism problem. In *Proceedings of the Thirty-Third Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 66–75, Oxford, United Kingdom, July 2018.

[9] László Babai. Lectures on graph isomorphism, 1979. Mimeographed lecture notes.

[10] László Babai. Monte Carlo algorithms in graph isomorphism testing. Technical report 79–10, Université de Montréal, 1979.

[11] László Babai. *Handbook of combinatorics*. In volume 2. MIT Press, Cambridge, MA, USA, 1995. Chapter Automorphism groups, isomorphism, reconstruction, pages 1447–1540.

[12] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the Fourty-Eighth Annual ACM Symposium on Theory of Computing*, pages 684–697, June 2016.

[13] László Babai, Anuj Dawar, Pascal Schweitzer, and Jacobo Torán. The graph isomorphism problem (Dagstuhl seminar 15511). *Dagstuhl Reports*, 5(12):1–17, 2015.

[14] László Babai, Paul Erdős, and Stanley M. Selkow. Random graph isomorphism. *SIAM Journal on Computing*, 9(3):628–635, 1980.

[15] László Babai, D. Yu. Grigoryev, and David M. Mount. Isomorphism of graphs with bounded eigenvalue multiplicity. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 310–324, San Francisco, CA, USA, May 1982.

[16] László Babai and Ludek Kucera. Canonical labelling of graphs in linear average time. In *Proceedings of the Twentieth Annual Symposium on Foundations of Computer Science*, pages 39–46. IEEE Computer Society, October 1979.

[17] Luitpold Babel, Irina V. Chuvaeva, Mikhail Klin, and Dmitrii Pasechnik. Algebraic combinatorics in mathematical chemistry. Methods and algorithms, II. Program implementation of the Weisfeiler-Leman algorithm, 2010.

[18] Christoph Berkholz. The propagation depth of local consistency. In *Proceedings of the Twentieth International Conference on Principles and Practice of Constraint Programming*, pages 158–173, Lyon, France, September 2014.

[19] Christoph Berkholz, Paul S. Bonsma, and Martin Grohe. Tight lower and upper bounds for the complexity of canonical colour refinement. In *Proceedings of the Twenty-First Annual European Symposium*, pages 145–156, Sophia Antipolis, France, September 2013.

[20] Christoph Berkholz and Martin Grohe. Limitations of algebraic approaches to graph isomorphism testing. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Proceedings of the Fourty-Second International Colloquium on Automata, Languages and Programming*, volume 9134 of *Lecture Notes in Computer Science*, pages 155–166. Springer Verlag, 2015.

[21] Christoph Berkholz and Jakob Nordström. Near-optimal lower bounds on quantifier depth and Weisfeiler-Leman refinement steps. In *Proceedings of the Thirty-First Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 267–276, New York, NY, USA, July 2016.

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

[23] Hans L. Bodlaender. Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees. *Journal of Algorithms*, 11(4):631–643, 1990.

[24] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1–2):1–45, 1998.

[25] Béla Bollobás. Distinguishing vertices of random graphs. In *Graph theory.* Volume 62, North-Holland Math. Stud. Pages 33–49. North-Holland, Amsterdam–New York, 1982.

[26] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25(2):127–132, 1987.

[27] Andries E. Brouwer. Spectrum and connectivity of graphs. *CWI Quarterly*, 9(1-2):37–40, 1996.

[28] Andries E. Brouwer and Jack H. Koolen. The vertex-connectivity of a distance-regular graph. *European Journal of Combinatorics*, 30(3):668–673, 2009.

[29] Andries E. Brouwer and Dale M. Mesner. The connectivity of strongly regular graphs. *European Journal of Combinatorics*, 6(3):215–216, 1985.

[30] Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992.

[31] Peter J. Cameron. Coherent configurations, association schemes and permutation groups. In *Groups, combinatorics & geometry*, pages 55–71. World Scientific, 2003.

[32] Alain Cardon and Maxime Crochemore. Partitioning a graph in $O(|A|\log|A|)$. *Theoretical Computer Science*, 19:85–98, 1982.

[33] Ashok K. Chandra and David Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25(1):99–128, 1982.

[34] Li-Chien Chang. The uniqueness and nonuniqueness of the triangular association scheme. *Science Record*, 3:604–613, 1959.

[35] Gang Chen and Ilia N. Ponomarenko. Lectures on coherent configurations, 2019. `https://pdmi.ras.ru/~inp/ccNOTES.pdf`. Accessed: 2020-01-10.

[36] Paolo Codenotti, Hadi Katebi, Karem A. Sakallah, and Igor L. Markov. Conflict analysis and branching heuristics in the search for graph automorphisms. In *25th IEEE International Conference on Tools with Artificial Intelligence*, pages 907–914, Herndon, VA, USA, November 2013.

[37] Paul T. Darga, Mark H. Liffiton, Karem A. Sakallah, and Igor L. Markov. Exploiting structure in symmetry detection for CNF. In *Proceedings of the Fourty-First Design Automation Conference*, pages 530–534, San Diego, CA, USA. ACM, June 2004.

[38] Samir Datta, Nutan Limaye, Prajakta Nimbhorkar, Thomas Thierauf, and Fabian Wagner. Planar graph isomorphism is in log-space. In *Proceedings of the Twenty-Fourth IEEE Conference on Computational Complexity*, pages 203–214. IEEE Computer Society, 2009.

[39] Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász meets Weisfeiler and Leman. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *Proceedings of the Fourty-Fifth International Colloquium on Automata, Languages and Programming*, volume 107 of *LIPIcs*, 40:1–40:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

[40] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[41] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[42] Sergei Evdokimov and Ilia N. Ponomarenko. Separability number and schurity number of coherent configurations. *The Electronic Journal of Combinatorics*, 7, 2000.

[43] Sergei Evdokimov and Ilia N. Ponomarenko. On the vertex connectivity of a relation in an association scheme. *Journal of Mathematical Sciences*, 134(5):2354–2357, 2006.

[44] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In *Proceedings of a SIAM-AMS Symposium on Complexity of computation*, pages 43–73, 1974.

[45] Ion S. Filotti and Jack N. Mayer. A polynomial-time algorithm for determining the isomorphism of graphs of fixed genus. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, pages 236–243, Los Angeles, CA, USA, April 1980.

[46] Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. Identifiability of graphs with small color classes by the Weisfeiler-Leman algorithm. *Computing Research Repository*, abs/1907.02892, 2019. arXiv: `1907.02892`.

[47] Martin Fürer. Graph isomorphism testing without numerics for graphs of bounded eigenvalue multiplicity. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 624–631, San Francisco, CA, USA. ACM, New York, 1995.

[48] Martin Fürer. Weisfeiler-Lehman refinement requires at least a linear number of iterations. In *Proceedings of the Twenty-Eighth International Colloquium, Automata, Languages and Programming*, pages 322–333, Crete, Greece, July 2001.

[49] Martin Fürer. On the combinatorial power of the Weisfeiler-Lehman algorithm. In Dimitris Fotakis, Aris Pagourtzis, and Vangelis Th. Paschos, editors, *Proceedings of the Tenth International Conference on Algorithms and Complexity*, volume 10236 of *Lecture Notes in Computer Science*, pages 260–271, Athens, Greece. Springer Verlag, May 2017.

[50] Merrick L. Furst, John E. Hopcroft, and Eugene M. Luks. Polynomial-time algorithms for permutation groups. In *Proceedings of the Twenty-First Annual Symposium on Foundations of Computer Science*, pages 36–41, Syracuse, NY, USA, October 1980.

[51] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[52] Maximilian Gödicke. *The Iteration Number of the Weisfeiler-Lehman-Algorithm.* Master's thesis, RWTH Aachen University, 2015.

[53] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 285–306. 2019.

[54] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research*, 5:73–90, 1989.

[55] Erich Grädel, Martin Grohe, Benedikt Pago, and Wied Pakusa. A finite-model-theoretic view on propositional proof complexity. *Logical Methods in Computer Science*, 15(1), 2019.

[56] Erich Grädel and Martin Otto. On logics with two variables. *Theoretical Computer Science*, 224(1-2):73–113, 1999.

[57] Erich Grädel, Martin Otto, and Eric Rosen. Two-variable logic with counting is decidable. In *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 306–317, Warsaw, Poland. IEEE Computer Society, June 1997.

[58] Martin Grohe. Finite variable logics in descriptive complexity theory. *The Bulletin of Symbolic Logic*, 4(4):345–398, 1998.

[59] Martin Grohe. Fixed-point logics on planar graphs. In *Proceedings of the Thirteenth Annual IEEE Symposium on Logic in Computer Science*, pages 6–15, Indianapolis, IN, USA, June 1998.

[60] Martin Grohe. Equivalence in finite-variable logics is complete for polynomial time. *Combinatorica*, 19(4):507–532, 1999.

[61] Martin Grohe. Isomorphism testing for embeddable graphs through definability. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 63–72, Portlang, OR, USA, May 2000.

[62] Martin Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *Journal of the ACM*, 59(5):27:1–27:64, 2012.

[63] Martin Grohe. Tangled up in blue (a survey on connectivity, decompositions, and tangles). *Computing Research Repository*, abs/1605.06704, 2016. arXiv: `1605.06704`.

[64] Martin Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017.

[65] Martin Grohe, Kristian Kersting, Martin Mladenov, and Erkal Selman. Dimension reduction via colour refinement. In *Proceedings of the Twenty-Second Annual European Symposium on Algorithms*, volume 8737 of *Lecture Notes in Computer Science*, pages 505–516, Wroclaw, Poland. Springer, September 2014.

[66] Martin Grohe and Sandra Kiefer. A linear upper bound on the Weisfeiler-Leman dimension of graphs of bounded genus. In *Proceedings of the Fourty-Sixth International Colloquium on Automata, Languages, and Programming*, 117:1–117:15, Patras, Greece, July 2019.

[67] Martin Grohe and Julian Mariño. Definability and descriptive complexity on databases of bounded tree-width. In Catriel Beeri and Peter Buneman, editors, *Proceedings of the Seventh International Conference on Database Theory*, volume 1540 of *Lecture Notes in Computer Science*, pages 70–82, Jerusalem, Israel. Springer, January 1999.

[68] Martin Grohe and Dániel Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. *SIAM Journal on Computing*, 44(1):114–159, 2015.

[69] Martin Grohe and Daniel Neuen. Canonisation and definability for graphs of bounded rank width. In *Proceedings of the Thirty-Fourth Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–13, Vancouver, BC, Canada, June 2019.

[70] Martin Grohe, Daniel Neuen, and Pascal Schweitzer. A faster isomorphism test for graphs of small degree. In *Proceedings of the Fifty-Ninth Annual IEEE Symposium on Foundations of Computer Science*, pages 89–100, Paris, France, October 2018.

[71] Martin Grohe and Martin Otto. Pebble games and linear equations. *Journal of Symbolic Logic*, 80(3):797–844, 2015.

[72] Martin Grohe and Pascal Schweitzer. Isomorphism testing for graphs of bounded rank width. In *Proceedings of the Fifty-Sixth Annual IEEE Symposium on Foundations of Computer Science*, pages 1010–1029, Berkeley, CA, USA, October 2015.

[73] Martin Grohe and Oleg Verbitsky. Testing graph isomorphism in parallel by playing a game. In *Proceedings of the Thirty-Third International Colloquium on Automata, Languages and Programming*, pages 3–14, Venice, Italy, July 2006.

[74] Yuri Gurevich. Toward logic tailored for computational complexity. In Egon Börger, Walter Oberschelp, Michael M. Richter, Brigitta Schinzel, and Wolfgang Thomas, editors, *Computation and Proof Theory*, pages 175–216, Berlin, Heidelberg. Springer, 1984.

[75] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Proceedings of the Thirty-First Annual Conference on Neural Information Processing Systems*, pages 1024–1034. Long Beach, CA, USA, December 2017.

[76] Mahdieh Hasheminezhad, Brendan D. McKay, and Tristan Reeves. Recursive generation of simple planar 5-regular graphs and pentangulations. *Journal of Graph Algorithms and Applications*, 15(3):417–436, 2011.

[77] Lauri Hella. Logical hierarchies in PTIME. *Information and Computation*, 129(1):1–19, 1996.

[78] Lauri Hella, Phokion G. Kolaitis, and Kerkko Luosto. Almost everywhere equivalence of logics in finite model theory. *Bulletin of Symbolic Logic*, 2(4):422–443, 1996.

[79] Alan J. Hoffman. On the uniqueness of the triangular association scheme. *Annals of Mathematical Statistics*, 31(2):492–497, 1960.

[80] Alan J. Hoffman. On the line graph of the complete bipartite graph. *The Annals of Mathematical Statistics*, 35(2):883–885, 1964.

[81] John E. Hopcroft and Robert E. Tarjan. A V^2 algorithm for determining isomorphism of planar graphs. *Information Processing Letters*, 1(1):32–34, 1971.

[82] John E. Hopcroft and Robert E. Tarjan. Isomorphism of planar graphs. In *Proceedings of a Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 131–152, Yorktown Heights, NY, USA, March 1972.

[83] John E. Hopcroft and Robert E. Tarjan. A $V \log V$ algorithm for isomorphism of triconnected planar graphs. *Journal of Computer and System Sciences*, 7(3):323–331, 1973.

[84] John E. Hopcroft and Jin K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, pages 172–184, Seattle, WA, USA, May 1974.

[85] Neil Immerman. Number of quantifiers is better than number of tape cells. *Journal of Computer and System Sciences*, 22(3):384–406, 1981.

[86] Neil Immerman. Upper and lower bounds for first order expressibility. *Journal of Computer and System Sciences*, 25(1):76–98, 1982.

[87] Neil Immerman. Relational queries computable in polynomial time. *Information and Control*, 68(1-3):86–104, 1986.

[88] Neil Immerman. Expressibility as a complexity measure: results and directions. In *Proceedings of the Second Annual Conference on Structure in Complexity Theory*, Ithaca, NY, USA. IEEE Computer Society, June 1987.

[89] Neil Immerman. *Descriptive Complexity.* Springer Verlag, 1999.

[90] Neil Immerman and Eric Lander. Describing graphs: a first-order approach to graph canonization. In Alan L. Selman, editor, *Complexity theory retrospective*, pages 59–81. Springer, 1990.

[91] Neil Immerman and Rik Sengupta. The k-dimensional Weisfeiler-Leman algorithm. *Computing Research Repository*, abs/1907.09582, 2019. arXiv: `1907.09582`.

[92] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[93] Birgit Jenner, Johannes Köbler, Pierre McKenzie, and Jacobo Torán. Completeness results for graph isomorphism. *Journal of Computer and System Sciences*, 66(3):549–566, 2003.

[94] Birgit Jenner, Johannes Köbler, Pierre McKenzie, and Jacobo Torán. Corrigendum to “Completeness results for graph isomorphism”. *Journal of Computer and System Sciences*, 72(4):783, 2006.

[95] Tommi A. Junttila and Petteri Kaski. Engineering an efficient canonical labeling tool for large and sparse graphs. In *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments*, New Orleans, LA, USA. SIAM, January 2007.

[96] Tommi A. Junttila and Petteri Kaski. Conflict propagation and component recursion for canonical labeling. In *Proceedings of the First International ICST Conference on Theory and Practice of Algorithms in (Computer) Systems*, pages 151–162, Rome, Italy, April 2011.

[97] Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a Symposium on the Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103, Yorktown Heights, NY, USA, March 1972.

[98] Dmitri Karpov and Alexey Pastor. On the structure of a k-connected graph. *Journal of Mathematical Sciences*, 113(4):584–597, 2003.

[99] Sandra Kiefer and Daniel Neuen. The power of the Weisfeiler-Leman algorithm to decompose graphs. In *Proceedings of the Fourty-Fourth International Symposium on Mathematical Foundations of Computer Science*, 45:1–45:15, Aachen, Germany, August 2019.

[100] Sandra Kiefer, Ilia N. Ponomarenko, and Pascal Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. In *Proceedings of the Thirty-Second Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–12, Reykjavík, Iceland, June 2017.

[101] Sandra Kiefer, Ilia N. Ponomarenko, and Pascal Schweitzer. The Weisfeiler-Leman dimension of planar graphs is at most 3. *Journal of the ACM*, 66(6):44:1–44:31, 2019.

[102] Sandra Kiefer and Pascal Schweitzer. Upper bounds on the quantifier depth for graph differentiation in first order logic. In *Proceedings of the Thirty-First Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 287–296, New York, NY, USA, July 2016.

[103] Sandra Kiefer and Pascal Schweitzer. Upper bounds on the quantifier depth for graph differentiation in first-order logic. *Logical Methods in Computer Science*, 15(2), 2019.

[104] Sandra Kiefer, Pascal Schweitzer, and Erkal Selman. Graphs identified by logics with counting. *Computing Research Repository*, abs/1503.08792, 2015. arXiv: `1503.08792`.

[105] Sandra Kiefer, Pascal Schweitzer, and Erkal Selman. Graphs identified by logics with counting. In *Proceedings of the Fortieth International Symposium on Mathematical Foundations of Computer Science*, volume 9234 of *Lecture Notes in Computer Science*, pages 319–330, Milan, Italy. Springer, August 2015.

[106] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the Fifth International Conference on Learning Representations*. April 2017.

[107] Mikhail Klin, Christian Pech, and Sven Reichard. COCO2P: gap-package for the computation with coherent configurations – version 0.17, 2018. Available at `https://github.com/chpech/COCO2P`. Accessed: 2020-01-10.

[108] Johannes Köbler, Sebastian Kuhnert, Bastian Laubner, and Oleg Verbitsky. Interval graphs: canonical representations in logspace. *SIAM Journal on Computing*, 40(5):1292–1315, 2011.

[109] Johannes Köbler and Oleg Verbitsky. From invariants to canonization in parallel. In *Proceedings of the Third International Computer Science Symposium in Russia*, volume 5010 of *Lecture Notes in Computer Science*, pages 216–227, Moscow, Russia. Springer, June 2008.

[110] Stephen G. Kobourov. Force-directed drawing algorithms. In *Handbook of Graph Drawing and Visualization*, pages 383–408. Chapman and Hall/CRC, 2013.

[111] Brian G. Kodalen and William J. Martin. On the connectivity of graphs in association schemes. *The Electronic Journal of Combinatorics*, 24(4):P4.39, 2017.

[112] Dénes Kőnig. Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre. *Mathematische Annalen*, 77(4):453–465, December 1916.

[113] Andreas Krebs and Oleg Verbitsky. Universal covers, color refinement, and two-variable counting logic: lower bounds for the depth. In *Proceedings of the Thirtieth Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 689–700, Kyoto, Japan, July 2015.

[114] Ludek Kucera. Canonical labeling of regular graphs in linear average time. In *Proceedings of the Twenty-Eighth Annual Symposium on Foundations of Computer Science*, pages 271–279, Los Angeles, California, October 1987.

[115] Casimir Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae*, 15(1):271–283, 1930.

[116] Bastian Laubner. Capturing polynomial time on interval graphs. In *Proceedings of the Twenty-Fifth IEEE Symposium on Logic in Computer Science*, pages 199–208, 2010.

[117] Frank Thomsons Leighton. Circulants and the characterization of vertex-transitive graphs. *Journal of Research of the National Bureau of Standards*, 88(6):395–402, 1983.

[118] Wenchao Li, Hossein Saidi, Huascar Sanchez, Martin Schäf, and Pascal Schweitzer. Detecting similar programs via the Weisfeiler-Leman graph kernel. In *Proceedings of the Fifteenth International Conference on Software Reuse: Bridging with Social-Awareness*, pages 315–330, Limassol, Cyprus, June 2016.

[119] Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[120] Moritz Lichter, Ilia N. Ponomarenko, and Pascal Schweitzer. Walk refinement, walk logic, and the iteration number of the Weisfeiler-Leman algorithm. In *Proceedings of the Thirty-Fourth Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–13, Vancouver, BC, Canada, June 2019.

[121] Charles C. Lindner, Eric Mendelsohn, and Alexander Rosa. On the number of 1-factorizations of the complete graph. *Journal of Combinatorial Theory, Series B*, 20(3):265–282, 1976.

[122] José Luis López-Presa, Antonio Fernández Anta, and Luis Núñez Chiroque. Conauto-2.0: fast isomorphism testing and automorphism group computation. *Computing Research Repository*, abs/1108.1060, 2011. arXiv: `1108.1060`.

[123] Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982.

[124] Peter N. Malkin. Sherali-Adams relaxations of graph isomorphism polytopes. *Discrete Optimization*, 12:73–97, 2014.

[125] Rudolf Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–132, 1979.

[126] Brendan D. McKay. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.

[127] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60:94–112, 2014.

[128] Jenish C. Mehta. Dynamic complexity of planar 3-connected graph isomorphism. In *Proceedings of the Ninth International Computer Science Symposium in Russia*, volume 8476 of *Lecture Notes in Computer Science*, pages 273–286. Springer, June 2014.

[129] Gary L. Miller. Isomorphism testing for graphs of bounded genus. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, pages 225–235, Los Angeles, CA, USA, April 1980.

[130] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces.* Johns Hopkins series in the mathematical sciences. Johns Hopkins University Press, 2001.

[131] Ji Won Moon. On the line-graph of the complete bigraph. *The Annals of Mathematical Statistics*, 34(2):664–667, 1963.

[132] Harry L. Morgan. The generation of a unique machine description for chemical structures – a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965.

[133] Christopher Morris and Petra Mutzel. Towards a practical k-dimensional Weisfeiler-Leman algorithm. *Computing Research Repository*, abs/1904.01543, 2019. arXiv: `1904.01543`.

[134] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan E. Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: higher-order graph neural networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, January 2019.

[135] Ryan O'Donnell, John Wright, Chenggang Wu, and Yuan Zhou. Hardness of robust graph isomorphism, Lasserre gaps, and asymmetry of random graphs. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1659–1677, 2014.

[136] Yota Otachi and Pascal Schweitzer. Isomorphism on subgraph-closed graph classes: A complexity dichotomy and intermediate graph classes. In *Proceedings of the Twenty-Fourth International Symposium on Algorithms and Computation*, pages 111–118, Hong Kong, China, December 2013.

[137] Martin Otto. Canonization for two variables and puzzles on the square. *Annals of Pure and Applied Logic*, 85(3):243–282, 1997.

[138] Martin Otto. *Bounded Variable Logics and Counting: A Study in Finite Models*, volume 9 of *Lecture Notes in Logic.* Cambridge University Press, 2017.

[139] Robert Paige and Robert E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

[140] Ilia N. Ponomarenko. The isomorphism problem for classes of graphs that are invariant with respect to contraction. *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov.*, 174(Teor. Slozhn. Vychisl. 3):147–177, 182, 1988.

[141] Ian Pratt-Hartmann. Complexity of the two-variable fragment with counting quantifiers. *Journal of Logic, Language and Information*, 14(3):369–395, 2005.

[142] Motakuri V. Ramana, Edward R. Scheinerman, and Daniel Ullman. Fractional isomorphism of graphs. *Discrete Mathematics*, 132(1–3):247–265, 1994.

[143] Ronald C. Read and Derek G. Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1(4):339–363, 1977.

[144] Joachim Redies. *Defining PTIME Problems on Planar Graphs with few Variables.* Master's thesis, RWTH Aachen University, 2014.

[145] Neil Robertson and Paul D. Seymour. Graph minors. IX. Disjoint crossed paths. *Journal of Combinatorial Theory, Series B*, 49(1):40–77, 1990.

[146] Neil Robertson and Richard Vitray. Representativity of surface embeddings. In Bernhard Korte, László Lovász, Hans J. Prömel, and Alexander Schrijver, editors, *Paths, Flows and VLSI-Layout*, pages 293–328. Springer Verlag, 1990.

[147] Karem A. Sakallah. Symmetry and satisfiability. In *Handbook of Satisfiability*, pages 289–338. 2009.

[148] Pascal Schweitzer. *Problems of unknown complexity: graph isomorphism and Ramsey theoretic numbers.* PhD thesis, Universität des Saarlandes, Saarbrücken, 2009.

[149] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011.

[150] Sharadchandra Shankar Shrikhande. The uniqueness of the L_2 association scheme. *The Annals of Mathematical Statistics*, 30(3):781–798, 1959.

[151] Symmetry vs Regularity: the first 50 years since Weisfeiler-Leman stabilization. `https://www.iti.zcu.cz/wl2018/`. Accessed: 2020-01-10.

[152] Gottfried Tinhofer. Graph isomorphism and theorems of Birkhoff type. *Computing*, 36(4):285–300, 1986.

[153] Gottfried Tinhofer. A note on compact graphs. *Discrete Applied Mathematics*, 30(2–3):253–264, 1991.

[154] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.

[155] Jacobo Torán. On the hardness of graph isomorphism. *SIAM Journal on Computing*, 33(5):1093–1108, 2004.

[156] William T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 13:743–767, 1963.

[157] Moshe Y. Vardi. The complexity of relational query languages. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 137–146, San Francisco, CA, USA, May 1982.

[158] Moshe Y. Vardi. On the complexity of bounded-variable queries. In *Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 266–276, San Jose, CA, USA, May 1995.

[159] Oleg Verbitsky. Planar graphs: logical complexity and parallel isomorphism tests. In *Proceedings of the Twenty-Fourth Annual Symposium on Theoretical Aspects of Computer Science*, volume 4393 of *Lecture Notes in Computer Science*, pages 682–693, Aachen, Germany. Springer, February 2007.

[160] Klaus Wagner. Über eine Eigenschaft der ebenen Komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.

[161] Boris Weisfeiler. *On construction and identification of graphs*, volume 558 of *Lecture Notes in Mathematics*. Springer, 1976.

[162] Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series 2*, 1968. English translation by Grigory Ryabov available at `https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf`.

[163] Hassler Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54:150–168, 1932.