The combinatorics of minimal unsatisfiability: connecting to graph theory

Hoda Abbasizanjani

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Doctor of Philosophy.

Department of Computer Science
Swansea University

August 2021

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed .....[redacted].. (candidate)
Date ....27/08/2021..

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed .....[redacted]... (candidate)
Date ....27/08/2021..

Statement 2

I hereby give my consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ....[redacted].. (candidate)
Date ....27/08/2021..

Abstract

Minimally Unsatisfiable CNFs (MUs) are unsatisfiable CNFs where removing any clause destroys unsatisfiability. MUs are the building blocks of unsatisfiability, and our understanding of them can be very helpful in answering various algorithmic and structural questions relating to unsatisfiability. In this thesis we study MUs from a combinatorial point of view, with the aim of extending the understanding of the structure of MUs. We show that some important classes of MUs are very closely related to known classes of digraphs, and using arguments from logic and graph theory we characterise these MUs.

Two main concepts in this thesis are isomorphism of CNFs and the implication digraph of 2-CNFs (at most two literals per disjunction). Isomorphism of CNFs involves renaming the variables, and flipping the literals. The implication digraph of a 2-CNF F has both arcs $(\neg a \to b)$ and $(\neg b \to a)$ for every binary clause $(a \vee b)$ in F.

In the first part we introduce a novel connection between MUs and Minimal Strong Digraphs (MSDs), strongly connected digraphs, where removing any arc destroys the strong connectedness. We introduce the new class DFM of special MUs, which are in close correspondence to MSDs. The known relation between 2-CNFs and implication digraphs is used, but in a simpler and more direct way, namely that we have a canonical choice of one of the two arcs. As an application of this new framework we provide short and intuitive new proofs for two important but isolated characterisations for nonsingular MUs (every literal occurs at least twice), both with ingenious but complicated proofs: Characterising 2-MUs (minimally unsatisfiable 2-CNFs), and characterising MUs with deficiency 2 (two more clauses than variables).

In the second part, we provide a fundamental addition to the study of 2-CNFs which have efficient algorithms for many interesting problems, namely that we provide a full classification of 2-MUs and a polytime isomorphism decision of this class. We show that implication digraphs of 2-MUs are "Weak Double Cycles" (WDCs), big cycles of small cycles (with possible overlaps). Combining logical and graph-theoretical methods, we prove that WDCs have at most one skew-symmetry (a self-inverse fixed-point free anti-symmetry, reversing the direction of arcs). It follows that the isomorphisms between 2-MUs are exactly the isomorphisms between their implication digraphs (since digraphs with given skew-symmetry are the same as 2-CNFs). This reduces the classification of 2-MUs to the classification of a nice class of digraphs.

Finally in the outlook we discuss further applications, including an alternative framework for enumerating some special Minimally Unsatisfiable Subclause-sets (MUSs).

Acknowledgements

Firstly I would like to thank my examiners, Professor Olaf Beyersdorff and Dr Ulrich Berger. Their comments and questions helped shape this thesis into a better and more well presented whole and for that I am very grateful.

Working on this thesis has been a challenging, rewarding and wonderful experience. I am deeply grateful to the people who have shared it with me:

- To my supervisor, Oliver Kullmann, for his guidance, support, enthusiasm, and encouragements. His insights (technical and otherwise) have been invaluable to me; and he has helped me become not only a better researcher, but a better writer and a more productive person.

- To my amazing parents, Afkham and Hadi, for their unwavering love and support throughout my entire life, and for their faith in me and teaching me to be as ambitious as I wanted. It would be impossible for me to express my gratitude towards them in mere words.

- To my dear brother, Hossein, for his kindness and unconditional support. He has been a pillar in my life for as long as I can remember.

- And to my wonderful husband and best friend, Meghdad, for his love, patience and endless support over this often tricky journey. Without his tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

Finally, this research has been fully funded by EPSRC and the College of Science at Swansea University, whom I am very grateful for this great opportunity.

Contents

Chapter 1

Introduction

1.1 The satisfiability problem

The propositional satisfiability problem (SAT) is the problem of determining whether for a propositional formula there exists an assignment to boolean variables such that the formula evaluates to true. SAT is known to be NP-complete (independently proved by Cook [36] and Levin [113]), and so all problems in the complexity class NP can be transformed into SAT and solved with a SAT solver. There is no known algorithm that solves each SAT instance in polynomial time in the size of the instance description, and it is unlikely that such algorithm exists (unless P=NP). However over the past two decades SAT and SAT solving have been studied extensively, and due to the development of many successful SAT solvers, SAT has now various applications in hardware and software design and verification, artificial intelligence, cryptography, database systems, machine vision, and VLSI design and testing. For overviews of SAT solvers see [99] and the Handbook chapters [39] and [75], while a general overview of SAT is the Handbook chapter [55].

For satisfiable problems SAT solvers can provide a short proof (that is, a satisfying assignment), while for unsatisfiable problems a (complete) solver will have to certify that there is no satisfying assignment. Establishing unsatisfiability is generally done using some kind of proof system (such as resolution). For a recent overview of the proof systems underlying current approaches to SAT solving see [30].

1.2 Minimal unsatisfiability

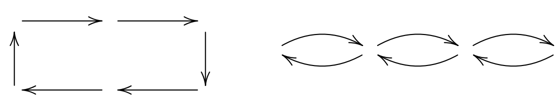
While SAT solvers can decide the satisfiability of propositional formulas, in the case of unsatisfiable formulas typically no explanation of the causes of unsatisfiability (that is, small unsatisfiable sub-formulas) is given. Explanation of "unsatisfiability" (also called "inconsistency" and "infeasibility") is often valuable and has a wide range of practical applications. For example, formulation of a typical design task as an instance of SAT falls into two categories: One, a propositional formula is formed such that a feasible design is obtained when the formula is satisfiable. In such case, unsatisfiability in the design context implies a negative result, and needs further analysis of the causes of unsatisfiability to obtain a feasible design. The other, a propositional formula is formed such that unsatisfiability is the desired result. This often happens when we want to modify the system, e.g., to reduce its cost. In this scenario, understanding causes of unsatisfiability helps to identify which part of the system should not be changed, and which one can be modified (or possibly removed). Other applications include SAT-based model checking ([34], [21], [124]), FPGAs routing ([126], [128]), artificial intelligence ([147], [26], [65]), operations research ([9], [5]), unsatisfiability-based MaxSAT algorithms ([122], [56]) and constraint programming ([31], [73]).

Consider a propositional formula F in Conjunctive Normal Form (CNF), that is, a conjunction of disjunctions of literals. Then F is minimally unsatisfiable if F is unsatisfiable and any of its sub-formulas is satisfiable. Minimally Unsatisfiable CNFs (MUs), are considered as providing a *single cause* of unsatisfiability. In general a CNF contains many MUs, then called Minimally Unsatisfiable Sub-formulas (MUSs), so contains *many causes* of unsatisfiability. Finding some reasonable MUSs or enumerating all of them has been the topic of several researches. The earliest work on finding MUSs mainly focused on identification of a single MUS or a small set of MUSs. With improvements in the scalability of MUS identification techniques, finding all MUSs has found practical applications ([119], [120], [117], [12]). However, since there can be up to exponentially many MUSs with respect to the size of the formula, their complete enumeration might be intractable. Therefore, several "online" MUS enumeration algorithms have been proposed which identify MUSs gradually, one by one, and thus identify at least some MUSs even in the intractable cases ([11], [118], [16], [127], [15]). Another approach is to compute the union of all MUSs, as it summarizes all the causes of unsatisfiability for a given formula ([125]). For an overview of existing algorithms for computing MUSs and their applications see [123], [14], [17].

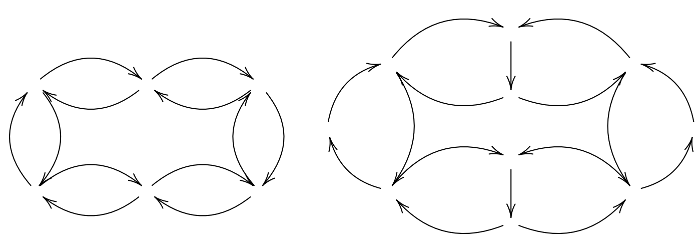
1.3 Motivation

The main motivation of this work is to provide a deeper understanding of the structure of minimally unsatisfiable CNFs as they are building blocks for understanding MUSs, and they are also interesting in their own right, as the hardest unsatisfiable formulas. Another motivation of this thesis is the fascinating relations of minimal unsatisfiability and combinatorics. There are strong connections in both directions between minimal unsatisfiability and combinatorics. Overviews on various connections between MUs and combinatorics and their applications can be found in [79] and in the Introduction of [109].

In this thesis we consider minimal unsatisfiability from a combinatorial point of view, and we will show that one can understand many subclasses of MUs directly (and actually characterise them precisely) via their connections with combinatorics. For example a collection of minimal strong digraphs (strongly connected digraphs where removing any arcs destroys this property), shown below, plays an important role in characterising two basic subclasses of MUs. When considering vertices as variables, and arcs as implications between these variables (e.g., $a \to b$), it is easy to see that in the following digraphs and their corresponding 2-CNFs (at most two literals per disjunction) all variables are equivalent (for the corresponding 2-CNF of the left and right digraphs see Examples 5.2.4 and 5.2.16, respectively). Then the equivalences of variables together with the requirement that these variables do not have the same value in the left digraph (cycle), and the equivalence of the starting point and the negated end point in the right digraph (dipath), yield unsatisfiable CNFs. Furthermore since removing any arc destroys this property, these CNFs are minimally unsatisfiable.



A different class of graphs, called weak-double-cycles (Definition 6.5.1), corresponds to minimally unsatisfiable 2-CNFs, yielding their complete characterisation. Below are some examples of weak-double-cycles (see Section 6.1 for the corresponding 2-CNFs). In the above digraphs we only considered positive implications, while in the following digraphs we have both implications (i.e., $a \to b$ and $\neg b \to \neg a$).



1.4 Overview of the literature on MUs

With respect to complexity, it is shown in [130] that the decision problem whether a CNF is minimally unsatisfiable is D^P-complete. The complexity class D^P, introduced in [131], is defined as the set of problems which can be described as the difference of two NP-problems. Therefore it is of interest which natural subclasses of MUs have an easier decision problem. The main complexity measure for MUs is the "deficiency", the difference of the number of clauses and the number of variables, where the classes of MUs with fixed deficiency have polytime recognition (independently proved by [91] and [52]). While the classification of MUs with fixed deficiency is a main theme for this thesis, in Sections 1.4.1, 1.4.2, 1.4.3 we will give an overview of this line of research.

Another important class of MUs with polytime decision is that of 2-CNF cases. The class of 2-CNFs has a linear algorithm for satisfiability testing, first shown in [50] and [8]. 2-CNFs are interesting and worthy of investigation for many reasons. For examples there are two-way connections between 2-CNFs and graph theory, and many significant combinatorial problems can be reduced to 2-SAT. A fundamental overview of 2-CNFs, their underlying boolean function and their applications is given in [38, Chapter 5]. In Section 1.4.4 we will review basic results regarding 2-CNFs mainly focusing on researches related to unsatisfiable and minimally unsatisfiable 2-CNFs, while a main contribution of this thesis is to provide a full classification of minimally unsatisfiable 2-CNFs in Chapter 6.

Other examples of polytime classes of MUs are Horn formulas (each disjunction has at most one positive literal) and hitting CNFs (every two disjunctions have a clash). Horn formulas are the basis of logic programming ([86]), and have various applications in expert systems, deductive databases, artificial intelligence and machine learning. The minimal unsatisfiability problem for Horn CNFs can be solved in quadratic time as the satisfiability is decidable in linear time, first proved in [47] using unit-resolution (the resolution rule where at least one of the clauses involved is a unit-clause). A generalisation of Horn formulas is renamable Horn CNFs, meaning that a change in the sign of some variables results in an equivalent Horn CNF. It is shown in [7] that the satisfiability problem for renamable Horn CNFs is solvable in linear time. An overview of Horn CNFs, their applications can be found in the Handbook chapter [55], while Horn functions and their applications are reviewed in [38, Chapter 6]. In Section 4.6.3 we will discuss the class of renamable Horn MUs and their characterisation.

Hitting CNFs as DNFs (Disjunctive Normal Form, disjunction of conjunctions of literals) are known as "orthogonal" or "disjoint" DNFs (referring to the fact that no two of conjunctions can take value 1 simultaneously). Unlike for 2-CNFs and Horn formulas, the polytime satisfiability (and so minimal unsatisfiability) of hitting CNFs is not related to the resolution calculus but purely to counting the number of falsifying total assignments of the given formula ([72]). Hitting CNFs and their DNF representation have been playing an important role for boolean functions and a detailed overview of their applications is given in [38, Chapter 7]. In Chapter 4 we will discuss some characterisations of hitting MUs.

Before continuing with the overview, we introduce a few basic notations. We consider CNFs as clause-sets, finite sets of clauses, where a clause is a finite and clash-free set of literals, and a literal is either a variable or a negated/-complemented variable. We use $\perp$ to denote the empty clause. The class of MUs as clause-sets is formally denoted by $\mathcal{MU}$, while "MU" is used in text in a substantival role. $\mathcal{MU}' \subset \mathcal{MU}$ is the set of "nonsingular" MUs, that is, the set of $F \in \mathcal{MU}$ such that every literal occurs at least twice. The number of clauses of a clause-set F is $c(F)$, the number of (occurring) variables is $n(F)$, and the deficiency is $\delta(F) := c(F) - n(F) \in \mathbb{Z}$ (Definition 2.1.3). The class of MUs with fixed deficiency $\delta = k$ is denoted by $\mathcal{MU}_{\delta=k}$. DP-reduction (Definition 2.6.1, also known as "variable elimination") of a clause-set F on a variable

v replaces all clauses in F containing v by their (non-tautological) resolvents on v. Finally, like in the DIMACS file format for clause-sets, we use natural numbers for variables and non-zero integers for literals. So the clause $\{-1, 2\}$ stands for the usual clause $\{\overline{v_1}, v_2\}$, where we just got rid off the superfluous variable-symbol "v". In propositional calculus, this would mean $\neg v_1 \vee v_2$, or, equivalently, $v_1 \to v_2$.

1.4.1 Minimal unsatisfiability and deficiency

The basic notion for investigations on the combinatorics of minimally unsatisfiable clause-sets (MUs) F is the deficiency $\delta(F)$ (Definition 2.1.3). The notion of "deficiency" for clause-sets was introduced in [54], while the concept had been used in [137]. The basic fact that for MUs F we have $\delta(F) \geq 1$ was shown in [4] (as "Tarsi's Lemma"), and later in [18].

Study of the decision complexity for classes $\mathcal{MU}_{\delta=k}$ with fixed deficiency $k \geq 1$, started in [45] where it was shown that the class of MUs with deficiency 1 is polytime decidable (quadratic time). In [146] some subclasses of MUs with deficiency 3 and 4 were shown to be polytime decidable. Later in [78] it was shown that the decision whether "$F \in \mathcal{MU}_{\delta=k}$?" is in NP, based on the upper bound $2^{k-1} \cdot n(F)^2$ for minimal resolution refutations for $F \in \mathcal{MU}_{\delta=k}$ obtained in [77] ($n(F)$ is the number of variables in F). Furthermore the author conjectured that the classes $\mathcal{MU}_{\delta=k}$ have polytime decision. In [78] the class of MUs with deficiency 2 was shown to be polytime decidable. Finally the conjecture that the classes $\mathcal{MU}_{\delta=k}$ are polytime decidable was proved true in [91] (based on searching for a resolution refutation), and independently in [52] (based on the fact that the search for a satisfying truth assignment can be restricted to certain assignments which correspond to matchings in bipartite graphs), and then improved proofs from [52] were presented in [51]. Furthermore it was shown in [142] that whether $F \in \mathcal{MU}_{\delta=k}$ can be decided in time $O(2^k \cdot n(F)^4)$. Therefore the class $\mathcal{MU}_{\delta=k}$ is fixed-parameter tractable in the parameter k in the sense of [48].

"Classification" of MUs is concerned with determining all the "isomorphism types" of MUs with fixed deficiency $k \geq 1$, that is, an easily accessible catalogue of the essentially different elements of $\mathcal{MU}_{\delta=k}$. The starting point of the investigation into the structure of $\mathcal{MU}_{\delta=k}$ is the basic fact, shown in [91], that $F \in \mathcal{MU}$ with $n(F) \geq 1$ contains a variable occurring positively and negatively each at most $\delta(F)$ times, that is, a variable of degree (the number of variable occurrences) at most $2\delta(F)$. So the minimum variable degree (Definition 2.1.5) is at most $2\delta(F)$ (sharper bounds are obtained in [109]). A major use of the variables of minimum degree is in proofs of properties of MUs, where we use "splitting" (instantiating a variable by both truth values 0, 1; Definition 4.4.1). We want to split on a variable occurring as few times as possible so that we have control over the changes imposed by the substitution. In Section 4.4 we will discuss some applications of such variables for investigating the structure of MUs.

The most basic MUs are those with deficiency 1, i.e., $F \in \mathcal{MU}_{\delta=1}$. The structure of the class $\mathcal{MU}_{\delta=1}$ is very well-known, and the earliest result regarding this class is the characterisation of its "saturated" elements in [4] (called "strongly minimal unsatisfiable formulas" there). The class of saturated MUs, denoted by $\mathcal{SMU} \subset \mathcal{MU}$ (Definition 4.1.4), is the class of MUs where adding any literal to any clause yields a satisfiable clause-set (this terminology was introduced in [53]; also called "maximal" in [85]). Later in [45] it was shown that an MU F has deficiency 1 iff DP-reduction (Definition 2.6.1) for variables of degree 2 yields the empty clause $\perp$. Therefore this whole class is explained by the expansion rule, which is the reverse of such DP-reduction. Furthermore in [45] this class was characterised via matrices (based on the so-called "basic matrices" from [44]).

In [91] the structure of clause-sets $F \in \mathcal{MU}_{\delta=1}$ was characterised as binary trees. Also a nice characterisation of the class of saturated $F \in \mathcal{MU}_{\delta=1}$ ($\mathcal{SMU}_{\delta=1}$) via (full) binary trees was obtained in [91, Lemma C.5], where it was shown that these MUs are exactly the clause-sets introduced in [37]. These two characterisations were then generalised in [102]. An important subclass of $\mathcal{MU}_{\delta=1}$ is the class of minimally unsatisfiable renamable Horn clause-set. It is well-known that for an element F of this subclass there exists an input-resolution tree yielding the empty clause ([70]), and in [45] the fact that $\delta(F) = 1$ was first established (Section 4.6.3 for a detailed overview).

For deficiencies $k \geq 2$ the classes $\mathcal{MU}_{\delta=k}$ contain singular (MUs with a variable occurring exactly once positively or exactly once negatively) and nonsingular MUs. However so far only classification of the nonsingular cases has been investigated in the literature (see the Handbook chapter [79] for a general overview). The nonsingular elements of $\mathcal{MU}_{\delta=2}$ have been characterised in the literature and we will give a detailed overview in Section 1.4.3, while a main contribution of this thesis is to provide a new short proof via a novel connection to graph theory in Chapter 5. In [110] a subclass of $\mathcal{MU}_{\delta=3}$ has been characterised, and for $k \geq 4$ the structure of classes $\mathcal{MU}_{\delta=k}$ is unknown.

We conclude this section by a short overview of the class of unsatisfiable hitting clause-sets (every pair of clauses has a clash). In [94] it is shown that unsatisfiable hitting clause-sets are MUs, while by [94] they are saturated. Therefore the set of unsatisfiable hitting clause-sets of fixed deficiency $k \geq 1$ is a subset of $\mathcal{SMU}_{\delta=k} \subset \mathcal{MU}_{\delta=k}$. In [91] the case of $k = 1$ has been shown to be exactly the class $\mathcal{SMU}_{\delta=1}$, and in [78] and [110] unsatisfiable hitting clause-sets of deficiency 2 and 3 have been characterised.

1.4.2 The program of "classifying MUs"

As already mentioned, the main goal of this work is to develop a deeper understanding of minimal unsatisfiability. In general it is a very complicated problem to determine what is the "classification" of a collection of objects. In the case of MUs we approach this problem as follows: we classify a subclass $\mathcal{C} \subseteq \mathcal{MU}$ "fully" if we have a full grasp on its elements, which includes a complete understanding of the isomorphism types involved. An isomorphism between two clause-sets is a renaming of variables and potentially flipping some literals (see

Section 2.4). The study of the "isomorphism problem" for some polytime subclasses $\mathcal{C} \subseteq \mathcal{MU}$ (that is, for given $F, F' \in \mathcal{C}$ decide whether F is isomorphic to F') has been carried out in [78, 81, 108, 102, 110, 82, 91] and in the Handbook chapter [79].

A reasonable approach for classifying MUs is to consider three basic levels of understanding of isomorphism types. The hardest level with respect to understanding is "GI-completeness", that is, the class is as complex as the class of all (undirected) graphs via ordinary graph isomorphism. The graph isomorphism problem (GI) is known to be in NP, while it is an open problem whether GI is NP-complete (see Section 3.2 for definition and details).

The next complexity level is to assume that a class $\mathcal{C}$ should have at least "polytime isomorphism decision" in order to be understandable, i.e., there exists a polytime algorithm which for inputs $F, F' \in \mathcal{C}$ decides whether F is isomorphic to F'. However this level might not provide full understanding if $\mathcal{C}$ has superpolynomially many isomorphism types. The third level, which is the easiest case and yields full understanding of the isomorphism types, is to have "efficient isomorphism type determination" (EID). We will discuss the precise formulation of EID in Section 7.3, but intuitively it means that the isomorphism types of $\mathcal{C}$ can be efficiently enumerated (without repetitions), and for any given $F \in \mathcal{C}$ its isomorphism type can be determined in polynomial time.

It is shown in [81] that the isomorphism problem for MUs with fixed deficiency $k \geq 1$ is GI-complete, and that even the class of minimally unsatisfiable Horn clause-sets (which is a subset of $\mathcal{MU}_{\delta=1}$) is still GI-complete. Therefore in order to understand their isomorphism types, a reduction is needed for the general classes $\mathcal{MU}_{\delta=k}$, or we consider some restricted classes of MUs (e.g., 2-CNF MUs or hitting cases). Furthermore, reductions are an important tool for understanding MUs: first we concentrate on understanding (only) reduced cases, and then we extend to other cases.

The main conjecture regarding MUs layered by deficiency is the "Finite-Patterns Conjecture" ([109]), stating that for fixed deficiency $k \geq 1$ there are only finitely many "patterns" in $\mathcal{MU}_{\delta=k}$, given a certain basic reduction. This implies that an MU F in general might not present the "reason of unsatisfiability" clearly, as there might be some "trivial details", and some reduction is essential to remove these trivialities. According to the formulation of the Finite-Patterns Conjecture in [109], singular variables (variables occurring in one sign only once, Definition 4.3.1) are the trivial details, and complete reduction with respect to singular variables is fundamental to see the basic patterns. The study of "singular DP-reduction" (DP-reduction for singular variables) for MU is the topic of [108], containing results related to "confluence" (that is, the result of the reduction is independent of the choices made during the nondeterministic reduction and the result is always unique). Singular DP-reduction for MUs maintains minimal unsatisfiability and deficiency, and a main result, shown in [108], is that in general the number of variables in the result is unique. In Section 4.5 we will go through the basic results regarding singular DP-reduction of MUs, and in Section 7.3 we will discuss some variations of the Finite-Patterns Conjecture and the related results.

So far the Finite-Patterns Conjecture has been shown for deficiencies $k \leq 2$. For deficiency $k = 1$, singular DP-reduction is confluent to $\{\perp\}$ ([45]) and so we consider the reasons for unsatisfiability as given by the elements of this class as "pure trivialities". For deficiency 2 it is shown that singular DP-reduction for the elements of $\mathcal{MU}_{\delta=2}$ is confluent modulo isomorphism to a single pattern, namely cycle digraphs over some variables where the parameter is the length of cycles ([78], [108]). So $F \in \mathcal{MU}_{\delta=2}$ contains a unique (and possibly hidden) reason of unsatisfiability. We have already seen an example of these cycle digraphs in Section 1.3. In Chapter 5 we will see more on that type of patterns, which are minimal strong digraphs (MSDs), and in this way we explore what "pattern" means. However for $k \geq 3$ the Finite-Patterns Conjecture is an open question and even the precise meaning of pattern is not known.

A special case of singular DP-reduction is "1-singular DP-reduction", the DP-reduction for "1-singular variables", variables occurring positively and negatively exactly once. By [108, Section 5] 1-singular DP-reduction for an MU is confluent, yielding a unique "normalform", while the importance of this reduction for understanding the isomorphism types of MUs has not been considered in the literature. In Chapter 6 we will show that 1-singular DP-reduction for 2-CNF MUs is closely related to the homeomorphism of their implication graphs. Furthermore we obtain full understanding of these homeomorphism types (and so normalforms), namely that they correspond to the class of binary strings called "bracelets", or "turnover necklaces" ([61], [25]; see Definition 6.5.8).

For the general classes $\mathcal{MU}_{\delta=k}$ with $k \geq 3$, a question is that which level of reduction is right to establish finitely many patterns. Singular DP-reduction could be too weak or too strong. For example for 2-CNF MUs (short as 2-MUs), singular DP-reduction is confluent modulo isomorphism to precisely one basic pattern, namely dipaths ([82]) and the parameter is the length of dipaths (see Section 3.3 for definition of dipaths). However by a weaker reduction, 1-singular DP-reduction, we obtain another pattern, namely the homeomorphism types which correspond to binary bracelets.

In Chapter 6 we obtain a polytime isomorphism decision together with a complete classification for the class of 2-MUs with fixed deficiency. Furthermore we show that the class of 2-MUs of deficiency m without 1-singular variables corresponds to the class of binary bracelets with m strings, which has super-polynomially many isomorphism types. Known examples of such class are $\mathcal{SMU}_{\delta=1}$ and the set of marginal elements of $\mathcal{MU}_{\delta=1}$ (MUs where removing any literal occurrences yields a satisfiable clause-sets, Definition 4.2.2), denoted by $\mathcal{MMU}_{\delta=1}$ (recall that we need these restriction as the general complexity is GI-complete). The special case of $\mathcal{SMU}_{\delta=1}$ can be naturally identified with the class of all full binary trees ([91]) which has a polytime isomorphism decision, and for given number of vertices n we have super-polynomially many isomorphism types. By [103] the isomorphism types of $F \in \mathcal{MMU}_{\delta=1}$ correspond exactly to the isomorphism types of trees, where the class of trees has a polytime isomorphism decision. All these three classes seem to have similar exponential growth rate for asymptotic complexity, and their isomorphism types can be presented by well-known classes of graphs. Furthermore they all have polytime isomorphism decision for singular cases under restriction.

For classes $\mathcal{SMU}_{\delta=1}$ and $\mathcal{MMU}_{\delta=1}$, given their nature, their classification is recursive (see Section 4.6 for more details); while different from them is the 2-MU cases which we show that the isomorphism types are not recursive. Another difference is that for $\mathcal{SMU}_{\delta=1}$ and $\mathcal{MMU}_{\delta=1}$ singular DP-reduction collapse everything. But for 2-MUs we show that the one main pattern in abstract way is WDCs, and intermediate cases obtained by 1-singular DP-reduction (normal-forms) correspond to binary bracelets while still having finitely many isomorphism types given the important concept of deficiency (the length of the string), and the nonsingular cases have a unique isomorphism type for fixed deficiency.

Finally, Table 1.1 shows an overview of main classes of MUs where the isomorphism types are already known, or will be characterised in this thesis.

1.4.3 The two fundamental characterisations

As already mentioned, the most basic class of MUs is $\mathcal{MU}_{\delta=1}$, where singular DP-reduction is confluent to $\{\bot\}$, and so in a sense only "trivial" reasoning takes place for the elements of this class (as in general removing all singular variables in an MU can be done in polytime). Also somewhat surprisingly, this class covers all minimally unsatisfiable renamable Horn clause-sets. The "real" reasoning starts at the next level, where there are two classes, namely MUs of deficiency 2 (i.e., $\mathcal{MU}_{\delta=2}$), and minimally unsatisfiable 2-CNFs, short 2-MUs.

The central family of MUs with deficiency 2 is the following MUs which have been introduced in [78] (see Section 5.2.1 for more details on these formulas):

$$\begin{aligned} \boldsymbol{\mathcal{F}_n} \quad &:= \quad (v_1 \rightarrow v_2) \wedge (v_2 \rightarrow v_3) \wedge \ \dots \ \wedge (v_{n-1} \rightarrow v_n) \wedge (v_n \rightarrow v_1) \\ & \qquad \wedge (v_1 \vee \dots \vee v_n) \wedge (\neg v_1 \vee \dots \vee \neg v_n). \end{aligned}$$

When using natural numbers as variables (e.g., using $\{-1, 2\}$ instead of $\{\overline{v_1}, v_2\}$ to get rid of the superfluous variable-symbol v), the MUs $\mathcal{F}_n$ take the following form as clause-sets for $n \in \mathbb{N}$, $n \geq 2$:

$$\begin{aligned} \mathcal{F}_n \quad &:= \quad \{\{-1, 2\}, \{-2, 3\}, \dots, \{-(n-1), n\}, \{-n, 1\}, \\ & \qquad \{1, \dots, n\}, \{-1, \dots, -n\}\} \in \mathcal{MU}_{\delta=2}. \end{aligned}$$

As shown in the seminal paper [78], the nonsingular elements of $\mathcal{MU}_{\delta=2}$ are exactly (up to isomorphism, of course) the MUs $\mathcal{F}_n$. The elimination of singular variables by singular DP-reduction is not confluent in general for MUs. However in [108] it is shown, that we have confluence up to isomorphism for deficiency 2.

2-MUs have been also studied in the literature and we give an overview in Section 1.4.4, while here we only focus on the nonsingular cases. In the report [82], as in [112] (called "$F^{(2)}$"), the following 2-MUs have been introduced (recall that $a \leftrightarrow b$ is $(a \rightarrow b) \wedge (b \rightarrow a)$).

$$\boldsymbol{\mathcal{B}_n} := v_1 \leftrightarrow v_2 \leftrightarrow v_3 \leftrightarrow \dots \leftrightarrow v_{n-1} \leftrightarrow v_n \leftrightarrow \neg v_1.$$

Table 1.1: An overview of main characterisations of MUs, from the top to down complexity increases. $\mathcal{MU}'$ is the set of nonsingular MUs and $\mathcal{MU}^+$ is the set of MUs with no 1-singular variables. $\mathcal{SMU}$ and $\mathcal{MMU}$ are the sets of saturated and marginal MUs, respectively. $\mathcal{RHO}$ is the set of renamable Horn clause-sets, $2\text{–}\mathcal{MU}$ is the set of 2-CNF MUs, while $2\text{–}\mathcal{MU}' = 2\text{–}\mathcal{MU} \cap \mathcal{MU}'$. Also, $\mathcal{DFM}$ is the set of MUs consisting of two full positive and negative clauses plus mixed binary clauses. For the definition of efficient isomorphism type determination (EID) see Section 7.3.

Classes	Complexity	Isomorphism types	Connection to graph theory
$\mathcal{RHO} \cap \mathcal{SMU}$	EID	Full binary trees with the Horton-Strahler number ≤ 1 (Lemma 4.6.22)	The structure tree (Lemma 4.6.3)
$\mathcal{MU}'_{\delta=2}$	EID	Cycle digraphs (Corollary 5.3.2)	The positive implication digraph (Definition 5.2.3)
$2\text{–}\mathcal{MU}'$	EID	Dipaths (Corollary 5.4.8)	The positive implication digraph
$2\text{–}\mathcal{MU} \cap \mathcal{MU}^+$	Polytime (Corollary 6.7.1)	Binary bracelets (nonlinear weak-double-cycles, Corollary 6.7.5 and Lemma 6.5.11)	The implication digraph (Definition 6.2.1)
$\mathcal{SMU}_{\delta=1}$	Polytime (Corollary 4.6.9)	Full binary trees (Lemma 4.6.3)	The structure tree
$\mathcal{MMU}_{\delta=1}$	Polytime (Corollary 4.6.39)	Trees (Lemma 4.6.38)	The conflict graph (Definition 4.6.31)
$2\text{–}\mathcal{MU}$	Polytime (Corollary 6.7.3)	Weak-double-cycles (Corollary 6.7.1)	The implication digraph
$\mathcal{DFM}$	GI-complete	Minimal strong digraphs (Theorem 5.2.9)	The positive implication digraph

And as clause-sets (using natural numbers as variables) MUs $\mathcal{B}_n$ have the following form for $n \in \mathbb{N}$, $n \geq 2$ (see Section 5.2.2 for more details):

$$\mathcal{B}_n = \{\{-1,2\},\{1,-2\},\ldots,\{-(n-1),n\},\{n-1,-n\},\{-1,-n\},\{1,n\}\}$$

In the technical report [82] it is shown that the nonsingular 2-MUs are exactly the 2-MUs $\mathcal{B}_n$ (up to isomorphism). By [108] it follows again that we have confluence modulo isomorphism for singular DP-reduction on 2-MUs.

In Chapter 5 we will provide new proofs for these two characterisations which reveal their underlying common structure.

1.4.4 2-CNFs and minimal unsatisfiability

Restrictions to the lengths of the clauses resp. the "terms" (in DNFs the conjunctions) were studied especially with the advent of automated theorem proving in the middle of the 20th century. 2-CNFs were also called "Krom formulas" in the context of first-order logic. The first explicit proof of a polytime SAT decision for 2-CNFs (via resolution closure, in the context of first-order logic) seems to be in [87]. Another proof for propositional logic was pointed out in the seminal paper [36] using the Davis-Putnam algorithm in [43]. Later the bound was improved by the linear time algorithms of [50] and [8] (the latter even for quantified 2-CNFs). For an overview on the dual form of (general) 2-DNFs and their underlying boolean functions, called "quadratic functions" (which are constant zero for unsatisfiable 2-CNFs resp. constant one for tautological 2-DNFs), see [38, Chapter 5]. Irredundant 2-CNFs (no clause can be removed without changing the underlying boolean function) are studied in [116], mostly concentrating on satisfiable cases.

A classical connection of SAT to combinatorics is random satisfiability. For random 2-CNFs with c clauses and n variables, the satisfiability threshold was proven for the critical density $\frac{c}{n} = 1$ by [35] and independently by [63]. That is, a random 2-CNF F with $\frac{c}{n} > 1$ is unsatisfiable with high probability, while F with $\frac{c}{n} < 1$ is satisfiable with high probability. A more precise picture of phase transition and its scaling window for random 2-CNFs was achieved in [24], and an overview is given in [46].

2-CNFs are close to renamable Horn formulas in the following sense: It is well-known that satisfiable 2-CNFs are renamable Horn formulas. Regarding unsatisfiable cases, [70] established the basic fact that every unsatisfiable formula is renamable Horn iff it is refutable by unit-resolution, and thus we see that an unsatisfiable 2-CNF without a unit-clause is not renamable Horn. Below we look at the case with a unit-clause, which in the MU-case is renamable Horn.

We now turn to unsatisfiable 2-CNFs. The concept of the "implication digraph" (Definition 6.2.1) was introduced in [8], and an overview is given in [38, Section 5.4.3]). For a 2-CNF F with variables $v_1, \ldots, v_n$, which does not contain the empty clause, the vertices of the implication digraph are the literals $v_1, \ldots, v_n, \overline{v_1}, \ldots, \overline{v_n}$ of F. A clause $\{x, y\}$ in F yields the two arcs $\overline{x} \to y$,

$\overline{y} \to x$ in the implication digraph; these arcs become one in case $x = y$. [8] showed that a 2-CNF F is unsatisfiable iff the implication digraph of F has a strongly connected component containing a literal and its complement. Every unsatisfiable 2-CNF has a variable v such that via input-resolution one can derive v and $\overline{v}$ ([90, Lemma 5.6]), where the length of each chain of resolution steps is at most the number of variables. Considering resolution complexity, [28] obtained a polytime algorithms for finding a smallest tree-like resolution refutation for 2-CNFs, while [29] provided a polytime algorithm for finding a smallest general resolution refutation, both using implication digraphs of 2-CNFs. Study of some incomplete refinements of resolution has been carried out in [27], namely so-called "read-once" resolution refutation and its variations, and the authors have investigated the complexity of finding such resolution refutations.

A different study of graphs related to 2-CNFs is the recent [74] which is mainly interested in distinguishing satisfiability and unsatisfiability. For a 2-CNF F they obtain a graph by first applying some form of preprocessing of F which destroys information on isomorphism types, but the obtained graph can distinguish satisfiable and unsatisfiable F. In this thesis we are only interested in 2-MUs. Before considering the literature here, we mention that MUSs of 2-CNFs have been studied in [28], showing how to compute shortest MUSs in polytime.

Running through all clauses and testing their irredundancy, the minimal unsatisfiability problem for 2-CNFs can be decided in quadratic time. Just expressing the above special form of resolution refutations for 2-CNFs, [116, Lemma 19] states a general pattern of 2-MUs. We have already mentioned that 2-MUs $\mathcal{B}_n$ have been used in the report [82] and also in [112]. Regarding the number of clauses $c(F)$ for a 2-MU F, [116], [40] and [112] provide some bounds, while we will give the sharp bound $c(F) \leq 4n(F) - 2$, attained exactly for the $\mathcal{B}_n$, in Chapter 5.

A 2-MU F with a unit-clause has a unit-resolution refutation, since otherwise unit-clause propagation would yield a non-trivial autarky (a partial assignment satisfying some clauses and not touching the others). Thus as mentioned before F is renamable Horn, and so $\delta(F) = 1$. In [27] the isomorphism types of 2-MUs with a unit-clause are determined, leaving open the determination of (singular) 2-MUs of deficiency 1 without unit-clauses. 2-MUs of higher deficiencies are all 2-uniform, and only the nonsingular cases have been characterised in the literature, which were discussed in the previous section.

1.5 Contributions

1.5.1 Understanding MUs via connection to MSDs

The proofs of the characterisation of nonsingular elements of $\mathcal{MU}_{\delta=2}$ in [78], and characterisation of nonsingular 2-MUs in [82] are both an impressive feat. A main contribution of this thesis, presented in Chapter 5, is to introduce a new unifying reasoning scheme for these two important but isolated results based on graph theory. This reasoning scheme considers MUs with two parts. The clauses of the "core" represent AllEqual, that is, all variables are equal. The two "full monotone clauses" (a clause over all positive literals and a clause over all negative literals) represent the negation of AllEqual. This is the new class FM ("full monotone") of MUs, which still is as complex as all of MU. So we demand that the reasoning for AllEqual is graph theoretical, arriving at the new class DFM ("D" for digraph).

Establishing AllEqual on the variables happens via strong digraphs (SDs), where between any two vertices there is a path. For minimal reasoning we use minimal strong digraphs (MSDs), where every arc is necessary. Then, just demanding to have an MU with two full monotone clauses, while the rest are binary clauses, is enough to establish precisely MSDs. The two most fundamental classes of MSDs are the cycle digraphs (those strong digraphs where every vertex is linear) and the dipaths (i.e., the directed versions of the path graphs where every undirected edge is replaced by two directed arcs, for both directions). The cycle digraphs are at the heart of MUs with deficiency 2, while the dipaths are at the heart of 2-MUs.

After this general overview, we now gain a deeper understanding of how the characterisations nonsingular $F \in \mathcal{MU}_{\delta=2}$ and nonsingular 2-MUs work:

(I) For a nonsingular $F \in \mathcal{MU}_{\delta=2}$ the main step is to make the connection to the class DFM. We show that, up to flipping of signs, it actually already holds that F is DFM. Then using graph-theoretical reasoning we show that the MSDs of minimal deficiency (the difference of the number of arcs and the number of vertices) 0 are the cycles which correspond to DFMs $\mathcal{F}_{n(F)}$, up to isomorphism:

$$\begin{tikzcd} 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[d, dashed] \\ n \arrow[u] & n-1 \arrow[l] & n-3 \arrow[l] \end{tikzcd}$$

(II) For a nonsingular 2-MU F the main step is to show that there must exist exactly one positive and one negative clause and these can be saturated to full positive resp. full negative clauses, and so F' is DFM. Then via graph-theoretical reasoning we show that the only MSDs G such that the corresponding DFMs can be obtained as partial saturations of nonsingular 2-MUs are the dipaths, since we can only have two linear vertices in G (i.e., vertices of in- and out-degree 1). That is, nonsingular 2-MUs are $\mathcal{B}_{n(F)}$ up to isomorphism.

$$\begin{tikzcd} 1 \arrow[r, bend left] & 2 \arrow[l, bend left] \arrow[r, bend left] & 3 \arrow[l, bend left] \arrow[r, bend left, dashed] & n-1 \arrow[l, bend left, dashed] \arrow[r, bend left] & n \arrow[l, bend left] \end{tikzcd}$$

Finally an overview on the main results of Chapter 5 is given in Figure 1.1.

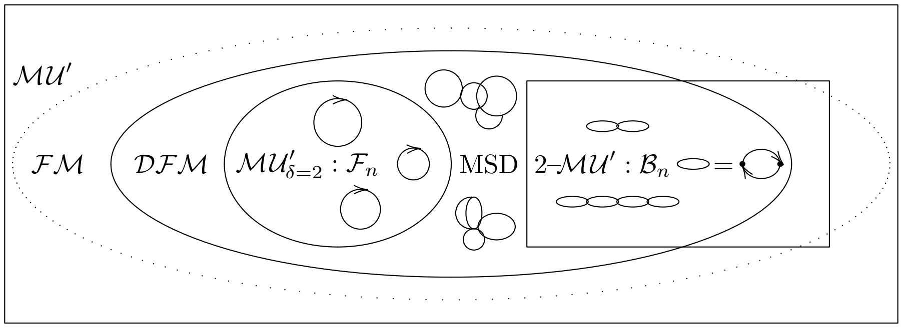


Figure 1.1: Cycle digraphs at the heart of $\mathcal{MU}_{\delta=2}$ (MUs of deficiency 2), and dipaths at the heart of 2–$\mathcal{MU}$ (2-MUs). $\mathcal{MU}'$ is the set of nonsingular MUs, $\mathcal{FM}$ is the set of MUs with full monotone clauses. $\mathcal{DFM}$ is the subset of $\mathcal{FM}$ where every clause other than the full monotone clauses is a mixed binary clause. For definitions of $\mathcal{F}_n, \mathcal{B}_n$ see Section 1.4.3.

1.5.2 The Splitting Ansatz

A main method for analysing $F \in \mathcal{MU}$ is splitting (Definition 4.4.1): choose an appropriate variable v in $F \in \mathcal{MU}$, set v to $0, 1$ and obtain F_0, F_1, analyse them, and lift the information obtained back to F (see Section 4.4 for more details). An essential point here is to have $F_0, F_1 \in \mathcal{MU}$, but in general this is not the case. The approach of [83, Section 3] is to remove clauses appropriately in F_0, F_1 and study various conditions in order to obtain some minimally unsatisfiable sub-formulas $F'_0 \subseteq F_0$ and $F'_1 \subseteq F_1$ and to characterise them.

Our method (used for characterising nonsingular $F \in \mathcal{MU}_{\delta=2}$ and nonsingular 2-MUs) is based on the observation, that if a clause say in F_0 became redundant, then $\overline{v}$ can be added to this clause in F, while still remaining MU, and so the assignment $v \to 0$ then takes care of the removal. This is the essence of saturation, with the advantage that we are dealing again with MUs. A saturated MU is characterised by the property, that for any variable, splitting yields two MUs ([102]). For classes like 2-MUs, which are not stable under saturation, we introduce "local saturation" (Definition 4.1.7), which only saturates the variable we want to split on. In our application for characterising nonsingular 2-MUs, the local saturation uses all clauses, and this is equivalent to a "disjunctive splitting" as surveyed in [27, Definition 8]. On the other hand, for deficiency 2 the method of saturation is more powerful, since we have stability under saturation, and the existence of a variable occurring twice positively and twice negatively holds *after* saturation. Splitting needs to be done on nonsingular variables (Definition 4.3.1), so that the deficiency becomes strictly smaller in F_0, F_1, as we want these instances to be "easy", to know them well. In both of our cases we obtain renamable Horn clause-sets. For deficiency 2 we exploit, that the splitting involves the minimal number of clauses, while for 2-MUs we exploit that the splitting involves the maximal number of clauses after local saturation. In order to get say F_0 "easy", while F is "not easy", the part which gets removed, which is related to F_1, must have special properties.

1.5.3 Classification of 2-MUs

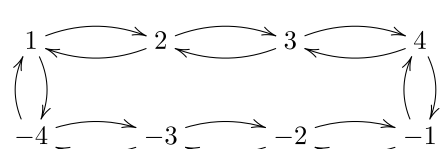
As already mentioned, it is known that the isomorphism problem for classes $\mathcal{MU}_{\delta=k}$ and even for the class of minimally unsatisfiable renamable Horn clause-sets (which is a subset of $\mathcal{MU}_{\delta=1}$) are GI-complete. In Chapter 6 we give the first example of a class of restricted but still rich MUs, namely 2-MUs, with polytime isomorphism decision. More importantly, we obtain a very precise overview of their structure. The simplest variables in any MU are 1-singular variables. The subclass of 2-MUs without 1-singular variables corresponds exactly to the class of binary bracelets. This shows that there are exponentially many isomorphism types of 2-MUs (in dependency on the number of variables).

The starting point of our investigations in Chapter 6 are the most basic 2-MUs, the nonsingular 2-MUs. As already discussed, a nonsingular 2-MU F with $n \geq 2$ variables is isomorphic to $\mathcal{B}_n$ (recall Section 1.4.3). For general MUs holds, that removing singular variables via DP-reduction preserves minimal unsatisfiability and deficiency (see Section 4.5). Thus the above says, that singular DP-reduction for a 2-MU of deficiency $k \geq 2$ yields some 2-MU isomorphic to $\mathcal{B}_k$. To refine this, 1-singular DP-reduction for an MU F is confluent ([108]), yielding the "non-1-singular normalform" of F. So we obtain a generation process for all 2-MUs of deficiency k: start with $\mathcal{B}_k$, first reverse non-1-singular DP-reductions, and then reverse 1-singular DP-reductions. Now how do the clause-sets generated in this way look?

As it turns out, they correspond closely to a nice class of digraphs, called "weak-double-cycles" (WDCs; studied in [136]). For this, the concept of the implication digraph of a 2-CNF F is needed. A basic observation is that the reversal of singular DP-reduction for 2-MUs corresponds to the following two graph-theoretical operations:

- "Splitting a vertex" replaces a vertex x by two new vertices u, v and adds the arc from u to v: u collects the ingoing arcs of x, and v collects the outgoing arcs.
- "Splitting an arc" adds a midpoint (a new vertex) to an arc.

Performing reverse non-1-singular DP-reduction for a 2-MU F corresponds to splitting vertices, while the reverse of 1-singular DP-reduction corresponds to splitting arcs (see Section 6.6 for details). WDCs are obtained from "double m-cycles" (undirected cycles of length m converted to digraphs, which are the implication digraphs of $\mathcal{B}_{\frac{m}{2}}$ for even m) by splitting some (possibly zero) vertices and arcs. For example the implication digraph of $\mathcal{B}_4$, shown below, is a double 8-cycle. We have already seen some other examples of WDCs in the first section of the Introduction.



Not all WDCs correspond to clause-sets (at all), and digraphs in general do not allow negation (complementation), and this further ingredient is needed. The corresponding concept indeed exists in the literature on digraphs under the name "skew-symmetry" (a self-inverse fixed-point free anti-symmetry, reversing the direction of arcs, Definition 6.2.4).

The implication digraph of a 2-CNF F, in its exact form, with the literals as vertices, allows exact reconstruction of F, since the arcs faithfully encode the clauses, while the vertices-as-literals reveal full information on the complement-relation between vertices. The complement-relation between vertices is provided *explicitly* by a skew-symmetry. Digraphs might have no skew-symmetry (then they do not correspond to 2-CNFs at all), or they might have many (then they correspond to several 2-CNFs). Digraphs with *given* skew-symmetry are basically the same as 2-CNFs. We show that WDCs have at most one skew-symmetry. That is, there is at most one way to add complementation of the vertices to a WDC and obtain a 2-CNF. The main result of Chapter 6 follows easily: The isomorphisms between 2-MUs are *exactly* the isomorphisms between their implication digraphs. So we reduce determining isomorphisms/automorphisms (the self-isomorphisms) of 2-MUs to a purely graph-theoretical problem between (nice) digraphs. It follows that the automorphisms of a 2-MU F with deficiency $k \geq 2$ form a subgroup of the Dihedral group with $4k$ elements (the group of symmetries of a regular polygon with $2k$-sides, which includes $2k$ rotations and $2k$ reflections), and this allows efficient enumeration and counting of isomorphism types of 2-MUs.

1.6 Summary and publications

The fundamental new definitions and concepts introduced in Chapter 5, as well as associated theorems and applications are:

1. Definitions 5.1.1 introduces the basic new class $\mathcal{FM} \subset \mathcal{MU}$, which consists of all $F \in \mathcal{MU}$ containing the full positive clause and the full negative clause.

2. Theorem 5.1.9 demonstrates the D^P-completeness of this new class.

3. Theorem 5.1.8 characterises $\mathcal{FM}$ as the set of clause-sets F with two full complementary clauses where the core (other clauses) is a CNF-realisation of the AllEqual boolean function on the variables of F and is irredundant.

4. Definition 5.2.1 introduces the most important new class of this chapter $\mathcal{DFM} \subset \mathcal{FM}$, which consists of all $F \in \mathcal{FM}$ such that the clauses of the core are binary. Then Definition 5.2.3 defines the "positive implication digraph" (only the implications between positive literals) for $F \in \mathcal{DFM}$ where the mixed binary clauses of the core are exactly the arcs of the corresponding positive implication digraph.

5. Theorem 5.2.9 is a main contribution of this chapter which shows that there is a bijection between $\mathcal{DFM}$ and the set of MSDs via two formations given in this theorem. The correspondence between DFMs and the powerful world of MSDs enables us to use the strength of graph-theoretical reasoning once we connect a class of MUs to DFMs.

6. As the first applications of these new classes and their connection to graph theory, we give unifying and insightful proofs for two fundamental results:

   - Theorem 5.3.1 and Corollary 5.3.2 characterise the nonsingular elements of $\mathcal{MU}_{\delta=2}$ by showing that these MUs are precisely the elements of $\mathcal{DFM}$ (up to isomorphism) with deficiency 2, and then connection to cycle digraphs is established by graph-theoretical reasoning.
   - Theorems 5.4.5, 5.4.7 together with corollary 5.4.8 characterise the nonsingular 2-MUs by showing that these MUs can be locally saturated to those elements of $\mathcal{DFM}$ (up to isomorphism) whose corresponding MSDs are dipaths.

7. Another (conceptual) contribution of this chapter, used in the proof of the above characterisations, is the strengthening of the Splitting Ansatz by saturation, in two forms, full saturation for $\mathcal{MU}_{\delta=2}$, and local saturation (Definition 4.1.7 and Lemma 4.4.4), which is introduced for the first time, for 2-MU.

8. Corollary 5.4.9 gives the sharp bound for the number of clauses in 2-MUs, attained exactly for the $\mathcal{B}_n$.

In Chapter 6 we study all 2-MUs, and the key contributions are as follows:

1. Theorem 6.3.10 provides full classification of the isomorphism types of 2-MUs with a unit-clause which were implicitly handled in [27], and also the 2-uniform cases of deficiency 1 which is new. Then Theorem 6.3.11 states the exact number of isomorphism types of 2-MUs with deficiency 1.

2. Lemma 6.4.7 shows that the smoothing process for digraphs is strongly related to 1-singular DP-reduction for 2-MUs. The concept of smoothing is known in graph theory, but we exploit it in more details, showing new connections between graph theory and propositional logic.

3. Lemma 6.5.3 states a polytime isomorphism decision for WDCs.

4. Lemma 6.5.11 shows that the isomorphism types of WDCs with no linear vertices, and so the homeomorphism types of WDCs (obtained by the smoothing process) corresponds exactly to binary bracelets.

5. Theorem 6.6.4 states that the implication digraphs of 2-MUs with deficiency $k \geq 2$ are WDCs with $2k$ small cycles.

6. Theorem 6.6.9 is a main result of this chapter and shows the uniqueness of skew-symmetry for WDCs by combining arguments from logic and graph theory, where both fields contribute significantly to the proof.

7. Theorem 6.6.10 is the second major result of this chapter, showing that for 2-MUs F, F' the set of isomorphisms between F, F' is equal to the set of isomorphisms between their implication digraphs. That is, we reduced determining isomorphisms of 2-MUs to a purely graph-theoretical problem between simple digraphs.

8. Then we obtain a variety of applications in Section 6.7:

   - Corollary 6.7.2 characterises the automorphism groups of WDCs with $2k$ small cycles and so $F \in \mathcal{MU}_{\delta=k}$ as subgroups of the Dihedral group with $4k$ elements. Therefore the isomorphism problem for 2-MUs of deficiency k is polytime decidable in k (Corollary 6.7.3.
   - Corollary 6.7.4 states that the number of isomorphism types of $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ with n variables is $\Theta(n^{3k-1})$.
   - Corollary 6.7.5 shows that the smoothing of skew-symmetric WDCs corresponds exactly to the canonical normalform of 2-MUs F, and so the isomorphism types of these normalforms are in one-to-one correspondence with binary bracelets of length k.

In the outlook (Chapter 7) we discuss how the understanding of the structure of MUs may be utilised to obtain an alternative framework for enumeration of MUSs. As an application, in Theorem 7.1.4 we show that for a 2-CNF plus two full complementary clauses, all contained MUSs can be enumerated in incremental polynomial time.

The publications related to this thesis are as follows:

- The content of Chapter 5 was published as a conference paper ([1]) at SAT 2018.
- The initial results concerning classification of 2-MUs (Chapter 6) first appeared in the technical report [2].
- Another paper featuring all the content of Chapter 6 is in preparation.

Furthermore this work has been presented by the author at the BCTCS 2018.

We conclude this section by an overview of this thesis. Preliminaries such as notations and basic concepts are introduced in Chapters 2 (propositional logic) and 3 (graph theory). Chapter 4 describes the main techniques and tools for systematic investigation of the structure of MUs, which have been used for the main results in the later sections. Chapter 5 introduces a deep connection between a class of MUs and MSDs which yields characterisation of two important classes of MUs. Chapter 6 covers 2-MUs and their classification via connection to graph theory. Finally in Chapter 7 the contributions of the thesis are summarised, and remaining conjectures and open questions, as well as some applications concerning enumeration MUSs of are discussed.

Chapter 2

Preparations on logic

The aim of this chapter is to establish basic terminologies and notations on clause-sets used in this thesis. We start off with definition of boolean variables, clauses and clause-sets, and their basic notations in Section 2.1, and then present the definition of partial assignments, satisfiability and unsatisfiability in Section 2.2. In Section 2.3 irredundancy and minimal unsatisfiability of clause-sets are discussed. Isomorphism and automorphism of clause-sets are considered in Section 2.4. Finally the last sections 2.5 and 2.6 provide the definition of the resolution operation and DP-reduction.

2.1 From variables to clause-sets

The infinite set of variables is denoted by $\boldsymbol{\mathcal{VA}}$. For every variable $v \in \mathcal{VA}$ its **domain** is a finite and non-empty set, denoted by $\boldsymbol{D_v} \neq \emptyset$. A **boolean variable** v has domain $D_v = \{0, 1\}$. Literals are boolean variables $v \in \mathcal{VA}$ and their complementations $\overline{v}$; and the underlying variable of a literal x is denoted by $\mathbf{var}(\boldsymbol{x}) \in \mathcal{VA}$. The set of all literals is $\boldsymbol{\mathcal{LIT}}$. For a set L of literals, $\overline{\boldsymbol{L}} := \{\overline{x} : x \in L\}$ is the elementwise complementation, $\mathbf{var}(\boldsymbol{L}) := \{\mathrm{var}(x) : x \in L\}$ is the set of variables of L and $\mathbf{lit}(\boldsymbol{L}) := \{x \in \mathcal{LIT} : \mathrm{var}(x) \in \mathrm{var}(L)\}$ is the set of all possible literals over $\mathrm{var}(L)$.

Example 2.1.1 *For $L_1 = \{a, b, c\} \subset \mathcal{LIT}$ and $L_2 = \{a, \overline{a}, \overline{b}, c\} \subset \mathcal{LIT}$ we have* $\mathrm{lit}(L_1) = \mathrm{lit}(L_2) = \{a, \overline{a}, b, \overline{b}, c, \overline{c}\}$.

Literals $x, y \in \mathcal{LIT}$ **clash** (or "have a conflict") if $x = \overline{y}$. For a set of literals $L \subseteq \mathcal{LIT}$ we say that L is **clash-free** if there are no $x, y \in L$ which clash. A **clause** is defined as a finite and clash-free (i.e., non-tautological) set of literals. The set of variables occurring in a clause C is $\mathbf{var}(\boldsymbol{C})$.

A **clause-set** is a finite set of clauses, and we use $\boldsymbol{\mathcal{CLS}}$ for the set of all clause-sets. The empty clause-set is denoted by $\top := \emptyset \in \mathcal{CLS}$ and the empty clause by $\bot := \emptyset$. Clause-sets are interpreted as propositional formulas in Conjunctive Normal Forms (CNFs), conjunctions of disjunctions of literals.

Example 2.1.2 *A CNF representation of a clause-set* $F := \{\{a,b\},\{\overline{c},d\}\} \in \mathcal{CLS}$ *is* $F = (a \vee b) \wedge (\neg c \vee d)$*; and a CNF* $F' := a \wedge (\neg b \vee c) \wedge (c \vee \neg b) \wedge (\neg a \vee b \vee \neg c)$ *as a clause-set is* $F' = \{\{a\},\{\overline{b},c\},\{\overline{a},b,\overline{c}\}\} \in \mathcal{CLS}$*.*

We use the notation "$A \mathbin{\dot\cup} B$" to denote disjoint union, i.e., $A \mathbin{\dot\cup} B = A \cup B$ when $A \cap B = \emptyset$. By $\mathbb{N} = \{1,2,\ldots\}$ we denote the set of natural numbers, while $\mathbb{N}_0 := \mathbb{N} \mathbin{\dot\cup} \{0\}$. We use natural numbers $\mathbb{N}$ as variables (i.e., $\mathbb{N} \subseteq \mathcal{VA}$) as in the DIMACS format. This makes creating certain examples easier, since we can use integers different from zero as literals, and complementation is represented as negation (i.e., $\overline{v} = -v$). For example the clause $\{-1,2\}$ stands for the usual clause $\{\overline{v_1}, v_2\}$ (in propositional calculus, $\neg v_1 \vee v_2$, or, equivalently, $v_1 \to v_2$).

Definition 2.1.3 *For a clause-se* $F \in \mathcal{CLS}$*:*

1. $\mathbf{var}(\boldsymbol{F}) := \bigcup_{C \in F} \mathrm{var}(C) \subset \mathcal{VA}$ *(the set of variables occurring in* F*).*
2. $\mathbf{lit}(\boldsymbol{F}) := \mathrm{var}(F) \cup \overline{\mathrm{var}(F)}$ *(the set of all possible literals over the variables in* F*).*
3. $\boldsymbol{n(F)} := |\mathrm{var}(F)| \in \mathbb{N}_0$ *(the number of variables in* F*).*
4. $\boldsymbol{c(F)} := |F| \in \mathbb{N}_0$ *(the number of clauses in* F*).*
5. $\boldsymbol{\ell(F)} := \sum_{C \in F} |C| \in \mathbb{N}_0$ *(the number of literal occurrences in* F*).*
6. $\boldsymbol{\delta(F)} := c(F) - n(F) \in \mathbb{Z}$ *is the* ***deficiency*** *of* F*.*

Example 2.1.4 *The only clause-sets with no variables are* $\top, \{\bot\}$*. For* $F := \{\bot\}$ *we have* $c(F) = 1$ *and so* $\delta(F) = 1 - 0 = 1$*, while* $\delta(\top) = 0 - 0 = 0$*. Now consider another clause-set* $F' := \{\{x\},\{\overline{x},y\}\} \in \mathcal{CLS}$ *(as a CNF,* $x \wedge (\neg x \vee y)$*). We have* $\mathrm{var}(F') = \{x,y\}$*,* $\mathrm{lit}(F') = \{x,\overline{x},y,\overline{y}\}$*,* $\ell(F') = 3$ *and* $n(F') = c(F') = 2$*, and so* $\delta(F') = 2 - 2 = 0$*.*

For a literal x, the literal-degree $\mathbf{ld}_{\boldsymbol{F}}(\boldsymbol{x}) := |\{C \in F : x \in C\}| \in \mathbb{N}_0$ is the number of clauses of F containing x, while the variable-degree of a variable v is $\mathbf{vd}_{\boldsymbol{F}}(\boldsymbol{v}) := \mathrm{ld}_F(v) + \mathrm{ld}_F(\overline{v}) \in \mathbb{N}_0$. A literal x is **pure** (also called "monotone literal") for F if $\mathrm{ld}_F(\overline{x}) = 0$.

Definition 2.1.5 *The* ***minimum variable degree*** *(or min-var-degree) is defined as the minimum of the variable degrees over all variables in* $F \in \mathcal{CLS}$ *and is denoted by* $\boldsymbol{\mu}\mathbf{vd}(\boldsymbol{F}) := \min_{v \in \mathrm{var}(F)} \mathrm{vd}_F(V) \in \mathbb{N}$ *for* $n(F) > 0$*, while* $\mu\mathrm{vd}(F) := +\infty$ *in case of* $n(F) = 0$*. Also, the set of variables of minimum variable degree in* F *is denoted by* $\mathbf{var}_{\boldsymbol{\mu}\mathbf{vd}}(\boldsymbol{F})$*.*

In Lemma 4.4.5 we will see an application of variables of minimum degree as a general tool for study of MUs, while in Chapters 5 and 6 we will use such variables for characterising various subclasses of MUs.

Example 2.1.6 *Consider a clause-set* $F := \{\{1\}, \{1, -2\}, \{-1, -2, 3\}\} \in \mathcal{CLS}$ *(as a CNF,* $v_1 \wedge (v_1 \vee \neg v_2) \wedge (\neg v_1 \vee \neg v_2 \vee v_3)$*). We have* $\mathrm{ld}_F(1) = 2$, $\mathrm{ld}_F(-1) = 1$, $\mathrm{ld}_F(2) = 0$, $\mathrm{ld}_F(-2) = 2$, $\mathrm{ld}_F(3) = 1$ *and* $\mathrm{ld}_F(-3) = 0$ *(so* $-2, 3$ *are pure literals for* F*). Also* $\mu\mathrm{vd}(F) = 1$ *with* $\mathrm{var}_{\mu\mathrm{vd}}(F) = \{3\}$.

A clause C is **positive** if $C \subset \mathcal{VA}$, while C is **negative** if $C \subset \overline{\mathcal{VA}}$, and C is **mixed** otherwise; a non-mixed clause is called **monotone**. A clause-set F is **uniform** resp. k-uniform, if all clauses of F have the same length resp. length k. By **2–$\mathcal{CLS}$** we denote the set of clause-sets $F \in \mathcal{CLS}$ such that for all clauses $C \in F$ holds $|C| \leq 2$. A **full clause** of a clause-set F is some $C \in F$ with $\mathrm{var}(C) = \mathrm{var}(F)$. A **full clause-set** is an $F \in \mathcal{CLS}$ where all $C \in F$ are full. In Example 2.1.6, the clause $\{-1, -2, 3\}$ is the only full clause in F. By $\boldsymbol{A_n}$ we denote the full clause-set consisting of the 2^n full clauses over variables $1, \ldots, n$ for $n \in \mathbb{N}_0$. So $A_0 = \{\bot\}$, $A_1 = \{\{-1\}, \{1\}\}$, and $A_2 = \{\{-1, -2\}, \{1, 2\}, \{-1, 2\}, \{1, -2\}\}$.

2.2 Partial assignments and unsatisfiability

Definition 2.2.1 *A* ***partial assignment*** *is a map* $\varphi : V \to \{0, 1\}$ *for some finite* $V \subset \mathcal{VA}$*. We use* $\mathbf{var}(\boldsymbol{\varphi}) := V$ *for the set of variables in* φ *and* $\mathbf{lit}(\boldsymbol{\varphi}) := \mathrm{lit}(\mathrm{var}(\varphi))$ *for the set of all possible literals over* $\mathrm{var}(\varphi)$*. The set of all partial assignments is* $\boldsymbol{\mathcal{PASS}}$.

For a fixed set of variables V*, a partial assignment* φ *with* $\mathrm{var}(\varphi) = V$ *is called a* ***total assignment*** *over* V.

A partial assignment φ satisfies a clause C iff φ satisfies at least one literal x in C (i.e., $\varphi(x) = 1$), and φ satisfies a clause-set F iff φ satisfies all clauses in F (recall that clause-sets are interpreted as CNFs, where a clause is considered as a disjunction of literals and a clause-set is considered as a conjunction of its clauses). We use for example $\langle x \to 1, y \to 0, z \to 0\rangle$ to denote a partial assignment that sets variables x to 1 and y, z to 0.

The application of partial assignments φ to $F \in \mathcal{CLS}$, denoted by $\boldsymbol{\varphi * F}$, yields the clause-set obtained from F by removing all clauses satisfied by φ, and then removing all falsified literals x from the remaining clauses (i.e., all literals x with $\varphi(x) = 0$ are removed). A partial assignment φ can be considered as a clause containing the set of falsified literals in $\mathrm{lit}(\varphi)$. For example for a variable x the partial assignment $\langle x \to 0\rangle$ as a clause is $\{x\}$, while $\langle x \to 1\rangle$ as a clause is $\{\overline{x}\}$. Then we say that a partial assignment φ (as a clause) satisfies a clause C if φ and C clash in a variable. Now we can define the operation of partial assignments on clause-sets as follows:

Definition 2.2.2 *For* $\varphi \in \mathcal{PASS}$ *and* $F \in \mathcal{CLS}$*:*

$$\varphi * F := \{C \setminus \varphi : C \in F \text{ and } C \cap \overline{\varphi} = \emptyset\} \in \mathcal{CLS}.$$

A **contraction** of a clause-set means that some previously unequal clauses become equal as a result of some operation, and so the number of clauses drops since we are dealing with clause-*sets*. We note that when applying partial assignment contractions can occur, and so more clauses might disappear than expected. Also more variables than just those in φ might disappear, since we consider only occurring variables. For more details on partial assignments and their operations see [79].

Remarks:

1. Simple properties of partial assignments:
   (a) $\varphi * (F \cup G) = \varphi * F \cup \varphi * G$.
   (b) $\bot \in F \Rightarrow \bot \in \varphi * F$.
   (c) $\varphi * \top = \top$.
   (d) $\varphi * \{C\} = \{\bot\}$ iff $C \subseteq \varphi$.
   (e) $\varphi * \{C\} = \top$ iff $C \cap \overline{\varphi} \neq \emptyset$.
   (f) $\varphi = \{x \in \mathcal{LIT} : \varphi * \{x\} = \{\bot\}\}$.
   (g) $\varphi * F = \top \Leftrightarrow \forall C \in F : C \cap \overline{\varphi} \neq \emptyset$.
   (h) $\bot \in \varphi * F \Leftrightarrow \exists C \in F : C \subseteq \varphi$.

Example 2.2.3 *Consider* $F := \{\{a, b\}, \{\overline{a}, c\}, \{a, b, d\}, \{\overline{c}, \overline{b}\}\} \in \mathcal{CLS}$.

- $\varphi := \langle c \to 0, d \to 0\rangle$ *as clause is* $\{c, d\}$ *and we obtain* $\varphi * F = \{\{a, b\}, \{\overline{a}\}\} \in \mathcal{CLS}$.
- *We have* $\langle a \to 0, b \to 0, c \to 1\rangle * F = \{a, b, \overline{c}\} * F = \{\bot, \{d\}\} \in \mathcal{CLS}$.
- *And* $\langle a \to 0, b \to 1, c \to 0\rangle * F = \{a, \overline{b}, c\} * F = \top \in \mathcal{CLS}$.

Definition 2.2.4 *A clause-set* F *is* ***satisfiable*** *if there is a partial assignment* φ *with* $\varphi * F = \top$*, otherwise* F *is* ***unsatisfiable****. We use* $\boldsymbol{\mathcal{SAT}} := \{F \in \mathcal{CLS} \mid \exists \varphi \in \mathcal{PASS} : \varphi * F = \top\}$ *for the set of all satisfiable clause-sets and* $\boldsymbol{\mathcal{USAT}} := \mathcal{CLS} \setminus \mathcal{SAT}$ *for the set of all unsatisfiable clause-sets. A partial assignment* $\varphi \in \mathcal{PASS}$ *with* $\varphi * F = \top$ *is called a* ***satisfying assignment*** *for* $F \in \mathcal{CLS}$.

Remarks:

1. $\top \in \mathcal{SAT}$ and $\{\bot\} \in \mathcal{USAT}$.
2. If $\bot \in F$, then $F \in \mathcal{USAT}$.
3. If $F \in \mathcal{USAT}$, the $\varphi * F \in \mathcal{USAT}$.
4. If $F \in \mathcal{SAT}$ and $F' \subseteq F$, then also $F' \in \mathcal{SAT}$.

2.3 Irredundancy and minimal unsatisfiability

Definition 2.3.1 *For $F, F' \in \mathcal{CLS}$ the **implication-relation** is defined as $\boldsymbol{F \models F'} \Leftrightarrow \forall \varphi \in \mathcal{PASS} : \varphi * F = \top \Rightarrow \varphi * F' = \top$ (F' is called a logical consequence of F). Two clause-sets $F, G \in \mathcal{CLS}$ are **logically equivalent** if $F \models G$ and $G \models F$.*

For a clause C we write $F \models C$ if $F \models \{C\}$, and we have $F \in \mathcal{USAT}$ iff $F \models \bot$. A clause-set F is **redundant** if there exists a clause $C \in F$ with $F \setminus C \models C$, while otherwise F is **irredundant** (also called "clause minimal" in [84]). In other words, $F \in \mathcal{CLS}$ is irredundant iff for every $C \in F$ there exists a total assignment φ which satisfies $F \setminus \{C\}$ while falsifying C. See [84] and [115] for a detailed studies of irredundant clause-sets. A clause-set F is called **subsumption-free** if F has no clauses like $C, D \in F$ where C subsumes D, i.e., $C \subset D$ ([98]). It is easy to see that irredundant clause-sets are subsumption-free.

Example 2.3.2 *The clause-set F in Example 2.2.3 is satisfiable and redundant (as $\{a, b\} \in F$ subsumes $\{a, b, d\} \in F$).*

And examples of unsatisfiable clause-sets are the full clause-sets A_n, $n \in \mathbb{N}_0$ which are also irredundant as removing any clause in A_n yields a satisfiable clause-set (note that for $A_0 = \{\bot\} \in \mathcal{USAT}$ removing the empty clause yields the empty clause-set $\top \in \mathcal{SAT}$).

A **minimally unsatisfiable clause-set** (MU), also called "minimal unsatisfiable clause-set/formulas", is some $F \in \mathcal{USAT}$ where removing any clause renders it satisfiable. That is, F is minimally unsatisfiable iff F is unsatisfiable and irredundant ([84]). It is well-known that for MUs F we have $\delta(F) \geq 1$, first shown in [4] ("Tarsi's Lemma"). We use $\boldsymbol{\mathcal{MU}}$ to denote the set of all unsatisfiable clause-sets, and $\boldsymbol{\mathcal{MU}_{\delta=k}}$ for the set of all MUs with fixed deficiency $k \geq 1$. Also **2–$\boldsymbol{\mathcal{MU}}$** $:= \mathcal{MU} \cap 2\text{–}\mathcal{CLS}$ is the set of 2-CNF MUs (short 2-MUs), while **2–$\boldsymbol{\mathcal{MU}_{\delta=k}}$** $:= 2\text{–}\mathcal{MU} \cap \mathcal{MU}_{\delta=k}$ denotes the set of 2-MUs with fixed deficiency $k \geq 1$. In Chapter 4 we will provide a fundamental review of minimally unsatisfiable clause-sets, while in Chapters 5 and 6 we will characterise some important subclasses of MUs.

Example 2.3.3 *For all full clause-sets A_n, $n \in \mathbb{N}_0$ we have $A_n \in \mathcal{MU}$ (see Example 2.3.2).*

2.4 Isomorphism of clause-sets

An isomorphism between two clause-sets (also called "boolean congruence" in [3] and "literal renaming" in [81]) is a permutation (or renaming) of variables and potentially flipping some literals. More precisely:

Definition 2.4.1 *An* ***isomorphism*** *from $F \in \mathcal{CLS}$ to $G \in \mathcal{CLS}$, denoted by $\alpha : \mathrm{lit}(F) \to \mathrm{lit}(G)$, is a bijection such that $\forall\, x \in \mathrm{lit}(F) : \alpha(\overline{x}) = \overline{\alpha(x)}$ and the clauses of F are precisely mapped to the clauses of G, that is, $\alpha(F) = \{\alpha(C) : C \in F\} = G$.*

Clause-sets $F, G \in \mathcal{CLS}$ are called ***isomorphic****, denoted by $\boldsymbol{F \cong G}$, if there exists an isomorphism from F to G.*

Example 2.4.2 *The only minimally unsatisfiable clause-set with no variable is $A_0 = \{\bot\}$, while any MU F with $n(F) = 1$ is isomorphic to $A_1 = \{\{1\}, \{-1\}\} \in \mathcal{MU}_{\delta=1}$. In Section 5.3 we will show that for any $F \in \mathcal{MU}_{\delta=2}$ with precisely two variables we have $F \cong A_2 = \{\{-1,-2\}, \{1,2\}, \{-1,2\}, \{1,-2\}\} \in \mathcal{MU}_{\delta=2}$. And it is easy to see that for any full clause-set F we have $F \cong A_{n(F)}$.*

Definition 2.4.3 *The* ***automorphisms*** *of $F \in \mathcal{CLS}$ are the isomorphisms from F to itself, also called* ***symmetries****.*

Example 2.4.4 *The trivial automorphism for any clause-set F is the identity map on $\mathrm{lit}(F)$. For $A_1 = \{\{1\}, \{-1\}\}$ the only non-trivial automorphism is flipping the literals. And for $A_2 = \{\{-1,-2\}, \{1,2\}, \{-1,2\}, \{1,-2\}\}$ flipping a literal, swapping the variables, or any combination of these is an automorphism.*

2.5 Resolution

The resolution operation, introduced in [132], is to replace two clauses containing complementary literals with a new clause implied by these clauses.

Definition 2.5.1 *Two clauses C, D are* ***resolvable*** *if they clash in exactly one variable v, i.e., $|C \cap \overline{D}| = 1$. Then v is called the* ***resolution variable****. For two resolvable clauses C, D the* ***resolvent*** *$\boldsymbol{C \diamond D} := (C \cup D) \setminus \{v, \overline{v}\}$ for $C \cap \overline{D} = \{v\}$ is the union of the two clauses minus the two clashing literals $v, \overline{v}$.*

Unit-resolution is the resolution rule where at least one of the clauses involved is a unit-clause. A **resolution refutation** of an unsatisfiable clause-set $F \in \mathcal{USAT}$ is to derive the empty clause $\bot$ from F by repeated applications of the resolution rule, and a **resolution tree** is an ordered rooted tree (see Section 3.3) formed by resolution operations. More precisely:

Definition 2.5.2 *A resolution tree is an ordered rooted tree, where every inner vertex has exactly two children, and every vertex is labelled with a clause such that the label of an inner vertex is the resolvent of the labels of its two parents. We write $T : F \vdash C$ if T is a resolution tree such that the root of T is labelled by C, while each clause labelling a leaf of T is element of F.*

A resolution tree is called **regular** if along every path from the root to some leaf no resolution variable occurs more than once.

Example 2.5.3 *For $F = \{\{a,b\},\{\overline{a},b\},\{a,\overline{b}\},\{\overline{a},\overline{b}\}\} \cong A_2 \in \mathcal{MU}_{\delta=1}$, a (regular) resolution tree $T : F \vdash \bot$ is shown below.*

$$\begin{tikzcd}[column sep=small]
\{a,b\} \arrow[dr, dash] & & \{a,\overline{b}\} \arrow[dl, dash] & & \{\overline{a},c\} \arrow[dr, dash] & & \{\overline{a},\overline{c}\} \arrow[dl, dash] \\
& \{a\} \arrow[drr, dash] & & & & \{\overline{a}\} \arrow[dll, dash] & \\
& & & \bot & & &
\end{tikzcd}$$

For an overview on resolution and its complexity see Sections 1.15 and 1.16 in [55].

2.6 DP-reduction

An important reduction for investigation of minimally unsatisfiable clause-sets is the DP-reduction (also called "Davis-Putnam resolution/reduction" and "variable elimination"), which was first introduced in [43] by Davis and Putnam. The DP-reduction of a clause-set F on a variable v is to remove all clauses in F containing v and replace them with all resolvents on v. More formally:

Definition 2.6.1 *The **DP-reduction** of $F \in \mathcal{CLS}$ on $v \in \mathcal{VA}$ is defined as:*

$\boldsymbol{\mathrm{DP}_v(F)} := \{C \in F : v \notin \mathrm{var}(C)\} \cup \{C \diamond D : C, D \in F \text{ and } C \cap \overline{D} = \{v\}\} \in \mathcal{CLS}$

Remarks:

1. We have $\mathrm{var}(\mathrm{DP}_v(F)) \subseteq \mathrm{var}(F) \setminus \{v\}$.
2. If two clauses C, D in F clash in more than one variables then when performing DP-reduction on one of these variables both clauses are removed. For example for $F := \{\{u,v\},\{\overline{u},\overline{v}\}\}$ we have $\mathrm{DP}_u(F) = \mathrm{DP}_v(F) = \top$.
3. It is well-known that $\mathrm{DP}_v(F)$ is logically equivalent to the existential quantification of v in F, i.e., $\mathrm{DP}_v(F)$ is satisfiable iff F is satisfiable ([108]).

DP-reduction is commutative ([104]), and in general does not maintain minimal unsatisfiability. In Section 4.5 we discuss an special case of DP-reduction which was first called "singular DP-reduction" in [97]. This reduction is the most harmless reduction for MUs and is used for understanding the underlying structure of MUs. See [108] for a full overview on DP-reduction and its applications in the search for MUSs and in practical SAT-algorithms.

Example 2.6.2 *For $F := \{\{v\},\{\overline{v}\}\}$ we have $\mathrm{DP}_v(F) = \{\bot\}$ and so F is unsatisfiable. Now consider the following examples:*

- *For $F' := \{\{x,y\},\{\overline{x},y\},\{\overline{y}\}\}$ we obtain $\mathrm{DP}_y(F') = \{\{x\},\{\overline{x}\}\} \cong F \in \mathcal{USAT}$. Thus $F' \in \mathcal{USAT}$.*
- *For $F'' := \{\{u,w,\overline{z}\},\{\overline{u},\overline{z}\},\{u\},\{\overline{u},w\}\}$ the DP-reduction on u yields $\mathrm{DP}_u(F'') = \{\{w,\overline{z}\},\{\overline{z}\},\{w\}\}$ where no further resolution is possible. So F'' is satisfiable.*

Chapter 3

Preparations on graph theory

This chapter provides basic terminologies and notations on graph theory. Section 3.1 contains the essential definitions for graphs, digraphs and multigraphs. Also strong connectivity of digraphs is defined in this section. Isomorphism and automorphism of digraphs and multigraphs are considered in Section 3.2. Finally, some basic examples of graphs and digraphs which will be used in this thesis are presented in Section 3.3.

3.1 Basic definitions

Definition 3.1.1 *A (finite)* ***graph*** *resp.* ***digraph*** *G is a pair (V, E), where $V(G) := V$ is a finite set of vertices and $E(G) := E$ is the set of edges resp. arcs defined as two-element subsets $\{a, b\} \subseteq V$ resp. pairs $(a, b) \in V^2$ with $a \neq b$.*

Note that we do not allow (self-)loops, and that there are no parallel edges resp. arcs (though there might be antiparallel arcs). A (di)graph G is a sub(di)graph of another (di)graph G' if $V(G) \subseteq V(G')$ and $E(G) \subseteq E(G')$.

A graph G is promoted to a digraph by $\mathbf{dg}(\boldsymbol{G}) := (V(G), \{(a, b), (b, a) : \{a, b\} \in E(G)\})$, converting every edge $\{a, b\}$ into two arcs $(a, b), (b, a)$. The conversion of a digraph G to its underlying graph (forgetting directions, and contracting antiparallel arcs into one edge) is denoted by $\mathbf{ug}(\boldsymbol{G})$.

Definition 3.1.2 *For a finite (di)graph G the* ***deficiency*** *is $\boldsymbol{\delta(G)} := |E(G)| - |V(G)| \in \mathbb{Z}$.*

The **in-degree** of a vertex $v \in V(G)$ of a digraph G is the number of arcs going into v, the **out-degree** is the number of outgoing arcs, and the **degree** of v is the sum of in- and out-degree. If G is a graph, then the degree of v is the number of vertices w adjacent to v (that is $|\{w \in V(G) : \{v, w\} \in E(G)\}| \in \mathbb{N}_0$).

Definition 3.1.3 *For a set V and $m \in \mathbb{N}_0$, let $\binom{V}{m} := \{S \subseteq V : |S| = m\}$ be the set of m–element subsets of V. A* ***multigraph*** *is a pair (V, E) where V is a set and $E : \binom{V}{1} \cup \binom{V}{2} \to \mathbb{N}_0$.*

A sub-multigraph G' of a multigraph G has $V(G') \subseteq V(G)$ and $\forall \{u, v\} \in \binom{V'}{1} \cup \binom{V'}{2} : E(G')(\{u, v\}) \leq E(G)(\{u, v\})$. A graph G is promoted to a multigraph $\mathbf{mg}(\boldsymbol{G})$ by using the same vertex-set $V(G)$, and using the characteristic function of $E(G) \subseteq \binom{V}{2}$, while the underlying graph $\mathbf{ug}(\boldsymbol{G})$ of a multigraph just forgets the multiplicities of edges and discards loops. A digraph G is converted to a multigraph $\mathrm{mg}(G)$ by forgetting the direction of arcs, while not contracting edges.

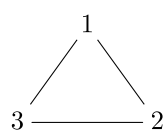
The degree of a vertex $v \in V(G)$ in a multigraph G, denoted by $\mathbf{deg}_{\boldsymbol{G}}(\boldsymbol{v})$, is the number of adjacent edges, that is,

$$\deg_G(v) := \sum_{w \in V(G)} E(G)(\{v, w\}) \in \mathbb{N}_0.$$

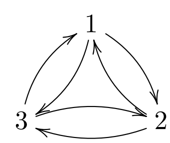
The set of neighbours of a vertex v in a multigraph G is $\mathbf{N}_{\boldsymbol{G}}(\boldsymbol{v}) := \{w \in V(G) : E(G)(\{v, w\}) \neq 0\} \subseteq V(G)$.

Example 3.1.4 *Consider a graph $G := (\{a, b, c\}, \{\{a, b\}, \{b, c\}, \{a, c\}\})$. The digraph $\mathrm{dg}(G)$ and the multigraph $\mathrm{mg}(\mathrm{dg}(G))$ are obtained as follows:*

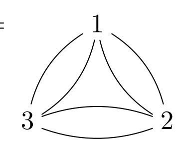
$G =$



$\mathrm{dg}(G) =$



$\mathrm{mg}(\mathrm{dg}(G)) =$



And we have $\delta(G) = 3 - 3 = 0$ while $\delta(\mathrm{dg}(G)) = 6 - 3 = 3$.

Definition 3.1.5 *A* ***linear vertex*** *in a (multi)graph G is a vertex $v \in G$ of degree 2, while a linear vertex in a digraph is a vertex of in- and out-degree 1.*

A digraph G is a **Strong Digraph** (SD), if G is strongly connected, i.e., for every two vertices $a, b \in V(G)$ there is a path from a to b. In an SD G with $|V(G)| \geq 2$ the in-degree and out-degree of the vertices are at least 1. A **Minimal Strong Digraph** (MSD) is an SD G, such that for every arc $e \in E(G)$ holds that $(V(G), E(G) \setminus \{e\})$ is not strongly connected. Every digraph G with $|V(G)| \leq 1$ is an MSD. Every MSD with $|V(G)| \geq 2$ has at least two linear vertices ([59]). For a recent overview of MSDs see [58]. In Example 3.1.4 the digraph $\mathrm{dg}(G)$ is an SD but not an MSD. A main contribution of Chapter 5 is to show the strong correspondence between an important class of MUs and MSDs, which yields some fundamental characterisations of MUs.

3.2 Isomorphisms

Definition 3.2.1 *For two digraphs G_1, G_2, an* ***isomorphism*** *from G_1 to G_2 is a bijection $f : V(G_1) \to V(G_2)$ such that $f(E(G_1)) = \{(f(a), f(b)) : (a, b) \in$*

$E(G_1)\} = E(G_2)$; if G_1, G_2 are graphs, then the condition is that $f(E(G_1)) = \{\{f(a), f(b)\} : \{a, b\} \in E(G_1)\} = E(G_2)$. If there is an isomorphism between G_1 and G_2, then we write $\boldsymbol{G_1 \cong G_2}$.

*An **automorphism** (also called **symmetry**) of a (di)graph G is an isomorphism from G to itself.*

A map f is an isomorphism from graph G to graph G' iff f is an isomorphism from $\mathrm{dg}(G)$ to $\mathrm{dg}(G')$. Every isomorphism f from a digraph G to a digraph G' is also an isomorphism from $\mathrm{ug}(G)$ to $\mathrm{ug}(G')$ (but not vice versa).

The **graph isomorphism problem (GI)** is the computational problem of determining whether two finite graphs are isomorphic. This problem is in the complexity class NP, but it is neither known to be NP-complete nor known to be polynomial time. Therefore a new complexity class GI has been defined as the set of problems with a polynomial time reduction to the graph isomorphism problem. Over time, increasingly strong conjectural evidence has been found that the graph isomorphism problem is not NP-complete, and in a recent breakthrough it was proved that this problem is solvable in quasi-polynomial time ([10]). So if the graph isomorphism problem is solvable in polynomial time, GI would equal P; and if the problem is NP-complete, GI would equal NP and all problems in NP would be solvable in quasi-polynomial time. Isomorphism of clause-sets can be naturally reduced in polytime to graph isomorphism, and **GI-completeness** of such isomorphism problem means additionally that also the graph isomorphism problem can be reduced to it. Other examples of GI-complete problems are hypergraph isomorphism and 2-CNF isomorphism.

Definition 3.2.2 *An isomorphism from a multigraph G to a multigraph G' is a bijection $f : V(G) \to V(G')$ with $\forall\, v, w \in V(G) : E(G')(\{f(v), f(w)\}) = E(G)(\{v, w\})$.*

Every isomorphism $f : G \to G'$ between multigraphs is also an isomorphism $f : \mathrm{ug}(G) \to \mathrm{ug}(G')$ between the underlying graphs (but not vice versa). A map f is an isomorphism from a graph G to a graph G' iff f is an isomorphism from $\mathrm{mg}(G)$ to $\mathrm{mg}(G')$. Every isomorphism $f : G \to G'$ between digraphs is also an isomorphism $f : \mathrm{mg}(G) \to \mathrm{mg}(G')$ (but not vice versa).

3.3 Basic examples

A **cycle graph** is a connected graph, where every vertex is linear (so it has at least 3 vertices). The standardised cycle graph $\mathbf{CG}_{\boldsymbol{n}}$ for $n \geq 3$ is defined as follows:

$$\begin{aligned} V(\mathrm{CG}_n) &:= \{1, \dots, n\}, \\ E(\mathrm{CG}_n) &:= \{\{i, i+1\} : i \in \{1, \dots, n-1\}\} \mathbin{\dot\cup} \{\{1, n\}\}. \end{aligned}$$

A **double cycle** is a digraph isomorphic to $\mathrm{dg}(\mathrm{CG}_n)$ (see Section 6.5 for properties of double cycles). A **cycle multigraph** allows additionally for length

2 (two vertices and two parallel edges) and length 1 cycle (one vertex with a loop). A cycle in a (multi)graph G is a (sub)multigraph which is isomorphic to some cycle (multi)graph.

A **cycle digraph** is a strong digraph, where every vertex is linear (so it has at least two vertices). Cycle digraphs have deficiency zero, and some of their properties are studied in Chapter 5 (Section 5.2.1). A cycle in a digraph is a sub-digraph which is isomorphic to some cycle digraph. The standardised cycle digraph $\mathbf{CD}_{\boldsymbol{n}}$ for $n \geq 2$ is as follows:

$$
\begin{aligned}
V(\mathrm{CD}_n) &:= \{1, \dots, n\}, \\
E(\mathrm{CD}_n) &:= \{(i, i+1) : i \in \{1, \dots, n-1\}\} \cup \{(n, 1)\}.
\end{aligned}
$$

Example 3.3.1 *The cycle graph* CG_4*, the double cycle* $\mathrm{dg}(\mathrm{CG}_4)$ *and the cycle digraph* CD_4 *are as follows:*

$$
\mathrm{CG}_4 = \begin{tikzcd} 1 \arrow[r, dash] \arrow[d, dash] & 2 \arrow[d, dash] \\ 4 \arrow[r, dash] & 3 \end{tikzcd}
\quad
\mathrm{dg}(\mathrm{CG}_4) = \begin{tikzcd} 1 \arrow[r, bend left] \arrow[d, bend left] & 2 \arrow[l, bend left] \arrow[d, bend left] \\ 4 \arrow[r, bend left] \arrow[u, bend left] & 3 \arrow[l, bend left] \arrow[u, bend left] \end{tikzcd}
\quad
\mathrm{CD}_4 = \begin{tikzcd} 1 \arrow[r] & 2 \arrow[d] \\ 4 \arrow[u] & 3 \arrow[l] \end{tikzcd}
$$

A **tree** is a finite connected graph with at least one vertex and no cycle (i.e., acyclic). A **leaf** in a tree G is a vertex of degree 1 (i.e., there is exactly one edge $e \in E(G)$ with $v \in e$). The directed version $\mathrm{dg}(G)$ of trees G are called "directed trees" in [59], and we use **ditree** here. In Chapter 5 we study ditrees and their properties in order to characterise an important class of MUs.

A **path graph** is a tree with $n \in \mathbb{N}$ vertices, where at most two vertices have degree 1 (i.e., there are at most two leaves) and the other vertices are of degree 2. The standardised path graph is $\mathbf{PG}_{\boldsymbol{n}} := (\{1, \dots, n\}, \{\{1,2\}, \dots, \{n-1, n\}\})$. A **dipath** is a digraph isomorphic to $\mathrm{dg}(\mathrm{PG}_n)$. A **path digraph** is some digraph isomorphic to the standardised path digraph

$$\mathbf{PD}_{\boldsymbol{n}} := (\{1, \dots, n\}, \{(1,2), \dots, (n-1, n)\}).$$

Example 3.3.2 *The path graph* PG_4 *and the path digraph* PD_4 *are:*

$$
\mathrm{PG}_4 = \begin{tikzcd} 1 \arrow[r, dash] & 2 \arrow[r, dash] & 3 \arrow[r, dash] & 4 \end{tikzcd}
\quad
\mathrm{PD}_4 = \begin{tikzcd} 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \end{tikzcd}
$$

and the dipath $\mathrm{dg}(\mathrm{PG}_4)$ *is as follows:*

$$
\mathrm{dg}(\mathrm{PG}_4) = \begin{tikzcd} 1 \arrow[r, bend left] & 2 \arrow[l, bend left] \arrow[r, bend left] & 3 \arrow[l, bend left] \arrow[r, bend left] & 4 \arrow[l, bend left] \end{tikzcd}
$$

Definition 3.3.3 *A* ***rooted tree*** T *is a pair* $T = (T_0, r)$*, where* T_0 *is a finite tree and* $r \in V(T_0)$*, the* ***root****.*

A directed rooted tree is a finite acyclic digraph with exactly one source (a vertex of in-degree zero), where the in-degree of every vertex other than the source is precisely one. Various notions for rooted trees:

Definition 3.3.4 *Consider a rooted tree T with root r.*

1. *Any vertex y on the unique path from r to a vertex x is called an **ancestor** of x in case $y \neq x$. So if $r \neq x$, then r is the first ancestor of x.*

2. *If y is the first ancestor of x (that is there is no other vertex between x,y), then y is the **parent** of x, and x is a **child** of y.*

3. *A vertex with no children is a **leaf**, and a non-leaf vertex is a **inner (internal) vertex**.*

4. *A **near-leaf** is a vertex with at least one child being a leave.*

5. *T is called **trivial** if it has no inner vertex (such rooted trees are the only cases where the root is a leaf).*

6. *The **height** of x is the length of the longest path from x to a leaf, and the height of T is the height of its root.*

A **binary tree** is a rooted tree in which each vertex has at most two children. In the latter case, the first child is also called **left child**, the second child **right child**. For a binary tree T, the number of leaves is denoted by **#lvs(T)**. **Full binary trees** are rooted trees, where every vertex has either zero or two children.

Chapter 4

Basics of minimal unsatisfiability

In this chapter we study minimal unsatisfiability, and the main focus is to present basic methods and results used for investigating the structure of MUs and their isomorphism types. Furthermore characterisations of the most basic class of MUs (i.e., $\mathcal{MU}_{\delta=1}$) and its two important subclasses with polytime isomorphism decision are discussed.

The outline of this section is as follows. In Sections 4.1 and 4.2 saturation and its dual notion, marginalisation, are discussed. Singularity and singular MUs are defined in Section 4.3, and an upper bound for minimum variable degree of MUs is given in this section. Sections 4.4 and 4.5 review the main tools for investigating the structure of MUs, namely splitting and singular DP-reduction. Section 4.6 is devoted to the class $\mathcal{MU}_{\delta=1}$ which covers the class of minimally unsatisfiable renamable Horn clause-sets and the class of unsatisfiable hitting clause-sets with deficiency 1. Finally, Subsections 4.6.2 and 4.6.4 review characterisations of marginal and saturated elements of $\mathcal{MU}_{\delta=1}$ via connecting them to graph theory, from which the polytime isomorphism decision of these subclasses follows.

4.1 Saturation

In this section we discuss the process of "saturation" for MUs as introduced in [53]. First we define a special notation from [107] and [108] which is to add a literal to a clause C in a clause-set, under the restrictions that we obtain a new clause $C' \neq C$, and do not introduce a new variable (recall that "$\dot\cup$" denotes disjoint union, and $\mathrm{lit}(C)$ is the set of all possible literals over $\mathrm{var}(C)$):

Definition 4.1.1 *For $F \in \mathcal{CLS}$, $C \in F$ and $x \in \mathrm{lit}(F) \setminus \mathrm{lit}(C)$, such that $C \cup \{x\} \notin F$, we define*

$$\mathbf{S}(\boldsymbol{F}, \boldsymbol{C}, \boldsymbol{x}) := (F \setminus \{C\}) \mathbin{\dot\cup} (C \mathbin{\dot\cup} \{x\}) \in \mathcal{CLS}.$$

For $F \in \mathcal{MU}$ a ***one-step saturation (on** C**)*** *is the transition $F \leadsto \mathrm{S}(F, C, x)$ such that $\mathrm{S}(F, C, x) \in \mathcal{MU}$ (note that the usage of the term "$\mathrm{S}(F, C, x)$" implies that it is defined, i.e., all assumptions are fulfilled).*

Remarks:

1. Compared to [108, Definition 1] and [109, Definition 3.5], here the case $C \cup \{x\} \in F$ is disallowed, since it does not seem to be useful.
2. By definition for $F' := \mathrm{S}(F, C, x)$ we have:
   (a) $\mathrm{var}(F') = \mathrm{var}(F)$, and so $n(F') = n(F)$.
   (b) $c(F') = c(F)$, and therefore $\delta(F') = \delta(F)$.
   (c) $\ell(F') = \ell(F) + 1$.
   (d) $F \models F'$; and if F is irredundant, then so is F'.
3. By definition, for $F' := \mathrm{S}(F, C, x)$ there is a bijection $f : F \to F'$ with $C \subseteq f(C) \in F'$ for all $C \in F$.

Lemma 4.1.2 ([109]) *Consider $F, F' \in \mathcal{CLS}$ with $F' := \mathrm{S}(F, C, x)$.*

1. *If $F' \in \mathcal{USAT}$, then $F \in \mathcal{USAT}$.*
2. *If $F \in \mathcal{MU}$ and $F' \in \mathcal{USAT}$, then $F' \in \mathcal{MU}$.*

Proof: For Part 1 note that from $\varphi * F = \top$ follows $\varphi * F' = \top$. For Part 2, by definition if F' is redundant then F would also be redundant, contradicting minimally of F. $\square$
For more details on one-step saturation see [109, Lemma 3.9, Lemma 6.5].

Example 4.1.3 *Consider $F := \{\{a, b\}, \{\overline{a}\}, \{\overline{b}\}\} \in \mathcal{MU}_{\delta=1}$. Using Definition 4.1.1 we can only perform one-step saturation on $\{\overline{a}\} \in F$ or $\{\overline{b}\} \in F$ as follows (to obtain an MU):*

$$F' := \mathrm{S}(F, \{\overline{a}\}, b) = \{\{a, b\}, \{\overline{a}, b\}, \{\overline{b}\}\} \in \mathcal{MU}_{\delta=1},$$

$$F'' := \mathrm{S}(F, \{\overline{b}\}, a) = \{\{a, b\}, \{\overline{a}\}, \{a, \overline{b}\}\} \in \mathcal{MU}_{\delta=1}.$$

An MU F is called saturated if adding any literal occurrences to a clause in F yields a satisfiable clause-set.

Definition 4.1.4 ([108]) *A clause-set $F \in \mathcal{MU}$ is called* ***saturated*** *if no one-step saturation is possible, and the set of all saturated minimally unsatisfiable clause-sets is denoted by $\boldsymbol{\mathcal{SMU}} := \{F \in \mathcal{MU} \mid \neg\exists\, C, x : \mathrm{S}(F, C, x) \in \mathcal{MU}\}$. We also use $\boldsymbol{\mathcal{SMU}_{\delta=k}} := \mathcal{SMU} \cap \mathcal{MU}_{\delta=k}$ to denote the set of all saturated (minimally unsatisfiable) clause-sets with fixed deficiency $k \geq 1$.*

By Lemma 4.1.2 every $F \in \mathcal{MU}$ can be saturated (also noted in [53], [98]), that is, either F is saturated or there exist some $F' \in \mathcal{MU}$ such that F' is a saturation of F. Note that in general F might have many saturations. In Section 4.6.2 we will discuss characterisation of the class $\mathcal{SMU}_{\delta=1}$, while the nonsingular elements of $\mathcal{SMU}_{\delta=2}$ are characterised in Section 5.3.

Considering the decision complexity, in [85, Theorem 1] it is shown that the decision problem whether $F \in \mathcal{CLS}$ is saturated minimally unsatisfiable is D^P-complete, where the complexity class D^P is the set of problems which can be defined as the difference of two NP-problems. For the case of $\mathcal{SMU}_{\delta=1}$, a nice characterisation has been provided in [4] (called "strongly minimal unsatisfiable" there), yielding a polytime decision for this class. In general classes $\mathcal{SMU}_{\delta=k}$ for fixed deficiency $k \geq 1$ are polytime decidable as classes $\mathcal{MU}_{\delta=k}$ have been shown to be polytime decidable ([91], [52]).

Example 4.1.5 *MUs F', F'' in Example 4.1.3 are saturated. By definition, $A_0 = \{\bot\} \in \mathcal{MU}_{\delta=1}$ is saturated; and as A_1 is saturated, for any $F \in \mathcal{SMU}$ with $n(F) = 1$ holds $F \cong A_1 \in \mathcal{SMU}_{\delta=1}$ (see Example 2.4.2). In fact all full clause-sets A_n are saturated as all clauses in A_n are full (see Section 2.1).*

Definition 4.1.6 ([108]) *For $F \in \mathcal{MU}$, a **partial saturation** is some $G \in \mathcal{MU}$ which can be obtained from F by a series (possibly zero) of one-step saturations according to Definition 4.1.1, while we have a **saturation** if $G \in \mathcal{SMU}$.*

Remarks:

1. For every saturation G of an MU F we have $\mathrm{var}(G) = \mathrm{var}(F)$ and $\delta(G) = \delta(F)$.
2. Partial saturations of $F \in \mathcal{MU}$ are obtained by repeated applications of the transition $F \leadsto \mathrm{S}(F, C, x)$, such that we always stay within $\mathcal{MU}$. We have a (complete) saturation if and only if the sequence is maximal (can not be extended).

For two clause-sets $F, G \in \mathcal{MU}_{\delta=k}$, $k \geq 1$, Definitions 4.1.1 and 4.1.6 imply that G is a partial saturation of F iff $\mathrm{var}(G) = \mathrm{var}(F)$ and there is a bijection $f : F \to G$ such that for all clauses $C \in F$ we have $C \subseteq f(C)$. Furthermore, G is a saturation of F iff $G \in \mathcal{SMU}$ and G is a partial saturation of F.

We can also localise saturation:

Definition 4.1.7 *For $F \in \mathcal{MU}$ and a variable $v \in \mathrm{var}(F)$, we define **local saturation** as the process of adding literals $v, \overline{v}$ to some clauses in F (not already containing $v, \overline{v}$), until adding any additional v or $\overline{v}$ yields a satisfiable clause-set. Then the result is **locally saturated on** v.*

See Lemma 4.4.4 for an application of local saturation. We will use this method in Section 5.4 where we characterise the nonsingular elements of $\mathcal{MU}_{\delta=2}$.

Example 4.1.8 *Consider the following 2-MU*

$$F := \{\{-1,2\},\{1,-2\},\{-2,3\},\{2,-3\},\{1,3\},\{-1,-3\}\} \in \mathcal{MU}_{\delta=3}.$$

The result of locally saturating F on variable 3 is as follows:

$$F' := \{\{-1,2,3\},\{1,-2,-3\},\{-2,3\},\{2,-3\},\{1,3\},\{-1,-3\}\} \in \mathcal{MU}_{\delta=3}.$$

Finally we consider an important subclass of MUs, namely unsatisfiable **hitting clause-sets**. In a hitting clause-set $F \in \mathcal{CLS}$ all clauses $C, D \in F$, $C \neq D$, have a clash, i.e., $C \cap \overline{D} \neq \emptyset$ holds. By definition it follows that $F \in \mathcal{CLS}$ is hitting iff every $C \in F$, as partial assignment, is a satisfying assignment for $F \setminus \{C\}$ (see Section 2.2). Therefore unsatisfiable hitting clause-sets are in a sense the most extreme case of irredundant clause-sets as clauses have no common falsifying assignment ([102]). We use $\boldsymbol{\mathcal{UHIT}} \subset \mathcal{USAT}$ to denote the set of unsatisfiable hitting clause-sets, while $\boldsymbol{\mathcal{UHIT}_{\delta=k}}$ is the set all $F \in \mathcal{UHIT}$ with deficiency k. Furthermore $\boldsymbol{\mathcal{UHIT}'_{\delta=k}} := \mathcal{UHIT}_{\delta=k} \cap \mathcal{MU}'$. Example of unsatisfiable hitting clause-sets are the full clause-sets $A_n \in \mathcal{UHIT}$. By definition, $F \in \mathcal{UHIT}$ implies that $F \in \mathcal{MU}$ ([94]). Furthermore as for $F \in \mathcal{UHIT}$ and any partial assignment $\varphi \in \mathcal{PASS}$ we have $\varphi * F \in \mathcal{UHIT} \subset \mathcal{MU}$ ([94]), the elements of this class are actually saturated (see Lemma 4.4.5 Part 3):

Lemma 4.1.9 ([108]) $\mathcal{UHIT} \subset \mathcal{SMU}$.

Therefore we have $\mathcal{UHIT}_{\delta=k} \subseteq \mathcal{SMU}_{\delta=k} \subset \mathcal{MU}_{\delta=k}$, and so unsatisfiable hitting clause-sets have deficiency at least one. See Section 4.6.2 for the full characterisation of $\mathcal{UHIT}_{\delta=1}$, and Section 7.3 for the discussion on characterising classes $\mathcal{UHIT}_{\delta=k}$.

4.2 Marginalisation

In the previous section we discussed saturated MUs, that is, those $F \in \mathcal{MU}$ where adding any literal occurrence to any clause destroys minimal unsatisfiability. Here we study "marginal"MUs, that is, those $F \in \mathcal{MU}$ where removing any literal occurrence from any clause destroys the property of being minimally unsatisfiable. We use the following notation to remove a literal from a clause-set, under the restrictions that we obtain a new clause and do not eliminate a variable altogether:

Definition 4.2.1 *For $F \in \mathcal{CLS}$, $C \in F$ and $x \in C$, such that $C \setminus \{x\} \notin F$ and $\mathrm{var}(x) \in \mathrm{var}(F \setminus \{C\}$, we define*

$$\mathbf{M}(\boldsymbol{F, C, x}) := (F \setminus \{C\}) \cup (C \setminus \{x\}) \in \mathcal{CLS}.$$

*For $F \in \mathcal{MU}$ a **one-step marginalisation** is the transition $F \rightsquigarrow \mathrm{M}(F, C, x)$ such that $\mathrm{M}(F, C, x) \in \mathcal{MU}$.*

Remarks:

1. By definition we have for $F' := \mathrm{M}(F, C, x)$:

   (a) $\mathrm{var}(F') = \mathrm{var}(F)$, and so $n(F') = n(F)$.

   (b) $c(F') = c(F)$, and so $\delta(F') = \delta(F)$.

   (c) $\ell(F') = \ell(F) - 1$.

   (d) $F' \models F$, so if $F \in \mathcal{USAT}$, then also $F' \in \mathcal{USAT}$.

   (e) And if F' is irredundant, then so is F.

   (f) So if $F \in \mathcal{MU}$, then $F' \in \mathcal{MU}$ iff F' is irredundant.

2. Consider $F \in \mathcal{MU}$. The question about $\mathrm{S}(F, C, x)$ is whether (still) $\mathrm{S}(F, C, x) \in \mathcal{USAT}$ (then we get $\mathrm{S}(F, C, x) \in \mathcal{MU}$), while for $\mathrm{M}(F, C, x)$ we always have $\mathrm{M}(F, C, x) \in \mathcal{USAT}$, and the question is whether $\mathrm{M}(F, C, x)$ is (still) irredundant (then we get $\mathrm{M}(F, C, x) \in \mathcal{MU}$).

Definition 4.2.2 *A clause-set $F \in \mathcal{MU}$ is called* ***marginal*** *if no one-step marginalisation is possible. The set of all marginal minimally unsatisfiable clause-sets is denoted by* $\boldsymbol{\mathcal{MMU}} := \{F \in \mathcal{MU} \mid \neg\exists\, C, x : \mathrm{M}(F, C, x) \in \mathcal{MU}\}$, *and the set of all marginal (minimally unsatisfiable) clause-sets with fixed deficiency $k \geq 1$ is denoted by* $\boldsymbol{\mathcal{MMU}_{\delta=k}} := \mathcal{MMU} \cap \mathcal{MU}_{\delta=k}$.

As characterised in [83, Theorem 8] MUs that are both saturated and marginal are the full clause-sets $F \cong A_{n(F)}$:

Lemma 4.2.3 ([83]) *For $F \in \mathcal{MU}$ holds $F \in \mathcal{SMU} \cap \mathcal{MMU}$ iff $F \cong A_{n(F)}$.*

From Lemma 4.2.3 follows that if for a clause-set $F \in \mathcal{SMU}$ we have $F \ncong A_{n(F)}$ then F is not marginal and there is at least one literal occurrence in F which can be removed without destroying minimal unsatisfiability. Therefore there is some MU $F' \neq F$ where F is a saturation of F' ([109, Corollary 3.11]).

Definition 4.2.4 *For $F \in \mathcal{MU}$ a* ***partial marginalisation*** *is some $G \in \mathcal{MU}$ which can be obtained from F by a series (possibly zero) of one-step marginalisations according to Definition 4.2.1, while we have a* ***marginalisation*** *if $G \in \mathcal{MMU}$.*

Remarks:

1. Partial marginalisations of $F \in \mathcal{MU}$ are obtained by repeated applications of the transition $F \rightsquigarrow \mathrm{M}(F, C, x)$, such that we always stay within $\mathcal{MU}$. We have a (complete) marginalisation if and only if the sequence is maximal (can not be extended).

2. For $F, G \in \mathcal{MU}$ the following properties are equivalent:

   (a) F is a partial marginalisation of G.

   (b) G is a partial saturation of F.

3. An overview of marginal MUs and partial marginalisation can be found in [85] and [109].

By Definitions 4.2.1 and 4.2.4, a marginalisation G for $F \in \mathcal{MU}$ is some marginal $G \in \mathcal{MU}$ with $\mathrm{var}(G) = \mathrm{var}(F)$ such that there is a bijection $\alpha : F \to G$ with $\forall C \in F : C \supseteq \alpha(C)$. Similar to saturation, every MU can be marginalised, and so an MU F is either marginal or there is some $G \in \mathcal{MU}$ which is a marginalisation of F.

4.3 Singularity

A singular variable for a clause-set F is a variable occurring in one sign only once.

Definition 4.3.1 *A variable $v \in \mathcal{VA}$ is a* ***singular variable*** *for $F \in \mathcal{CLS}$ if*

$$\min(\mathrm{ld}_F(v), \mathrm{ld}_F(\overline{v})) = 1,$$

while otherwise is ***nonsingular****. If a clause-set F does not have any singular variable, it is called* ***nonsingular****. An* ***m-singular variable*** *is a singular variable with $\mathrm{vd}_F(v) = m + 1$ ($m \in \mathbb{N}$).*

We use the following notations from [108]. For $F \in \mathcal{CLS}$, $\mathbf{var_s(\boldsymbol{F})} \subseteq \mathrm{var}(F)$ is the set of singular variables. The set of nonsingular MUs is denoted by $\boldsymbol{\mathcal{MU}'} \subset \mathcal{MU}$, while $\boldsymbol{\mathcal{MU}'_{\delta=k}} := \mathcal{MU}' \cap \mathcal{MU}_{\delta=k}$ is the set of nonsingular MUs with fixed deficiency $k \in \mathbb{N}$. By $\boldsymbol{\mathcal{SMU}'} \subset \mathcal{SMU}$ we denote the set of nonsingular saturated MUs, while $\boldsymbol{\mathcal{SMU}'_{\delta=k}} := \mathcal{SMU}' \cap \mathcal{SMU}_{\delta=k}$. Furthermore we use $\mathbf{2}$–$\boldsymbol{\mathcal{MU}'} \subset 2$–$\mathcal{MU}$ for the set of nonsingular 2-MUs.

Example 4.3.2 *For $F := \{\{x\}, \{\overline{y}\}, \{\overline{x}, y, z\}, \{y, z\}, \{\overline{y}, \overline{z}\}\} \in \mathcal{CLS}$, variables x, z are singular variables for F, while y is a nonsingular variable. So $\mathrm{var_s}(F) = \{x, z\}$. Also x is a 1-singular variable, while z is a 2-singular variable for F.*

We recall that MUs have no pure literal, and that any variable occurring in a unit-clause in an MU is a singular variable, since otherwise the unit-clause would subsume another clause ([78], [108, Lemma 14]).

Lemma 4.3.3 *If an MU F has a unit-clause $\{x\} \in F$, then $\mathrm{var}(x) \in \mathrm{var_s}(F)$.*

Proof: If there would be $C \in F \setminus \{\{x\}\}$ with $x \in C$, then $\{x\} \subset C$ contradicting minimal unsatisfiability of F. So $\mathrm{ld}_F(x) = 1$. $\square$

As shown in [91, Lemma C.2], $F \in \mathcal{MU}$ with $n(F) > 0$ has a variable $v \in \mathrm{var}(F)$ with at most $\delta(F)$ positive and at most $\delta(F)$ negative occurrences. The special case of $\delta(F) = 1$ had been proved in [45, Theorem 12] (i.e., for $F \in \mathcal{MU}_{\delta=1}$ there exist a 1-singular variable).

Lemma 4.3.4 ([91]) *For $F \in \mathcal{MU}_{\delta=k}$, $k \in \mathbb{N}$ and $F \neq \{\bot\}$, there exists a variable $v \in \operatorname{var}(F)$ with $\operatorname{ld}_F(v), \operatorname{ld}_F(\overline{v}) \leq k$.*

Corollary 4.3.5 ([45]) *Every clause-set $F \in \mathcal{MU}_{\delta=1} \setminus \{\{\bot\}\}$ has a 1-singular variable (i.e., $\mu\mathrm{vd}(F) = 2$).*

By Lemma 4.3.4 for $F \in \mathcal{MU}_{\delta=k} \setminus \{\bot\}$ and $k \in \mathbb{N}$, it is obvious that $\mu\mathrm{vd}(F) \leq 2k$. However a sharper upper bound is given for $\mu\mathrm{vd}(F)$ in [109, Theorem 8.6]. For deficiency 2, Lemma 4.3.4 implies that $F \in \mathcal{MU}'_{\delta=2}$ has a variable occurring precisely twice positively and twice negatively, and so $\mu\mathrm{vd}(F) \leq 4$. We will use this fact in Section 5.3 to characterise the isomorphism types of $F \in \mathcal{MU}'_{\delta=2}$. In Section 4.5 we study how to eliminate singular variables without destroying minimal unsatisfiability, and when this process is confluent.

4.4 Splitting

A fundamental tool for understanding the structure of MUs is "splitting", which is to obtain two new clause-sets from a clause-set F by setting an appropriate variable in F to both truth values $0, 1$, and then analyse them and lift the information obtained back to F. An early use is in [42] where the "splitting rule" was used instead of the "rule for eliminating atomic formulas" to improve the implementation of the algorithm in [43].

Definition 4.4.1 ***Splitting*** *of a clause-set F is the process of assigning the truth values to a variable $v \in \operatorname{var}(F)$, that is, obtaining two clause-sets $F_0 := \langle v \to 0 \rangle * F$ and $F_1 := \langle v \to 1 \rangle * F$.*

Splitting of a clause-set $F \in \mathcal{CLS}$ is **disjoint** (also called "disjunctive splitting" in [27, Definition 8]) if there is no clause $C \in F$ that belongs to both F_0, F_1.

Example 4.4.2 *Consider $F, F' \in \mathcal{MU}_{\delta=3}$ in Example 4.1.8. Splitting $F = \{\{-1,2\},\{1,-2\},\{-2,3\},\{2,-3\},\{1,3\},\{-1,-3\}\}$ on variable 3 yields clause-sets F_0, F_1 as follows:*

$$F_0 := \langle 3 \to 0 \rangle * F = \{\{-1,2\},\{1,-2\},\{-2\},\{1\}\},$$

$$F_1 := \langle 3 \to 1 \rangle * F = \{\{-1,2\},\{1,-2\},\{2\},\{-1\}\}.$$

F_0, F_1 are both unsatisfiable. However they are not MU as $F_0 \setminus \{\{1,-2\}\}$ and $F_1 \setminus \{\{-1,2\}\}$ are still unsatisfiable. Now consider F', which is locally saturated on 3. Splitting F' on 3 is disjoint and yields two MUs F'_0, F'_1 as follows:

$$F'_0 = \{\{-1,2\},\{-2\},\{1\}\} \in \mathcal{MU}_{\delta=1}, \quad F'_1 = \{\{1,-2\},\{2\},\{-1\}\} \in \mathcal{MU}_{\delta=1}.$$

Clause-sets are closed under splitting (recall Definition 2.2.2), and it is clear that a clause-set F is unsatisfiable iff the results of splitting on a variable $v \in \operatorname{var}(F)$ are unsatisfiable ([42]). For an MU F splitting on any variable yields two unsatisfiable clause-sets which are in $\mathcal{MU}$ or have some sub-clause-sets in $\mathcal{MU}$. So in order to guarantee that minimal unsatisfiability is maintained by splitting a further condition is required. The approach of Kleine Büning and Zhao, as outlined in [83, Section 3], is to remove clauses appropriately in F_0, F_1, and study various conditions. In [53] it is shown that for $F \in \mathcal{SMU}$ and any literal $x \in \mathrm{lit}(F)$ we have $\langle x \to 1\rangle * F \in \mathcal{MU}$. Then the reverse direction was proved in [102, Corollary 5.3], obtaining an important characterisation of saturated MUs, namely that a clause-set F is saturated iff splitting on any variable in F yields two MUs. This characterisation has been generalised in [109, Lemma 3.15] as follows:

Lemma 4.4.3 ([109]) *Consider a subsumption-free clause-set $F \in \mathcal{CLS}$ (that is, F does not contain any clauses $C, D \in F$ with $C \subset D$ and $|C| + 1 = |D|$).*

1. *If there is a variable $v \in \mathrm{var}(F)$ with $\langle v \to 0\rangle * F, \langle v \to 1\rangle * F \in \mathcal{MU}$, then $F \in \mathcal{MU}$.*
2. *If there is a variable $v \in \mathrm{var}(F)$ with $\langle v \to 0\rangle * F, \langle v \to 1\rangle * F \in \mathcal{SMU}$, then $F \in \mathcal{SMU}$.*
3. *$F \in \mathcal{SMU}$ iff $F \neq \top$ and $\forall\, v \in \mathrm{var}(F)\ \forall\, \varepsilon \in \{0,1\} : \langle v \to \varepsilon\rangle * F \in \mathcal{MU}$.*

Now the proof of Lemma 4.4.3 (in [109]) yields in fact, that even for a locally saturated $F \in \mathcal{MU}$ on a variable v (Definition 4.1.7), splitting on v maintains minimal unsatisfiability:

Lemma 4.4.4 *Consider $F \in \mathcal{MU}$ and a variable $v \in \mathrm{var}(F)$. If for each $C \in F$ a one-step saturation with v or $\overline{v}$ is not possible (that is, F is locally saturated on v), then we have $\langle v \to \varepsilon\rangle * F \in \mathcal{MU}$ for both $\varepsilon \in \{0,1\}$.*

We will use Lemmas 4.4.3 and 4.4.4 to characterise the nonsingular elements of $\mathcal{MU}_{\delta=2}$ and 2–$\mathcal{MU}$ in Chapter 6. More details about splitting and its applications could be found in [77], [83], [85], [106] and [109].

After establishing the criteria to maintain minimal unsatisfiability, the next step is to predict how exactly the deficiency changes via splitting. In general the deficiency may increase or decrease by splitting, as after removing the satisfied clauses one or more variables could be eliminated. By [92, Corollary 7.10] for $F \in \mathcal{MU}_{\delta=k}$ and any variable $v \in \mathrm{var}(F)$, we obtain $F_0 := \langle v \to 0\rangle * F$ and $F_1 := \langle v \to 1\rangle * F$ with $\delta(F_0), \delta(F_1) \leq k$. It is shown in [78, Theorem 2] that if all variables in F occur at least twice positively and negatively (i.e., F is nonsingular) then the deficiency is strictly decreased for both splitting results (i.e., $\delta(F_0), \delta(F_1) < k$). Also as observed in [91, Lemma 3.10] for $k \geq 2$ and any $F \in \mathcal{MU}_{\delta=k}$ there is a variable $v \in \mathrm{var}(F)$ such that splitting yields F_0, F_1 with $\delta(F_0), \delta(F_1) < k$. Finally it was shown in [109, Lemma 8.2] that in order to have control over the changes on the deficiency, splitting should be performed on a variable v with minimal degree since no variable can have all its occurrences only in clauses containing v resp. $\overline{v}$ and no further variable can be lost in F_0, F_1.

Lemma 4.4.5 ([109]) *For $F \in \mathcal{CLS}$ and a non-pure variable $v \in \mathrm{var}_{\mu\mathrm{vd}}(F)$, let $m_0 := \mathrm{ld}_F(\overline{v})$ and $m_1 := \mathrm{ld}_F(v)$. Consider $\varepsilon \in \{0, 1\}$.*

1. *$\mathrm{var}(\langle v \to \varepsilon \rangle * F) = \mathrm{var}(F) \setminus \{v\}$ (thus $n(\langle v \to \varepsilon \rangle * F) = n(F) - 1$).*
2. *If F is subsumption-free, then $\delta(\langle v \to \varepsilon \rangle * F) = \delta(F) - m_\varepsilon + 1$, where $\delta(F) - m_\varepsilon + 1 < \delta(F)$ if and only if $m_\varepsilon \geq 2$.*
3. *If $F \in \mathcal{SMU}_{\delta=k}$ for $k \in \mathbb{N}$, then $\langle v \to \varepsilon \rangle * F \in \mathcal{MU}_{\delta=k-m_\varepsilon+1}$, where $m_\varepsilon \leq k$.*

Example 4.4.6 *Consider Lemma 4.4.5, Part 2. We show that the condition $v \in \mathrm{var}_{\mu\mathrm{vd}}(F)$ is not enough to guarantee $\delta(\langle v \to \varepsilon \rangle * F) = \delta(F) - m_\varepsilon + 1$ iff $m_\varepsilon \geq 2$. Consider $F := \{\{1, 2\}, \{1, 2, 3\}, \{-1, 2, -3\}, \{1, -2, -3\}, \{-1, -2, 3\}\} \in \mathcal{CLS} \setminus \mathcal{MU}$, where $\delta(F) = 2$ and $\mathrm{var}_{\mu\mathrm{vd}}(F) = \{3\}$ with $\mathrm{ld}_F(3) = \mathrm{ld}_F(-3) = 2$. We have $F_0 := \langle 3 \to 0 \rangle * F = \{\{1, 2\}, \{-1, -2\}\}$ with $\delta(F_0) = 0 < \delta(F) - \mathrm{ld}_F(3) + 1 = 1$.*

However for $F \in \mathcal{MU}$ the condition $v \in \mathrm{var}_{\mu\mathrm{vd}}(F)$ guarantees the assertion since there are no clauses $C, D \in F$ with $C \subset D$.

MUs are subsumption-free and so by Lemma 4.4.5, Part 2 we obtain:

Corollary 4.4.7 ([78]) *For $F \in \mathcal{MU}'_{\delta=k}$, $k \geq 2$ and $F_0 := \langle v \to 0 \rangle * F$, $F_1 := \langle v \to 1 \rangle * F$ we have $\delta(F_0), \delta(F_1) < k$.*

4.5 Singular DP-reduction/extension

Another major tool for the analysis of MUs is "singular DP-reduction", that is, to reduce $F \in \mathcal{MU}$ to $\mathrm{DP}_v(F)$ for some singular variable $v \in \mathrm{var}(F)$ (see Definitions 2.6.1 and 4.3.1). The study of special cases of DP-reduction and singular DP-reduction started in [89], [104], [105], while some early papers concerning the following application of DP-reduction for MUs are [91] and [142], under the name "Davis-Putnam resolution", and also [101] (see [108] for a full overview). In this thesis use the terminology "singular DP-reduction" which was first used in [97], and we follow the notations from [108].

Definition 4.5.1 ([108]) *For $F, F' \in \mathcal{CLS}$ the relation $\boldsymbol{F} \xrightarrow{sDP} \boldsymbol{F'}$ holds if there is $v \in \mathrm{var}_{\mathrm{s}}(F)$ with $F' = \mathrm{DP}_v(F)$ (**singular DP-reduction**).*

For $F \in \mathcal{MU}$, $\mathbf{sDP}(\boldsymbol{F}) := \{F' \in \mathcal{MU}' : F \xrightarrow{sDP}_* F'\} \in \mathbb{P}_{\mathrm{f}}(\mathcal{MU}')$ is the set of all clause-sets obtained by "complete" singular DP-reduction of F.

An m**-singular DP-reduction** is a singular DP-reduction for an m-singular variable, e.g., **1-singular DP-reduction** is a DP-reduction for a 1-singular variable.

As explained in Section 2.6, $\mathrm{DP}_v(F)$ is satisfiability-equivalent to F. But application of the DP-reduction to an MU F may or may not yield another

MU. A positive example for $n \in \mathbb{N}$ and $v \in \{1, \dots, n\}$ is $\mathrm{DP}_v(A_n) \cong A_{n-1}$. It is shown that if v is a singular variable then minimal unsatisfiability of $\mathrm{DP}_v(F)$ is guaranteed ([91, Appendix B], [101, Lemma 6.1]). Also by [108, Lemma 9] singular DP-reduction for an MU F does not yield tautological resolvents, and neither between the resolvents nor between resolvents and old clauses a contraction happens. Therefore for any $F' \in \mathrm{sDP}(F)$ we have $\delta(F') = \delta(F)$, and so the class of MUs with fixed deficiency $k \geq 1$ is stable under singular DP-reduction. Since $2\text{–}\mathcal{CLS}$ is stable under resolution, also the classes $2\text{–}\mathcal{MU}_{\delta=k}$ are stable under singular DP-reduction (we will characterise these classes in Chapter 6). Furthermore by [108, Lemma 12] singular DP-reduction preserves saturatedness of MUs (i.e., the class $\mathcal{SMU}$ is also stable).

Lemma 4.5.2 ([108]) *Consider $F, F' \in \mathcal{CLS}$ with $F \xrightarrow{sDP}_* F'$ (i.e., F' is obtained by complete singular DP-reduction of F).*

1. *$F \in \mathcal{MU}$ iff $\delta(F') = \delta(F)$ and $F' \in \mathcal{MU}$.*
2. *If $F \in \mathcal{SMU}$ then $F' \in \mathcal{SMU}$.*
3. *If $F \in \mathcal{UHIT}$ then $F' \in \mathcal{UHIT}$.*

Corollary 4.5.3 ([108]) *The classes $\mathcal{MU}_{\delta=k}$, $\mathcal{SMU}_{\delta=k}$ and $\mathcal{UHIT}_{\delta=k}$ for $k \geq 1$ are stable under singular DP-reduction.*

Another fundamental result concerning singular DP-reduction of an MU F, shown in [108], is that the elements of $\mathrm{sDP}(F)$ all have the same number of variables (while in general they are non-isomorphic). The proof idea is to show that all complete singular DP-reductions for F must have the same length, and this can be established by utilising the commutativity properties of 1-singular variables, so that induction on the number of singular variables removed by complete singular DP-reduction can be used.

Theorem 4.5.4 ([108]) *For $F \in \mathcal{MU}$ and any $F', F'' \in \mathrm{sDP}(F)$ holds $n(F') = n(F'')$.*

So for $F \in \mathcal{MU}$ we can define the **nonsingularity type** $\mathbf{nst}(\boldsymbol{F}) := n(F') \in \mathbb{N}_0$ via any $F' \in \mathrm{sDP}(F)$ (first introduced in [108]). We have $0 \leq \mathrm{nst}(F) \leq n(F)$, with $\mathrm{nst}(F) = 0$ iff $\delta(F) = 1$, and $\mathrm{nst}(F) = n(F)$ iff F is nonsingular. The nonsingularity type $\mathrm{nst}(F)$ provides basic information about the isomorphism type of MUs after (complete) singular DP-reduction, and suffices for deficiency 2 and 2-MUs (which will be shown in Chapter 5).

A (complete) singular DP-reduction for an MU F is called **confluent** if $|\mathrm{sDP}(F)| = 1$ (i.e., the result of removing all singular variables by DP-reduction is always unique and does not depend on the order of the variables removed); while we have **confluence modulo isomorphism** if all elements of $\mathrm{sDP}(F)$ are pairwise isomorphic (i.e., for all $F', F'' \in \mathrm{sDP}(F)$ we have $F' \cong F''$). In general elimination of singular variables by singular DP-reduction is not confluent for

MUs. Corollary 4.3.5 implies that complete singular DP-reduction on $\mathcal{MU}_{\delta=1}$ must yield $\{\perp\}$ (i.e., $\mathcal{MU}'_{\delta=1} = \{\perp\}$), and so $\mathcal{MU}_{\delta=1}$ is confluent. Also the confluence of saturated MUs under singular DP-reduction is shown in [108, Theorem 23], which implies the confluence of $\mathcal{UHIT}$ (recall Lemma 4.1.9). By [108] we have confluence modulo isomorphism for MUs with deficiency 2; while for higher deficiencies there are non-isomorphic elements in $\mathrm{sDP}(F)$.

Lemma 4.5.5 ([108]) *For $F \in \mathcal{MU}$ holds:*

1. *If $\delta(F) = 1$ then $\mathrm{sDP}(F) = \{\{\perp\}\}$.*
2. *If $\delta(F) = 2$ then the elements of $\mathrm{sDP}(F)$ are pairwise isomorphic.*
3. *If $\delta(F) \geq 3$ then $\mathrm{sDP}(F)$ has some non-isomorphic clause-sets.*
4. *If $F \in \mathcal{SMU}$ then $|\mathrm{sDP}(F)| = 1$.*

In Section 6.4.1 we will discuss a special case of singular DP-reduction where we always have confluence.

We now consider the reverse direction of the singular DP-reduction, i.e., "singular DP-extension". This process was first introduced in [108, Examples 15,19,54] (called "inverse singular DP-reduction" there). We use the following definition from [109]:

Definition 4.5.6 *For $F \in \mathcal{CLS}$, $m \in \mathbb{N}$ and $v \in \mathcal{VA} \setminus \mathrm{var}(F)$, a* ***singular m-extension*** *$G \in \mathcal{CLS}$ of F with v is obtained by the following steps:*

1. *Choose m different clauses $D_i \in F$ for $i \in \{1, \dots, m\}$.*
2. *Choose a subset $C \subseteq \bigcap_{i=1}^{m} D_i$.*
3. *Choose clauses D'_i for $i \in \{1, \dots, m\}$ such that $(D_i \setminus C) \subseteq D'_i \subseteq D_i$.*
4. *For $x \in \mathrm{lit}(\{v\})$ let $C' := C \cup \{x\}$ and $D''_i := D'_i \cup \{\overline{x}\}$ for $i \in \{1, \dots, m\}$.*
5. *$G := (F \setminus \{D_1, \dots, D_m\}) \cup \{C', D''_1, \dots, D''_m\}$.*

By definition, for a singular m-extension F' of $F \in \mathcal{CLS}$ ($m \in \mathbb{N}$) we have $n(F') = n(F) + 1$ and $c(F') = c(F) + 1$, and so $\delta(F') = \delta(F)$ (note that the D'_i are pairwise different). As shown in [109, Lemma 5.8] for $F, F' \in \mathcal{CLS}$, $m \in \mathbb{N}$ and $v \in \mathrm{var}(F)$, the relation $F \xrightarrow{\mathrm{sDP}} F'$ holds for a singular DP-reduction on an m-singular variable v with $c(F') = c(F) - 1$ iff F is obtained by a singular m-extension of F'. Therefore by Lemma 4.5.2, Part 1 minimal unsatisfiability is maintained by singular DP-extension (noted in [109, Lemma 5.9]).

Lemma 4.5.7 ([109]) *For $F \in \mathcal{CLS}$, $m \in \mathbb{N}$ and a singular m-extension F' of F holds: $F \in \mathcal{MU} \Leftrightarrow F' \in \mathcal{MU}$.*

4.6 MUs of deficiency one

This section is about understanding the most basic class of minimally unsatisfiable clause-sets, i.e., MUs with deficiency 1 (recall that for $F \in \mathcal{MU}$ holds $\delta(F) \geq 1$). First in Section 4.6.1 we discuss a generation process to produce all elements of this class. Then in Section 4.6.2 the saturated elements of $\mathcal{MU}_{\delta=1}$ are characterised via full binary trees, from which we obtain non-saturated elements of $\mathcal{MU}_{\delta=1}$ via partial marginalisation. Also Lemma 4.6.12 gives an easy criterion for partial marginalisation of any $F \in \mathcal{MU}_{\delta=1}$. In Section 4.6.3 we review minimally unsatisfiable renamable Horn clause-set. Finally in Section 4.6.4 the notion of conflict graph is defined and used to characterise the marginal elements of $\mathcal{MU}_{\delta=1}$.

4.6.1 Creation

The starting point of the investigation into the class $\mathcal{MU}_{\delta=1}$ is the basic fact that any $F \in \mathcal{MU}_{\delta=1}$ has a 1-singular variable (Corollary 4.3.5), i.e., a variable of degree 2. This fact was used in [45] to characterise $\mathcal{MU}_{\delta=1}$ as follows:

Lemma 4.6.1 ([45]) *For $F \in \mathcal{MU}$ the following properties are equivalent:*

1. $\delta(F) = 1$.
2. $\mathrm{sDP}(F) = \{\{\bot\}\}$.
3. *Repeated applications of 1-singular DP-reduction yield $\{\bot\}$.*

Lemma 4.6.1 implies that the decision whether "$F \in \mathcal{MU}_{\delta=1}$?" is polytime decidable (actually quadratic time, as shown in [45, Theorem 14]), while [4] had already shown polytime-decision of the class of saturated elements of $\mathcal{MU}_{\delta=1}$. Furthermore Lemma 4.6.1 yields the characterisation of the nonsingular MUs of deficiency 1, namely $\mathcal{MU}'_{\delta=1} = \mathcal{SMU}'_{\delta=1} = \{\{\bot\}\}$. Now using Part 3 of this lemma, we can produce all $F \in \mathcal{MU}_{\delta=1}$ by the reverse direction of 1-singular DP-reduction (i.e., singular 1-extension), starting from $\{\bot\}$, as follows:

Theorem 4.6.2 ([45]) *The following process creates exactly the elements of $\mathcal{MU}_{\delta=1}$:*

1. *Start with $\{\bot\} \in \mathcal{MU}_{\delta=1}$.*
2. *For every $F \in \mathcal{MU}_{\delta=1}$ already obtained, choose $C \in F$ and a new variable $v \notin \mathrm{var}(F)$, choose $C_1, C_2 \subseteq C$ with $C_1 \cup C_2 = C$, and replace C by $C_1 \cup \{v\}$, $C_2 \cup \{\overline{v}\}$.*

Proof: Given an element of $\mathcal{MU}_{\delta=1}$, a singular DP-extension preserves minimal unsatisfiability and deficiency (Lemma 4.5.7). So all clause-sets created by this process are in $\mathcal{MU}_{\delta=1}$.

Now we show that any $F \in \mathcal{MU}_{\delta=1}$ with $n := n(F) \geq 1$ can be produced via this process by induction on n. We know that $n = 1$ iff $F \cong A_1$, where A_1

is obtained from $\{\perp\}$ by this process. Assume $n \geq 2$. By Lemma 4.3.4 F has a singular variable $v \in \mathrm{var}(F)$ with $\mathrm{vd}_F(v) = 2$. Let $F' = \mathrm{DP}_v(F)$, where we have $F' \in \mathcal{MU}_{\delta=1}$ with $n(F') = n-1$. By induction hypothesis F' can be created by this process. Consider any $C \in F'$ and a new variable $v \notin \mathrm{var}(F')$ and choose $C_1, C_2 \subseteq C$ with $C_1 \cup C_2 = C$. Let $F := (F' \setminus C) \mathbin{\dot\cup} \{C_1 \cup \{v\}, C_2 \cup \{\overline{v}\}\}$. Clearly F can be constructed by this process. $\square$

As noted in [91, Lemma C.4] the saturated elements, $F \in \mathcal{SMU}_{\delta=1}$, are produced by the creation process in Theorem 4.6.2 if in each step the condition $C_1 = C_2 = C$ holds (i.e., the produced clauses are as large as possible). Now for $F \in \mathcal{SMU}_{\delta=1}$, $n(F) \geq 1$ created by this process, the variable $v \in \mathrm{var}(F)$ added in the first step is called a **full variable** for F as for all clauses $C \in F$ we have $v \in \mathrm{var}(C)$. Also to create the marginal elements, $\mathcal{MMU}_{\delta=1}$, the condition $C_1 \cap C_2 = \emptyset$ should be applied in each step. In Section 6.3 we will discuss another special case of this creation process which yields the elements of $2\text{–}\mathcal{MU}_{\delta=1}$.

4.6.2 The structure tree and saturated cases

In Section 4.6.1 we explained a creation process to produce the elements of $\mathcal{MU}_{\delta=1}$. In this section we first characterise saturated MUs of deficiency 1, i.e. $F \in \mathcal{SMU}_{\delta=1}$, and we show that the isomorphism problem for this class is polytime decidable. Furthermore we describe the class of unsatisfiable hitting clause-sets of deficiency 1 (recall Lemma 4.1.9). Then we show that all non-saturated elements of $\mathcal{MU}_{\delta=1}$ are obtained from the saturated cases via literal elimination in a way that no pure literal is created.

In [91, Lemma C.5] the structure of classes $\mathcal{SMU}_{\delta=1}$ and $\mathcal{MU}_{\delta=1}$ are described as a binary tree. These characterisations (which later were generalised in [102, Section 5.2]) just describe the expansion process in Theorem 4.6.2, and are basically the same as a resolution tree refuting $F \in \mathcal{MU}_{\delta=1}$ (the tree is not unique). Since the variables in the tree are all unique (the creation process in Theorem 4.6.2 does not reuse variables), any two clauses in F clash in at most one variable. Using the version of Theorem 4.6.2 for saturated cases we characterise the "structure tree" of $F \in \mathcal{SMU}_{\delta=1}$ as follows (recall that $F \in \mathcal{SMU}_{\delta=1} \setminus \{\{\perp\}\}$ has a unique full variable):

Lemma 4.6.3 *For $F \in \mathcal{SMU}_{\delta=1}$ let $\boldsymbol{T(F)}$ be the **structure tree**, a finite full binary tree where each inner vertex is labelled with a unique variable of F as follows:*

1. *If $F = \{\perp\}$, then $T(F)$ is the trivial tree. So assume $F \neq \{\perp\}$.*
2. *The root of $T(F)$ is labelled with the (unique) full variable v of F.*
3. *The left and right subtree of $T(F)$ is $T(\langle v \to 0\rangle * F)$ resp. $T(\langle v \to 1\rangle * F)$.*

On the other hand, for a full binary tree T, such that every inner vertex is labelled with a distinct variable, the clause-set $\boldsymbol{F(T)} \in \mathcal{SMU}_{\delta=1}$ is obtained by *associating with every leaf of T a clause, namely the clause collecting the literals along the path from the root to the leaf, with a left child meaning the positive literal of the variable at the vertex, and a right child the negative literal. We also write $F(T) = \{C_w : w$ leaf of $T(F)\}$.*

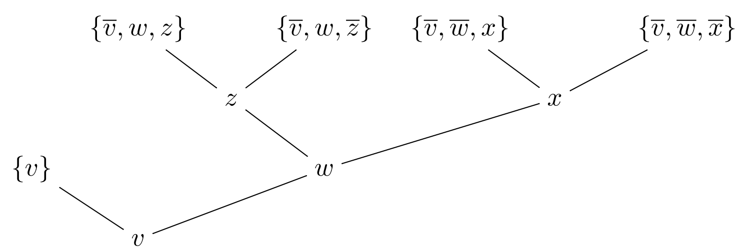
Remarks:

1. For a structure tree $T(F)$ of $F \in \mathcal{SMU}_{\delta=1}$ holds $\#\mathrm{lvs}(T(F)) = c(F)$.

2. For a structure tree $T(F)$ of $F \in \mathcal{SMU}_{\delta=1}$ we can extend the labelling of leaves to all vertices, that is, the label C_v of a vertex $v \in V(T(F)$ is the set of all literals in the path from v to the root of $T(F))$. Thus, the root is labelled by $\bot$. Now, for any inner vertex C_v with children C_1, C_2, holds $C_1 \diamond C_2 = C_v$. So, $T(F)$ can also be interpreted as a resolution refutation for F (where $T(F) : F \vdash \bot$ is a regular resolution tree).

3. The maps $F \mapsto T(F)$ and $T \mapsto F(T)$ are inverse bijections from $\mathcal{SMU}_{\delta=1}$ to the set of full binary trees, where inner vertices are labelled by distinct variables.

Example 4.6.4 *Consider $F := \{\{v\}, \{\overline{v}, w, z\}, \{\overline{v}, w, \overline{z}\}, \{\overline{v}, \overline{w}, x\}, \{\overline{v}, \overline{w}, \overline{x}\}\} \in \mathcal{SMU}_{\delta=1}$. The structure tree $T(F)$ is as follows, where v is the root of $T(F)$ (the full variable in F) and v, z, x are near-leaves of $T(F)$ (see Definition 3.3.4).*



By Lemma 4.6.3 we immediately obtain:

Corollary 4.6.5 *Consider $F \in \mathcal{SMU}_{\delta=1}$.*

1. *Any two clauses in F clash in exactly one variable;*

2. *For any clause $C \in F$ as a partial assignment (see Section 2.2) we have $C * (F \setminus \{C\}) = \top$;*

3. *The singular variables of F correspond one-to-one to the near-leaves of the structure tree $T(F)$ (Definition 3.3.4);*

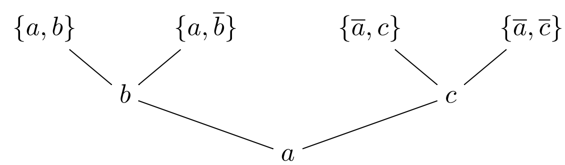
4. *1-singular variables of F correspond one-to-one to those near-leaves of $T(F)$ with both children being leaves.*

A **perfect** full binary tree is a binary tree in which all inner vertices have two children and all leaves have the same length (or same level). It is easy to see that the perfect structure trees correspond to those saturated MUs of deficiency 1 where all clauses have the same size.

Lemma 4.6.6 *Clause-sets $F \in \mathcal{SMU}_{\delta=1}$ are uniform iff they have a perfect structure tree $T(F)$.*

Proof: Consider a p-uniform $F \in \mathcal{SMU}_{\delta=1}$ (i.e., for all $C \in F$ holds $|C| = p$ and $p \in \mathbb{N}_0$). By Lemma 4.6.3 for a leaf in $T(F)$, labelled by C, $|C|$ is equal to the length of the path from that leaf to the root of $T(F)$. Thus, the length of all paths from leaves to the root is p (that is $T(F)$ is perfect). Also for a perfect structure tree $T(F)$ of $F \in \mathcal{SMU}_{\delta=1}$, all paths from leaves to the root of $T(F)$ have the same length. Thus, all clauses in F have the same length. $\square$

Example 4.6.7 *The only 1-uniform element of $\mathcal{SMU}_{\delta=1}$ and also $\mathcal{MU}_{\delta=1}$ is, up to isomorphism, A_1. Also any 2-uniform clause-set in $\mathcal{SMU}_{\delta=1}$ is isomorphic to $F := \{\{a, b\}, \{a, \overline{b}\}, \{\overline{a}, c\}, \{\overline{a}, \overline{c}\}\} \in \mathcal{SMU}_{\delta=1}$ where $T(F)$ is as follows:*



We will characterise the isomorphism types of 2-uniform MUs with deficiency 1 in Lemma 6.3.8.

We now come to the main characterisation of $\mathcal{SMU}_{\delta=1}$. Lemma 4.6.3 implies that the structure tree is a complete isomorphism invariant for the elements of $\mathcal{SMU}_{\delta=1}$, and so follows the polytime isomorphism decision of this class (note that trees have polytime isomorphism decision).

Lemma 4.6.8 *Consider two clause-sets $F_1, F_2 \in \mathcal{SMU}_{\delta=1}$. Then $F_1 \cong F_2$ if and only if $T(F_1)$ and $T(F_2)$ are isomorphic.*

Corollary 4.6.9 ([81]) *The class $\mathcal{SMU}_{\delta=1}$ has polytime isomorphism decision.*

The number of full binary trees with n vertices is asymptotically equal to $C\rho^{-n}n^{-3/2}$ where $\rho = 0.4026975$ and $C = 0.3188$ ([111]). Furthermore numerical data on the number of isomorphism types of full binary trees with m leaves (and so the number of isomorphism types of $F \in \mathcal{SMU}_{\delta=1}$ with $c(F) = m$) is given in the OEIS ([139, Sequence A001190]), called the Wedderburn-Etherington number, and starts as follows:

$$1, 1, 1, 2, 3, 6, 11, 23, 46, 98, 207, 451, \ldots$$

Illustration of the initial cases can be found in [138].

As already shown, the set of unsatisfiable hitting clause-sets fulfils $\mathcal{UHIT} \subset \mathcal{SMU}$ (Lemma 4.1.9). By characterising saturated elements of $\mathcal{MU}_{\delta=1}$ as structure trees it is easy to see that all $F \in \mathcal{SMU}_{\delta=1}$ are hitting clause-sets, i.e., $\mathcal{SMU}_{\delta=1} \subset \mathcal{UHIT}_{\delta=1}$. Therefore the elements of $\mathcal{UHIT}_{\delta=1}$ are precisely the saturated MUs of deficiency 1 and have a structure tree:

Corollary 4.6.10 ([91]) *$\mathcal{UHIT}_{\delta=1} = \mathcal{SMU}_{\delta=1}$, and $\mathcal{UHIT}'_{\delta=1} = \mathcal{SMU}'_{\delta=1} = \mathcal{MU}'_{\delta=1} = \{\{\bot\}\}$.*

Obviously every $F' \in \mathcal{MU}$ is obtained by partial marginalisation from some $F \in \mathcal{SMU}_{\delta=\delta(F')}$ (see Section 4.2). Any near-leaf of the structure tree $T(F)$ of $F \in \mathcal{SMU}_{\delta=1}$ is labelled with a singular variable in F (Corollary 4.6.5), i.e., every clause $C \in F$ has a literal x with $\mathrm{ld}_F(x) = 1$. Literal x can not be eliminated by partial marginalisation (otherwise there would be a pure literal), and so every clause in an MU with deficiency 1 has a literal of degree 1:

Corollary 4.6.11 ([91]) *Consider $F \in \mathcal{MU}_{\delta=1}$. For every clause $C \in F$ there exists a literal $x \in C$ with $\mathrm{ld}_F(x) = 1$.*

In general, partial marginalisations of $F \in \mathcal{MU}$ are hard to determine (see [109, Subsection 3.3]), however for $\mathcal{MU}_{\delta=1}$ we have a very easy criterion which is followed from Remarks of Definition 4.2.1 and Lemma 4.6.3, namely that we can remove any literal occurrence except where we create a pure variable:

Lemma 4.6.12 ([91]) *For $F \in \mathcal{MU}_{\delta=1}$, $C \in F$ and $x \in C$ we have $\mathrm{M}(F, C, x) \in \mathcal{MU}$ iff $\mathrm{ld}_F(x) \geq 2$.*

Therefore by Definition 4.2.1 and Lemma 4.6.12, $F \in \mathcal{MMU}_{\delta=1}$ is marginal iff all variables $v \in \mathrm{var}(F)$ are 1-singular.

Lemma 4.6.13 *The class of marginal MUs of deficiency 1, $\mathcal{MMU}_{\delta=1}$, is exactly the class of all $F \in \mathcal{MU}_{\delta=1}$ where all variables are 1-singular.*

So for $F \in \mathcal{MU}_{\delta=1}$ we obtain all marginalisation G of F by choosing for each $x \in \mathrm{lit}(F)$ one $C_x \in F$ with $x \in C_x$, and removing x from all other clauses of F. In Lemma 4.6.28 we show that in fact all elements of $\mathcal{MMU}_{\delta=1}$ are renamable Horn, while in Section 4.6.4 we fully characterise this class via trees.

Finally by Lemmas 4.6.3 and 4.6.12 we obtain:

Corollary 4.6.14 ([91]) *Consider $F \in \mathcal{MU}_{\delta=1}$. For all clauses $C, D \in F$ holds $|C \cap \overline{D}| \leq 1$, i.e., clauses C, D have at most one clash.*

4.6.3 Minimally unsatisfiable renamable Horn clause-sets

A clause-set $F \in \mathcal{CLS}$ is a **Horn clause-set** if any clause in F has at most one positive literal, i.e., $\forall C \in F : |C \cap \mathrm{var}(C)| \leq 1$. $\boldsymbol{\mathcal{HO}}$ is the set of Horn clause-sets, while $\boldsymbol{\mathcal{RHO}}$ is the set of **renamable Horn clause-sets** (this terminology was first used in [114]; also called "hidden" Horn clause-sets in the literature), i.e., clause-sets isomorphic to some $F \in \mathcal{HO}$. A well-known simple fact about unsatisfiable renamable Horn clause-sets is that they have a unit-clause:

Lemma 4.6.15 *For $F \in \mathcal{RHO} \cap \mathcal{USAT}$ there is $C \in F$ with $|C| \leq 1$.*

Proof: W.l.o.g. $F \in \mathcal{HO}$. Assume $\forall C \in F : |C| \geq 2$. Then the partial assignment setting all variables of F to 0 would be a satisfying assignment for F. □

In this section we study minimally unsatisfiable renamable Horn clause-set (short RHO-MUs). Horn clause-sets can be solved by unit-resolution in linear time ([47]), and the irredundancy of them can be decided in quadratic time ([84]). On the other hand as shown in [7] the problem whether a clause-set F is a renamable Horn clause-set is solvable in linear time. Therefore for a clause-set F the decision whether F is an RHO-MU can be solved in quadratic time. It is well-known (and we will show in Lemma 4.6.20) that RHO-MUs have deficiency 1, first noted in [45, Corollary 10]. In order to provide deeper understanding of the structure of RHO-MUs, we first discuss a complexity measure for resolution proofs, and then we use it to characterise clause-sets $F \in \mathcal{RHO} \cap \mathcal{MU}$.

In general there are various resolution complexity measures (the amount of effort needed to discover unsatisfiability) investigated in the literature for unsatisfiable clause-sets (see [20] and [19] for an overview). Here we consider the "tree-hardness" or just "hardness" $\mathrm{hd} : \mathcal{USAT} \to \mathbb{N}_0$, as a measure for understanding the complexity of resolution proofs. The notion of hardness was introduced in [90] for both satisfiable and unsatisfiable clause-sets, and in [49] only for unsatisfiable cases (called "space complexity of tree-like resolution"). The hardness was generalised to constraint satisfaction problems in [96]. In [6, Definition 8] we find a different extension of hardness to satisfiable clause-sets, while a more general form of hardness is investigated in [68] and [69]. Additional characterisations and the relation of tree-hardness to other resolution complexity measures are studied in [20] and [19].

The "Horton-Strahler number" of a tree T (also called "Strahler number" in [6], "levelled height" in [90] and "$d_c(T)$" in [49]) is a measure of its branching complexity, and is originally introduced by Horton [71] and Strahler [140] in the area of geology to study the morphology of rivers. The definition is re-invented in the literature and here we use the definition from [6].

Definition 4.6.16 *For a full binary tree T, the **Horton-Strahler** number, denoted by* $\mathbf{hs}(\boldsymbol{T}) \in \mathbb{N}_0$*, is defined as follows:*

1. *For a trivial T,* $\mathrm{hs}(T) := 0$.

2. *For a non-trivial T, there are two subtrees T_1, T_2 and we define*

$$\mathrm{hs}(T) := \begin{cases} \max(\mathrm{hs}(T_1), \mathrm{hs}(T_2)) & \mathrm{hs}(T_1) \neq \mathrm{hs}(T_2) \\ \mathrm{hs}(T_1) + 1 & \mathrm{hs}(T_1) = \mathrm{hs}(T_2) \end{cases}.$$

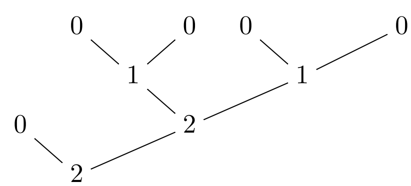
The hardness of an unsatisfiable clause-set is the minimum of the Horton-Strahler numbers of all the resolution trees driving the empty clause.

Definition 4.6.17 ([20]) *The* ***hardness*** *of $F \in \mathcal{USAT}$, denoted by* $\mathbf{hd}(\boldsymbol{F}) \in \mathbb{N}_0$*, is defined as the minimum of* $\mathrm{hs}(T)$ *over all resolution trees $T : F \vdash \bot$.*

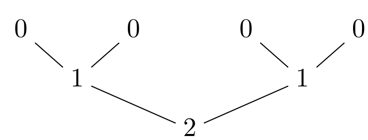
The Horton-Strahler number of the structure trees is useful for the general understanding of $\mathcal{MU}_{\delta=1}$. Also for the special case of $\mathcal{SMU}_{\delta=1}$ by the remarks of Lemma 4.6.3 we have (also shown in [67, Lemma 5.21]):

Lemma 4.6.18 *For $F \in \mathcal{SMU}_{\delta=1}$ and its structure tree $T(F)$ holds* $\mathrm{hd}(F) = \mathrm{hs}(T(F))$*.*

Example 4.6.19 *For A_0 we have* $\mathrm{hd}(A_0) = \mathrm{hs}(T(A_0)) = 0$*. Also $A_1 = \{\{1\}, \{-1\}\} \in \mathcal{SMU}_{\delta=1}$ has the hardness* $\mathrm{hd}(A_1) = \mathrm{hs}(T(A_1)) = 1$*. For F in Example 4.6.4 we obtain* $\mathrm{hd}(F) = \mathrm{hs}(T(F)) = 2$ *as follows:*



And for the uniform F in Example 4.6.7 we also have $\mathrm{hd}(F) = \mathrm{hs}(T(F)) = 2$*:*



An **input-resolution** of a clause-set F is a resolution tree $T : F \vdash C$ with $\mathrm{hs}(T) \leq 1$, that is, every vertex in T is either a leaf or has a leaf as a child. So by Definition 4.6.17 an unsatisfiable clause-set F has an input-resolution tree $T : F \vdash \bot$ iff $\mathrm{hd}(F) \leq 1$ ([90]). The following lemma states some characterisations of RHO-MUs from the literature, including that the elements of $\mathcal{RHO} \cap \mathcal{MU}$ are precisely MUs F with hardness $\mathrm{hd}(F) \leq 1$.

Lemma 4.6.20 *For $F \in \mathcal{MU}$ the following properties are equivalent:*

1. *$F \in \mathcal{RHO}$.*
2. $\mathrm{hd}(F) \leq 1$*, (i.e., F has an input-resolution refutation).*

3. $\delta(F) = 1$, *and there exists* $G \in \mathcal{SMU}_{\delta=1}$ *obtained by saturation of* F *with* $\mathrm{hs}(T(G)) \leq 1$.

Proof: By [33] it is known that refutation by unit-resolution is equivalent to refutation by input-resolution (i.e., hardness less-or-equal 1); this is also proved, in a more general context, in [96, Lemma 6.9]. While [70] showed that for $F \in \mathcal{MU}$ holds that $F \in \mathcal{RHO}$ if and only if F is refutable by unit-resolution. Thus the equivalence of Parts 1, 2 follows (also proved in [6, Lemma 4]).

[45, Corollary 10] noted first that for $F \in \mathcal{HO} \cap \mathcal{MU}$ we have $\delta(F) = 1$. Also for $F \in \mathcal{RHO}$ we consider a regularised input-resolution: the number of leaves is one more than the number of resolution variables, where due to regularisation the number of leaves is $c(F)$ and the number of resolution variables is $n(F)$. The hitting clause-set obtained from the tree is a saturation of F (again using regularity, which implies that no clause can be used twice as an axiom), with that tree yielding its structure tree (recall Corollary 4.6.10). Therefore Part 3 follows from Parts 1, 2.

It remains to show that Part 3 implies Parts 1 and 2. By Lemma 4.6.18 we have $\mathrm{hd}(G) = \mathrm{hs}(T(G)) \leq 1$. And as F is obtained by partial marginalisation of G, we get $\mathrm{hd}(F) \leq \mathrm{hd}(G)$. $\square$

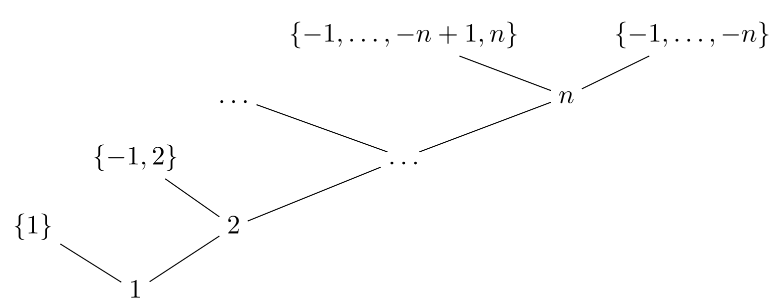
As already mentioned, the isomorphism problem for RHO-MUs is GI-complete (Section 1.4.2). However here we characterise the saturated RHO-MUs up to isomorphism, and in Section 4.6.4 we show that the marginal elements of $\mathcal{MU}_{\delta=1}$ and so the marginal RHO-MUs have polytime isomorphism decision.

An example of saturated MUs in $\mathcal{HO}$ from the literature is MUs S_n (occurred in [98], [102] and [67]), defined as follows:

Definition 4.6.21 *For* $n \in \mathbb{N}_0$ *we define*

$$\boldsymbol{\mathrm{S}_n} := \big\{ \{1\}, \{-1,2\}, \dots, \{-1,\dots,-(n-1),n\}, \{-1,\dots,-n\} \big\} \in \mathcal{SMU}_{\delta=1} \cap \mathcal{HO}.$$

Initial cases of S_n are $\mathrm{S}_0 = A_0$, $\mathrm{S}_1 = A_1$, $\mathrm{S}_2 = \{\{1\},\{-1,2\},\{-1,-2\}\}$ and $\mathrm{S}_3 = \{\{1\},\{-1,2\},\{-1,-2,3\},\{-1,-2,-3\}\}$. All variables in S_n are singular (i.e., $\mathrm{var}(\mathrm{S}_n) = \mathrm{var}_{\mathrm{s}}(\mathrm{S}_n)$), while the only 1-singular variable in S_n is n. Also S_n for $n \geq 1$ has precisely two full clauses. The structure tree $T(\mathrm{S}_n)$, shown below, has the Horton-Strahler number $\mathrm{hs}(T(\mathrm{S}_n)) = 1$ for $n \geq 1$, and so $\mathrm{hd}(\mathrm{S}_n) = 1$.



It is mentioned in [67, Example 6.10]) and we show that elements of $\mathcal{RHO} \cap \mathcal{SMU}$ are precisely MUs S_n, up to isomorphism.

Lemma 4.6.22 ([67]) *For $F \in \mathcal{RHO} \cap \mathcal{SMU}_{\delta=1}$ holds $F \cong \mathrm{S}_{n(F)}$.*

Proof: By Lemma 4.6.20 F has hardness $\mathrm{hd}(F) \leq 1$ and so the structure tree $T(F)$ has the Horton-Strahler number $\mathrm{hs}(T(F)) \leq 1$. Binary trees with $n+1$ leaves and the Horton-Strahler number less-or-equal one are isomorphic to the structure tree $T(\mathrm{S}_n)$, from which by Lemma 4.6.8 the assertion follows. □

This characterisation shows that clause-sets $F \in \mathcal{SMU}_{\delta=1}$ with a full clause have $\mathrm{hs}(T(F)) \leq 1$ and so are precisely MUs S_n (up to isomorphism). Therefore from Lemma 4.6.20 we get:

Lemma 4.6.23 *For $F \in \mathcal{MU}_{\delta=1}$ holds:*

1. *If F has a full clause then $F \in \mathcal{RHO}$.*
2. *For a full clause $C \in F$ and $x \in C$ holds $\mathrm{ld}_F(\overline{x}) = 1$.*
3. *If F has precisely two full clauses, then the (unique) clashing literal between these two full clauses must be 1-singular.*

Proof: For Part 1 we show that F has a saturation $G \in \mathcal{SMU}_{\delta=1}$ with the Horton-Strahler number $\mathrm{hs}(T(G)) \leq 1$, from which by Lemma 4.6.20 follows the assertion. As F has a full clause, by definition any saturation G of F has a full clause. Assume $\mathrm{hs}(T(G)) \geq 2$. Then there would be two 1-singular variables (a vertex whose children are leaves) which do not occur in a same clause and so G would not have any full clause.

Finally for Parts 2 and 3 we note that the vertices yielding a full clause in the structure tree $T(G)$ (which has $\mathrm{hs}(T(G)) \leq 1$) must be at the base of the structure tree (the two vertices with the largest depth). □

Indeed Corollary 4.6.5 together with Lemma 4.6.12 imply that all variables in an RHO-MU are singular:

Corollary 4.6.24 *For $F \in \mathcal{RHO} \cap \mathcal{MU}$ holds $\mathrm{var}(F) = \mathrm{var}_{\mathrm{s}}(F)$.*

Example 4.6.25 *The reverse of Corollary 4.6.24 does not hold, even if we require $F \in \mathcal{MU}_{\delta=1}$. Consider*

$$F := \{\{1,2,3\},\{1,2,-3\},\{1,-2,4\},\{1,-2,-4\},\\ \{-1,5\},\{-1,-5\}\} \in \mathcal{SMU}_{\delta=1}$$

with $\mathrm{hs}(T(F)) = 2$. We can obtain

$$G := \{\{1,3\},\{2,-3\},\{-2,4\},\{-2,-4\},\{-1,5\},\{-1,-5\}\} \in \mathcal{MU}_{\delta=1}$$

by partial marginalisation of F (Lemma 4.6.12), where all variables are singular. However as G has no unit-clause we have $F \notin \mathcal{RHO}$ (Lemma 4.6.15).

Some further observations concerning RHO-MUs (which will be used in Section 5.3 to characterise the clause-sets obtained by splitting nonsingular elements of 2–$\mathcal{MU}_{\delta=2}$) are as follows.

Lemma 4.6.26 *For $F \in \mathcal{SMU}$ holds: $F \in \mathcal{RHO}$ iff* $\mathrm{var}(F) = \mathrm{var}_{\mathrm{s}}(F)$.

Proof: The direction from left to right follows with Corollary 4.6.24 (also from the characterisation of saturated RHO-MUs in Lemma 4.6.22). So consider the reverse direction. Singular DP-reduction maintains saturatedness (Lemma 4.5.2, Part 2) and also can not increase literal degrees ([108, Corollary 26]). Therefore we obtain $\mathrm{sDP}(F) = \{\{\bot\}\}$, from which by Lemma 4.6.1 follows $\delta(F) = 1$. Using the structure tree $T(F)$, we see that the singular variables of F are precisely the near-leaves in $T(F)$ (Corollary 4.6.5) and so $T(F)$ has the Horton-Strahler number $\mathrm{hs}(T(F)) \le 1$. That is, $\mathrm{hd}(F) = \mathrm{hs}(T(F) \le 1$ (Lemma 4.6.18), and so by Lemma 4.6.20 follows the assertion. □

Lemma 4.6.27 *For $F \in \mathcal{MU}_{\delta=1}$ with precisely one 1-singular variable holds $F \in \mathcal{RHO}$.*

Proof: Assume $F \notin \mathcal{RHO}$. Thus by Lemma 4.6.20 for any saturation $G \in \mathcal{SMU}_{\delta=1}$ of F we have $\mathrm{hs}(T) \ge 2$. The structure trees with the Horton-Strahler number greater than 1 by Corollary 4.6.5 have at least two 1-singular variables, and thus obviously F has more than one 1-singular variables as well. □

We conclude this section by a basic result noted in [81], namely that all marginal elements of $\mathcal{MU}_{\delta=1}$ are renamable Horn clause-sets, and so have a unit-clause:

Lemma 4.6.28 ([81]) *$\mathcal{MMU}_{\delta=1} \subset \mathcal{RHO}$.*

Example 4.6.29 *We show that $\mathcal{MMU}_{\delta=1} \not\subset \mathcal{HO}$. For a clause-set $F := \{\{1,2\},\{1,-2\},\{-1,3\},\{-1,-3\}\} \in \mathcal{UHIT}_{\delta=1}$, we can obtain a marginalisation $G := \{\{1,2\},\{-2\},\{3\},\{-1,-3\}\}$ of F, where $G \notin \mathcal{HO}$, while $G \in \mathcal{RHO}$.*

Lemma 4.6.30 *$F \in \mathcal{MMU}_{\delta=1} \setminus \{\{\bot\}\}$ has at least two unit-clauses.*

Proof: All elements of $F \in \mathcal{MMU}_{\delta=1} \setminus \{\{\bot\}\}$ are in $\mathcal{RHO}$ (Lemmas 4.6.28) and so have a unit-clause (4.6.15). Let $C \in F$ be a unit-clause. All variables in F are 1-singular (Lemma 4.6.13). So there are precisely $2n(F)$ literal occurrences in F, and $2n(F) - 1$ literal occurrences in $F \setminus \{C\}$ with $c(F \setminus \{C\}) = n(F)$ clauses. Therefore $F \setminus \{C\}$ must contain at least one unit-clause. □

4.6.4 The conflict graph and marginal cases

In Section 4.6.2 we discussed a full classification of $\mathcal{SMU}_{\delta=1} = \mathcal{UHIT}_{\delta=1}$ as full binary trees using the structure tree. In this section we review another connection of clause-sets to graph theory which yields a full classification for the class $\mathcal{MMU}_{\delta=1}$, namely that the isomorphism types of $F \in \mathcal{MMU}_{\delta=1}$ are precisely the finite trees. This connection is established via the "conflict graph", where the conflict graph of a clause-set F has the clauses of F as vertices, with an edge joining two vertices iff they have at least one conflict.

The conflict graph and conflict "multigraphs" (allowing parallel edges) have been studied in the literature as a way of relating the conflict patterns of clause-sets to combinatorics and graph theory in both directions. A study of the "combinatorics of conflicts" for clause-sets has been initiated with [94] (with underlying report [93]) and continued with [57], [95], [98], [102] (the latter generalises the basic results from linear algebra applied to clause-sets regarding the conflict structure of clauses). In [94] the notion of "symmetric conflict matrix" was introduced (in the context of biclique partitions of multigraphs), which has an entry for each pair of clauses counting the number of conflicts between them. The conflict matrix interpreted as a graph, yields the conflict graph.

Conflict patterns of MUs have been also investigated in the literature. In [135] a lower bound on the number of edges in the conflict (multi)graph of a k-uniform MU F is provided, which is the same as number of clashes in F. This bound was later improved in [134]. The conflict-structure of clauses for unsatisfiable hitting clause-sets (which are saturated MUs) is studied in [94], and the class of conflict graphs for the elements of $\mathcal{MU}_{\delta=1}$ has been characterised in [103]. Here we follow the notation from [98].

Definition 4.6.31 ([98]) *The conflict graph of $F \in \mathcal{CLS}$, denoted by* $\mathbf{cg}(\boldsymbol{F})$*, is the graph where the set of vertices is F, and there is an edge between $C, D \in F$ if $C \cap \overline{D} \neq \emptyset$. That is,*

$$\begin{aligned} V(\mathrm{cg}(F)) &= F, \\ E(\mathrm{cg}(F)) &= \{\{C, D\} : C, D \in F \text{ and } C \cap \overline{D} \neq \emptyset\}. \end{aligned}$$

Example 4.6.32 *The conflict graph* $\mathrm{cg}(A_0)$ *is the trivial tree, and* $\mathrm{cg}(A_1)$ *is a path graph of length one. Consider $F := \{\{1\}, \{2\}, \{3\}, \{-1, -2, -3\}\} \in \mathcal{MMU}_{\delta=1}$, and also $G \in \mathcal{SMU}_{\delta=1}$ in Example 4.6.7. We have:*

$$\mathrm{cg}(F) = \begin{tikzcd} & \{2\} \arrow[d, dash] & \\ \{1\} \arrow[r, dash] & \{-1,-2,-3\} \arrow[r, dash] & \{3\} \end{tikzcd} \qquad \mathrm{cg}(G) = \begin{tikzcd} \{a,b\} \arrow[r, dash] \arrow[d, dash] \arrow[dr, dash] & \{a,\overline{b}\} \arrow[d, dash] \arrow[dl, dash] \\ \{\overline{a},\overline{c}\} \arrow[r, dash] & \{\overline{a},c\} \end{tikzcd}$$

In general we have:

Lemma 4.6.33 *For $F, G \in \mathcal{CLS}$ holds: $F \cong G \Rightarrow \mathrm{cg}(F) \cong \mathrm{cg}(G)$.*

Proof: Any isomorphism from F to G is a complement-preserving bijection $\alpha : \mathrm{lit}(F) \to \mathrm{lit}(G)$ which induces a bijection from the clauses of F to the clauses of G, i.e., $\alpha(F) = \{\alpha(C) : C \in F\} = G$ (Definition 2.4.1). By Definition 4.6.31 we have the same bijection $\alpha : V(\mathrm{cg}(F)) \to V(\mathrm{cg}(G))$. Furthermore for any $C, D \in F$ with $C \cap \overline{D} \neq \emptyset$, we have $\alpha(C) \cap \overline{\alpha(D)} \neq \emptyset$. That is, for any two adjacent vertices $C, D \in V(\mathrm{cg}(F))$, also vertices $\alpha(C)$ and $\alpha(D)$ are adjacent in $\mathrm{cg}(G)$. So we obtain $\mathrm{cg}(F) \cong \mathrm{cg}(G)$ (recall Section 3.2). $\square$

The reverse direction of Lemma 4.6.33 does not hold in general as the following example shows:

Example 4.6.34 *Consider* $F := \{\{1,2\},\{1,-2\},\{-1,3\},\{-1,-3\}\} \in \mathcal{SMU}_{\delta=1}$ *and* $G := \{\{1,2,3\},\{1,2,-3\},\{1,-2\},\{-1\}\} \in \mathcal{SMU}_{\delta=1}$. *The conflict graphs of these clause-sets are isomorphic to the complete graph (a graph where every pair of distinct vertices are connected with an edge) with four vertices, and so* $\mathrm{cg}(F) \cong \mathrm{cg}(G)$*; while clearly* $F \not\cong G$.

The conflict graph of any MU is connected (due to irredundancy). If there are multiple conflicts between clauses, the conflict-*graph* has only one edge between them. For $F \in \mathcal{MU}_{\delta=1}$, this is fully sufficient, since between two clauses there is at most one clash (see Corollary 4.6.14). Therefore the number of edges in $\mathrm{cg}(F)$ is the number of conflicts in $F \in \mathcal{MU}_{\delta=1}$.

Lemma 4.6.35 *Consider* $F \in \mathcal{MU}_{\delta=1}$. *If* $C \in F$ *is a leaf of* $\mathrm{cg}(F)$ *(has degree* 1*), then* $|C| = 1$.

Proof: Assume $|C| \geq 2$ (note that $C \neq \perp$). That is, there exist at least two different variables $v, w \in \mathrm{var}(C)$. W.l.o.g. assume $v, w \in C$. All variables in $\mathrm{var}(F)$ are non-pure. So there exist clauses $D_1, D_2 \in F$ with $\overline{v} \in D_1$ and $\overline{w} \in D_2$ and so $C \cap \overline{D}_1 \neq \emptyset$ and $C \cap \overline{D}_2 \neq \emptyset$. We have $D_1 \neq D_2$ since otherwise we would have $|C \cap \overline{D}_1| \geq 2$, which contradicts that C, D_1 have at most one clash (a basic property of $\mathcal{MU}_{\delta=1}$ in Corollary 4.6.14). Thus by Definition 4.6.31 the vertex C has at least two incident edges, contradicting that C is a leaf. $\square$

Lemma 4.6.36 *Consider* $F \in \mathcal{MMU}_{\delta=1}$ *and* $C \in F$. *Then* $|C| = 1$ *if and only if* C *is a leaf in* $\mathrm{cg}(F)$.

Proof: First assume C is a unit-clause in F. By Lemma 4.6.13 any variable $v \in \mathrm{var}(F)$ is 1-singular. So there exists exactly one clause $D \in F$ such that $C \cap \overline{D} \neq \emptyset$. Thus by Definition 4.6.31, vertex C has exactly one incident edge. The reverse direction follows by Lemma 4.6.35. $\square$

There are two extreme cases for conflict graphs of $F \in \mathcal{MU}_{\delta=1}$, namely conflict graphs of saturated and marginal elements. For $F \in \mathcal{SMU}_{\delta=1} = \mathcal{UHIT}_{\delta=1}$ every two different clauses clash (Corollary 4.6.5), and so the conflict graph is a complete graph (also noticed in [102], [103]):

Lemma 4.6.37 ([102]) *For $F \in \mathcal{MU}_{\delta=1}$ the conflict graph* $\mathrm{cg}(F)$ *is a complete graph iff F is saturated.*

Proof: Consider the direction from left to right. If $\mathrm{cg}(F)$ is complete, then there is an edge between any two vertices $C, D \in V(\mathrm{cg}(F))$ and by Definition 4.6.31, there is a conflict between any $C, D \in F$. Thus F is a hitting clause set and since $F \in \mathcal{MU}_{\delta=1}$, we have $F \in \mathcal{UHIT}_{\delta=1} = \mathcal{SMU}_{\delta=1}$.

Now consider the reverse direction. For $F \in \mathcal{SMU}_{\delta=1} = \mathcal{UHIT}_{\delta=1}$, there exists a conflict between any $C, D \in F$. Since by Definition 4.6.31 $V(\mathrm{cg}(F)) = \{C : C \in F\}$ and there is an edge between any $C, D \in F$ if and only if $|C \cap \overline{D}| \neq \emptyset$, each vertex C in $\mathrm{cg}(F)$ is connected to all vertices in $V \setminus \{C\}$. That is, $\mathrm{cg}(F)$ is complete. $\square$

We now come to the main characterisation of the class $\mathcal{MMU}_{\delta=1}$. We already showed that $F \in \mathcal{MU}_{\delta=1}$ is marginal iff every literal in F occurs exactly once (Lemma 4.6.13), that is, if there are altogether exactly $n(F)$ conflicts. It follows that F is marginal iff the number of edges in the conflict graph of F is $n(F)$, and since the conflict graph of F is connected and has $c(F) = n(F) + 1$ many nodes, we obtain that F is marginal iff the conflict graph of F is a tree. Furthermore it is shown in [103] that the isomorphism types of $F \in \mathcal{MMU}_{\delta=1}$ correspond exactly to the isomorphism types of finite trees.

Lemma 4.6.38 ([103]) *For $F \in \mathcal{MU}_{\delta=1}$ the conflict graph* $\mathrm{cg}(F)$ *is a tree if and only if F is marginal, and in case F is marginal, every tree can be realised in this way, and the isomorphism type of F is completely determined by $cg(F)$.*

In [129] it is shown that the number of different trees with n vertices is asymptotically equal to $D\alpha^{-n}n^{-5/2}$ where $\alpha = 0.3383219$ and $D = 0.5349485$. Furthermore, numerical data on the number of isomorphism types of trees with m vertices (and so the number of isomorphism types of $F \in \mathcal{MMU}_{\delta=1}$ with $c(F) = m$) is given in the OEIS ([139, Sequence A000055]), which starts as follows:

$$1, 1, 1, 2, 3, 6, 11, 23, 47, 106, 235, \ldots$$

Indeed the class $\mathcal{MMU}_{\delta=1}$ has polytime isomorphism decision (as finite trees have polytime isomorphism decision).

Corollary 4.6.39 ([81]) *The class $\mathcal{MMU}_{\delta=1}$ has polytime isomorphism decision.*

Finally we characterise $F \in \mathcal{MU}_{\delta=1}$ with the simplest case of conflict graphs, namely those MUs of deficiency 1 whose conflict graph is a path graph.

Lemma 4.6.40 *For $n \in \mathbb{N}_0$ consider*

$$F := \{\{1\}, \{-1, 2\}, \ldots, \{-(n-1), n\}, \{-n\}\} \in \mathcal{MMU}_{\delta=1}.$$

For $G \in \mathcal{MU}_{\delta=1}$ holds: $G \cong F$ iff $\mathrm{cg}(G)$ *is a path graph of length $n+1$.*

Proof: The conflict graph $\mathrm{cg}(F)$ is a path graph with $n+1$ vertices as follows:

$$\mathrm{cg}(F) = \ \{1\} \text{ —— } \{-1, 2\} \text{ —— } \ldots \text{ —— } \{-(n-1), n\} \text{ —— } \{-n\}$$

So we just need to show that $\mathrm{cg}(G)$ being a path graph implies that G is marginal, and then the assertion follows from Lemma 4.6.38. $\mathrm{cg}(G)$ has two leaves and so G has precisely two unit-clauses (Lemma 4.6.35); while every other clause in G is binary as non-leaf vertices of $\mathrm{cg}(G)$ have degree 2 (recall Corollary 4.6.14). Therefore G has precisely $2(n+1)-2 = 2n$ literal occurrences and so by Lemma 4.6.13 we have $G \in \mathcal{MMU}_{\delta=1}$. $\square$

In Lemma 5.4.3 we will show that clause-sets $F \in \mathcal{MU}_{\delta=1}$ with $\mathrm{cg}(F)$ isomorphic to a path graph are precisely 2-MUs with two unit-clauses, while in Section 6.3 we provide an alternative proof together with full classification of 2-MUs of deficiency 1.

Chapter 5

Minimal unsatisfiability and minimal strong digraphs

In Section 4.6 we discussed the most basic MUs which are those with deficiency 1, i.e., $F \in \mathcal{MU}_{\delta=1}$. And this whole class was explained by the expansion rule, which replaces a single clause C by two clauses $C' \cup \{v\}, C'' \cup \{\overline{v}\}$ for $C' \cup C'' = C$ and a new variable v, starting with the empty clause (Theorem 4.6.2). Also this class covers all RHO-MUs (Lemma 4.6.20). At the next level, there are two classes, namely $\mathcal{MU}_{\delta=2}$, and 2–$\mathcal{MU}$. As mentioned in Section 1.5.1, characterisations have been provided in the seminal paper [78] for the former class, and in the technical report [82] for the latter. We introduce in this chapter a new reasoning scheme based on graph theory, together with the first application, giving unifying and intuitive proofs for these two fundamental results.

In Section 5.1 the basic new class $\mathcal{FM} \subset \mathcal{MU}$ is introduced, and then the complexity of this class and its relation to the AllEqual boolean function are discussed. Section 5.2 introduces the most important new class of this chapter $\mathcal{DFM} \subset \mathcal{FM}$, which has a central place in $\mathcal{FM}$ due to its strong connection to graph theory (Theorem 5.2.9). Finally in Sections 5.3 and 5.4 the framework of DFM/FM is used to characterise classes $\mathcal{MU}'_{\delta=2}$ and 2–$\mathcal{MU}'$ via graph-theoretical reasoning.

5.1 MU with Full Monotone clauses (FM)

We first introduce formally the main classes of this chapter, $\mathcal{FM} \subset \mathcal{MU}$ (Definition 5.1.1) and $\mathcal{DFM} \subset \mathcal{FM}$ (Definition 5.2.1). Examples for these classes showed up in the literature, but these natural classes have not been studied yet.

Definition 5.1.1 *Let* $\boldsymbol{\mathcal{FM}}$ *be the set of* $F \in \mathcal{MU}$ *such that there is a full positive clause* $P \in F$ *and a full negative clause* $N \in F$ *(that is,* $\mathrm{var}(P) = \mathrm{var}(N) = \mathrm{var}(F)$*,* $P \subset \mathcal{VA}$*,* $N \subset \overline{\mathcal{VA}}$*). Using "monotone clauses" for positive* *and negative clauses, "$\mathcal{FM}$" denotes "full monotone". More generally, let* $\boldsymbol{\mathcal{FC}}$ *(denotes "full complementary") be the set of $F \in \mathcal{MU}$ such that there are full clauses $C, D \in F$ with $D = \overline{C}$.*

Example 5.1.2 *Examples of clause-sets in $\mathcal{FM}$ are the MUs A_n and $\mathcal{F}_n$.*

As mentioned before MUs are subsumption-free, and so in an FM every clause other than the full positive and negative clauses must be mixed (contains at least one positive and at least one negative literal). No element of $\mathcal{MU}_{\delta=1}$ with at least two variables can be in $\mathcal{FC}$, since clauses in any $\mathcal{MU}_{\delta=1}$ have at most one clash. Obviously an $F \in \mathcal{FC}$ contains a unit-clause iff $n(F) = 1$. Both classes $\mathcal{FM}$ and $\mathcal{FC}$ are stable under partial saturation and under such partial marginalisations which do not touch the full monotone clauses resp. some pair of complementary full clauses.

The closure of $\mathcal{FM}$ under isomorphism is $\mathcal{FC}$. In the other direction, for any $F \in \mathcal{FC}$ and any pair $C, D \in F$ of full clauses with $D = \overline{C}$ (note that in general such a pair is not unique), flip the signs so that C becomes a positive clause (and so D becomes a negative clause), and we obtain an element of $\mathcal{FM}$.

As usual we call the subsets of nonsingular elements $\boldsymbol{\mathcal{FM}'}$ resp. $\boldsymbol{\mathcal{FC}'}$. The *trivial elements* of $\mathcal{FM}$ and $\mathcal{FC}$ are the MUs with at most one variable: $\mathcal{FM}_{n\le 1} = \mathcal{FM}_{\delta=1} = \mathcal{FC}_{n\le 1} = \mathcal{FC}_{\delta=1} = \{\{\bot\}\} \cup \{\{v\}, \{\overline{v}\} : v \in \mathcal{VA}\}$. The singular cases in $\mathcal{FM}$ and $\mathcal{FC}$ are just these cases with only one variable:

Lemma 5.1.3 *$\mathcal{FM}' = \mathcal{FM}_{n\neq 1} = \mathcal{FM}_{\delta\ge 2} \cup \{\{\bot\}\}$, $\mathcal{FC}' = \mathcal{FC}_{n\neq 1} = \mathcal{FC}_{\delta\ge 2} \cup \{\{\bot\}\}$.*

Proof: Assume that there is a singular $F \in \mathcal{FC}$ with $n(F) \ge 2$. Let C, D be full complementary clauses in F. W.l.o.g. we can assume that there is $x \in C$ (so $\overline{x} \in D$) such that literal x only occurs in C. Consider now some $y \in D \setminus \{\overline{x}\}$ (exists due to $n(F) \ge 2$). There exists a satisfying assignment φ for $F' := F \setminus \{D\}$, and it must hold $\varphi(x) = 1$ and $\varphi(y) = 0$ (otherwise F would be satisfiable). Obtain φ' by flipping the value of x. Now φ' still satisfies F', since the only occurrence of literal x is C, and this clause contains $\overline{y}$, but now φ' satisfies F. □

So the study of $\mathcal{FM}$ is about studying special nonsingular MUs. In general we prefer to study $\mathcal{FM}$ over $\mathcal{FC}$, as we can define the "core" as a sub-clause-set:

Definition 5.1.4 *For $F \in \mathcal{FM}$ there is exactly one positive clause $P \in F$, and exactly one negative clause $N \in F$ (otherwise there would be subsumptions in F), and we call $F \setminus \{P, N\}$ the* ***core*** *of F.*

We note that cores consist only of mixed clauses, and in general any mixed clause-set (consisting only of mixed clauses) has always at least two satisfying assignments, the all-0 and the all-1 assignments. By Lemma 5.1.3 we get:

Corollary 5.1.5 *Consider $F \in \mathcal{FM}$ and its core F'. Then we have $F' = \top$ iff $n(F) \leq 1$, while for $n(F) \geq 2$ holds* $\mathrm{var}(F') = \mathrm{var}(F)$.

By definition, $\mathcal{FC}$ and $\mathcal{FM}$ are stable under (partial) saturation. Moreover they are stable under applications of DP-reduction in the following sense:

Lemma 5.1.6 *Consider $F \in \mathcal{MU}$, $v \in \mathrm{var}(F)$ and $F' \in \mathcal{MU}$ with $F' \subseteq \mathrm{DP}_v(F)$. Then we have $F \in \mathcal{FC} \Rightarrow F' \in \mathcal{FC}$, and $F \in \mathcal{FM} \Rightarrow F' \in \mathcal{FM}$.*

Proof: The resolvent of a full clause C with any other clause is $C' := C \setminus \{v, \overline{v}\}$, which is a full clause in $\mathrm{DP}_v(F)$; if this clause would become superfluous, then it would have been superfluous originally. □

5.1.1 The AllEqual function

We now turn to the semantics of the core:

Definition 5.1.7 *For a finite $V \subset \mathcal{VA}$ the **AllEqual** function on V is the boolean function which is true for a total assignment of V if all variables are assigned the same value, and false otherwise.*

*A **CNF-realisation** of AllEqual on V is a clause-set F with $\mathrm{var}(F) \subseteq V$, which is as a boolean function the AllEqual function on V.*

Obviously for $|V| \leq 1$ the only CNF-realisation of AllEqual on V is $\top$, while for $|V| \geq 2$ any realisation F must have $\mathrm{var}(F) = V$. The core of every FM F realises AllEqual on $\mathrm{var}(F)$ irredundantly, and this characterises $\mathcal{FM}$ as follows:

Theorem 5.1.8 *Consider $F \in \mathcal{CLS}$ with a full positive clause $P \in F$ and a full negative clause $N \in F$, and let $F' := F \setminus \{P, N\}$. Then $F \in \mathcal{FM}$ if and only if F' realises AllEqual on $\mathrm{var}(F)$, and F' is irredundant.*

Proof: First assume $F \in \mathcal{FM}$ (so F' is the core of F). Clearly F' is irredundant (as a subset). And since F is MU, F' as a boolean function must have exactly the satisfying assignments as forbidden by P, N in F, that is, F' realises AllEqual on $\mathrm{var}(F)$. Now assume that F' realises AllEqual on $\mathrm{var}(F)$, and F' is irredundant. We show $F \in \mathcal{MU}$, i.e., F is irredundant and unsatisfiable. The latter follows from the presence of P, N in F. While the irredundancy follows from the fact, that the set of falsifying assignments of clauses P, N are disjoint from each other and from the sets of falsifying assignments of the clauses of F'. □

So we can create exactly all FMs F by considering all irredundant clause-sets F' realising AllEqual, and letting $F := F' \cup \{\mathrm{var}(F), \overline{\mathrm{var}(F)}\}$.

5.1.2 The decision complexity

The decision complexity of $\mathcal{FM}$ is the same as that of $\mathcal{MU}$ (which is the same as $\mathcal{MU}'$), which has been determined in [130, Theorem 1] as complete for the class D^P (whose elements are differences of NP-problems):

Theorem 5.1.9 *For $F \in \mathcal{CLS}$, the decision whether "$F \in \mathcal{FM}$?" is D^P-complete.*

Proof: The decision problem is in D^P, since $F \in \mathcal{FM}$ iff F is irredundant with full monotone clauses and $F \notin \mathcal{SAT}$ (by [130] the decision problem whether a clause-set is irredundant is NP-complete).

To show hardness we reduce $\mathcal{MU}$ to $\mathcal{FM}$ by constructing a polytime function $t : \mathcal{CLS} \to \mathcal{CLS}$ such that for all $F \in \mathcal{CLS}$ holds $t(F) \in \mathcal{FM}$ if and only if $F \in \mathcal{MU}$. For an input clause-set F with $n := n(F)$ we consider any MU with full monotone clauses, e.g. $\mathcal{F}_n$, and w.l.o.g. we assume $\mathrm{var}(F) = \{1, \dots, n\}$. We define F' as the disjunction of two clause-sets obtained by adding a new variable $n+1$ positively to every $C \in \mathcal{F}_n$ and negatively to every $C \in F$. So F' has a full positive clause, and it is easy to see that $\langle (n+1) \to 0 \rangle * F', \langle (n+1) \to 1 \rangle * F' \in \mathcal{MU}$. So by Lemma 4.4.4 and Lemma 4.4.3, Part 1 we have $F \in \mathcal{MU} \iff F' \in \mathcal{MU}$.

Now to create a full negative clause, we define F'' as the disjunction of the two clause-sets obtained by adding a new variable positively to every clause in F' and negatively to every clause in $\mathcal{F}_{n+1}$. Again we have $F' \in \mathcal{MU} \iff F'' \in \mathcal{MU}$, and so $F \in \mathcal{MU} \iff F'' \in \mathcal{MU}$. Since performing these operations can be done in linear time, we can reduce MU to FM with a polytime reduction, and therefore the decision of FM is D^P-complete. $\square$

The use of $\mathcal{F}_n$ in the proof of Theorem 5.1.9 could be replaced by any scalable family in $\mathcal{FM}$. The reduction of $\mathcal{MU}$ to $\mathcal{FM}$ in the proof of Theorem 5.1.9 shows a "spreading" of deficiencies, namely $\delta(F'') = c(F) + n + 3$, and thus motivates our belief that classifying the levels $\mathcal{FM}_{\delta=k}$ is a useful stepping stone towards the classification of $\mathcal{MU}'_{\delta=k}$.

Corollary 5.1.10 *The decision "$F \in \mathcal{FC}$?" is D^P-complete.*

5.2 FM with binary clauses (DFM)

Definition 5.2.1 $\boldsymbol{\mathcal{DFM}}$ *is the subset of $\mathcal{FM}$ where the core (Definition 5.1.4) is in 2–$\mathcal{CLS}$, while* $\boldsymbol{\mathcal{DFC}}$ *is the set of $F \in \mathcal{FC}$, such that there are full complementary clauses $C, D \in F$ with $F \setminus \{C, D\} \in 2$–$\mathcal{CLS}$.*

The core of DFMs consists of clauses of length exactly 2. Examples in $\mathcal{DFM}$ are the $\mathcal{F}_n$. $\mathcal{DFC}$ is the closure of $\mathcal{DFM}$ under isomorphism. We will show a strong connection between these new classes and a class of digraphs in Theorem 5.2.9, and so "D" in $\mathcal{DFM}$ (and $\mathcal{DFC}$) denotes "digraph". For $\mathcal{DFC}$ and $\mathcal{DFM}$ we have stability under applications of DP:

Lemma 5.2.2 *Consider $F \in \mathcal{MU}$, $v \in \mathrm{var}(F)$ and $F' \in \mathcal{MU}$ with $F' \subseteq \mathrm{DP}_v(F)$. Then we have $F \in \mathcal{DFC} \Rightarrow F' \in \mathcal{DFC}$, and $F \in \mathcal{DFM} \Rightarrow F' \in \mathcal{DFM}$.*

Proof: By Lemma 5.1.6 we know $F' \in \mathcal{FC}$. The resolution of two clauses of length at most two yields a clause of length at most 2, while the resolvent of a full (positive/negative) clause is a full (positive/negative) clause. $\square$

Definition 5.2.3 *For $F \in \mathcal{DFM}$ the **positive implication digraph** $\mathbf{pdg}(\boldsymbol{F})$ is obtained as follows:*

1. *The vertex set is $\mathrm{var}(F)$, i.e., $V(\mathrm{pdg}(F)) := \mathrm{var}(F)$.*
2. *The arcs are the implications on the variables as given by the core F' of F, i.e., $E(\mathrm{pdg}(F)) := \{(a, b) : \{\overline{a}, b\} \in F', a, b \in \mathrm{var}(F)\}$.*

This can also be applied to any mixed binary clause-set F (note that the core F' is such a mixed binary clause-set).

Example 5.2.4 *The positive implication digraph of*

$$\begin{aligned}\mathcal{F}_6 = \{&\{1, 2, 3, 4, 5, 6\}, \{-1, -2, -3, -4, -5, -6\},\\ &\{-1, 2\}, \{-2, 3\}, \{-3, 4\}, \{-4, 5\}, \{-5, 6\}, \{-6, 1\}\} \in \mathcal{DFM}\end{aligned}$$

is a cycle digraph as follows:

$$\mathrm{pdg}(\mathcal{F}_6) = \begin{tikzcd} 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[d] \\ 6 \arrow[u] & 5 \arrow[l] & 4 \arrow[l] \end{tikzcd}$$

Note that the empty clause-set $\top$ yields the empty digraph (which is MSD). The essential feature of mixed clause-sets $F \in 2\text{–}\mathcal{CLS}$ is that for a clause $\{\overline{v}, w\} \in F$ we only need to consider the "positive interpretation" $v \to w$, not the "negative interpretation" $\overline{w} \to \overline{v}$, since the positive literals and the negative literals do not interact. So we do not need the (full) implication digraph (which we will discuss in Chapter 6). Via the positive implication digraphs we can understand when a mixed clause-set realises AllEqual.

Recall that every digraph G with $|V(G)| \leq 1$ is a minimal strong digraph (Section 3.1). Now we are ready to formulate the following lemma:

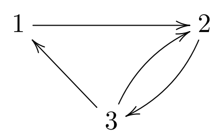
Lemma 5.2.5 *For a mixed binary clause-set F holds:*

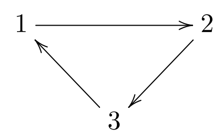
1. *F is a CNF-realisation of AllEqual iff $\mathrm{pdg}(F)$ is a strong digraph (SD).*
2. *F is an irredundant CNF-realisation of AllEqual iff $\mathrm{pdg}(F)$ is a minimal strong digraph (MSD).*

Proof: The main point here is that the resolution operation for mixed binary clauses $\{\overline{a}, b\}, \{\overline{b}, c\}$, resulting in $\{\overline{a}, c\}$, corresponds exactly to the formation of transitive arcs, i.e., from $(a, b), (b, c)$ we obtain (a, c). So the two statements of the lemma are just easier variations on the standard treatment of logical reasoning for 2-CNFs via "path reasoning". $\square$

Example 5.2.6 *Consider* $F_1 := \{\{-1, 2\}, \{-2, 3\}, \{-3, 1\}, \{-3, 2\}\} \in 2\text{–}\mathcal{CLS}$*, which is a (redundant) CNF-realisation of AllEqual on variables* $1, 2, 3$*. Then* $\mathrm{pdg}(F_1)$*, as shown below, is an SD with the set of vertices* $\{1, 2, 3\}$ *and the number of arcs the same as the number of clauses in* F *(so* $\delta(\mathrm{pdg}(F_1)) = \delta(F_1)$*). It is clear that the result of removing the clause* $\{-3, 2\}$ *is still an SD (i.e.,* $\mathrm{pdg}(F_1)$ *is not MSD).*

Now let F_2 *be the clause-set obtained by removing the clause* $\{-3, 2\}$ *from* F_1*. It is easy to see that* $F_2 = \{\{-1, 2\}, \{-2, 3\}, \{-3, 1\}\}$ *is irredundant and AllEqual on the same variables. Then* $\mathrm{pdg}(F_2)$ *is a cycle digraph, shown below, which is MSD.*

$\mathrm{pdg}(F_1) =$ 

$\mathrm{pdg}(F_2) =$ 

As explained before, $F \mapsto \mathrm{pdg}(F)$ converts mixed binary clause-sets with full monotone clauses to a digraph. Also the reverse direction is easy:

Definition 5.2.7 *Consider a finite digraph* G *with* $V(G) \subset \mathcal{VA}$*. Then the clause-set* $\mathbf{mcs}(\boldsymbol{G}) \in \mathcal{CLS}$ *("m" like "monotone") is obtained by interpreting the arcs* $(a, b) \in E(G)$ *as binary clauses* $\{\overline{a}, b\} \in \mathrm{mcs}(G)$*, and adding the two full monotone clauses* $\{V(G), \overline{V(G)}\} \subseteq \mathrm{mcs}(G)$*. That is,*

$$\mathrm{mcs}(G) := \{\{\overline{a}, b\} : (a, b) \in E(G)\} \cup \{V(G), \overline{V(G)}\} \in \mathcal{CLS}.$$

Example 5.2.8 *Consider a digraph* $G = (E, V)$ *with* $V(G) = \{1, 2, 3\}$ *and* $E(G) = \{(1, 2), (2, 1), (2, 3), (3, 2)\}$*. In the formation of* $\mathrm{mcs}(G)$*, an arc from vertex* 1 *to vertex* 2 *becomes the logical implication* $1 \to 2$*, represented by* $(-1 \vee 2)$ *(or as a set,* $\{-1, 2\}$*). So by adding the full monotone clauses over* $V(G)$*, we obtain*

$$\mathrm{mcs}(G) = \{\{-1, 2\}, \{-2, 1\}, \{-2, 3\}, \{-3, 2\}, \{1, 2, 3\}, \{-1, -2, -3\}\} \in \mathcal{CLS}$$

with $\mathrm{var}(\mathrm{mcs}(G)) = V(G) = \{1, 2, 3\}$ *and* $c(\mathrm{mcs}(G)) = |E(G)| + 2$ *(and so* $\delta(\mathrm{mcs}(G)) = \delta(G) + 2$*).*

We now show that DFMs and MSDs are basically the "same thing", only using different languages, which is now formulated as follows:

Theorem 5.2.9 *The two formations* $F \mapsto \mathrm{pdg}(F)$ *and* $G \mapsto \mathrm{mcs}(G)$ *are inverse to each other, that is,*

1. $\mathrm{mcs}(\mathrm{pdg}(F)) = F$ *for all* $F \in \mathcal{DFM}$*,*
2. *and* $\mathrm{pdg}(\mathrm{mcs}(G)) = G$ *for all MSDs* G *with* $V(G) \subset \mathcal{VA}$*.*

For every $F \in \mathcal{DFM}$ *the digraph* $\mathrm{pdg}(F)$ *is an MSD, and for every MSD* G *with* $V(G) \subset \mathcal{VA}$ *we have* $\mathrm{mcs}(G) \in \mathcal{DFM}$*.*

Proof: For the map $G \mapsto \mathrm{mcs}(G)$, we use the vertices of G as the variables of $\mathrm{mcs}(G)$ (Definition 5.2.7). An arc (a, b) naturally becomes a mixed binary clause $\{\overline{a}, b\}$, and we obtain the set F' of mixed binary clauses, where by definition we have $\mathrm{pdg}(F') = G$ (see Definition 5.2.3). That is, $\mathrm{pdg}(\mathrm{mcs}(G)) = G$, and similarly we obtain $\mathrm{mcs}(\mathrm{pdg}(F')) = F'$. This yields a bijection between the set of finite digraphs G with $V(G) \subset \mathcal{VA}$ and the set of mixed binary clause-sets. By Lemma 5.2.5 Part 2, minimal strong connectivity of G is equivalent to F' being an irredundant AllEqual-representation. So there is a bijection between MSDs and the set of mixed binary clause-sets which are irredundant AllEqual-representation. We "complete" the AllEqual-representations to MUs, by adding the full monotone clauses, and we get the DFM $\mathrm{mcs}(G)$ (recall Theorem 5.1.8). □

Theorem 5.2.9 can be considerably strengthened, by including other close relations, but here we formulated only what we need. For a DFM $F \neq \{\bot\}$ and an MSD $G \neq (\emptyset, \emptyset)$ we obtain $\delta(\mathrm{pdg}(F)) = \delta(F) - 2$ and $\delta(\mathrm{mcs}(G)) = \delta(G) + 2$, where $\delta(G)$ is the deficiency of G (Definition 3.1.2). Concerning isomorphisms there is a small difference between the two domains, since the notion of clause-set isomorphism includes flipping of variables, which for DFMs can be done all at once (flipping "positive" and "negative") — this corresponds in $\mathrm{pdg}(F)$ to the reversal of the direction of all arcs. For our two main examples, cycle digraphs and dipaths, this yields an isomorphic digraph, but this is not the case in general.

Marginalisation of DFMs concerns only the full monotone clauses and not the binary clauses, formulated as follows:

Lemma 5.2.10 *Consider a clause-set* F *obtained by partial marginalisation of a non-trivial DFM* F' *(Section 4.2). Then* F *has no unit-clause and its formation did not touch binary clauses but only shortened its monotone clauses.*

Proof: By definition, marginalisation can be reordered. So assume that partial marginalisation is done for a binary clause at first. It is clear that the resulting unit-clause would subsume one of the full monotone clauses. Thus F contains no unit-clause and is obtained by partial marginalisation of the positive or the negative clause. □

By Theorem 5.2.9, deciding whether for $F \in \mathcal{CLS}$ holds $F \in \mathcal{DFM}$ can be done in polynomial time: Check whether we have the two full monotone clauses, while the rest are binary clauses, if yes, translate the binary clauses to a digraph and decide whether this digraph is an MSD (which can be done in quadratic time; recall that deciding the SD property can be done in linear time). If yes, then $F \in \mathcal{DFM}$, otherwise $F \notin \mathcal{DFM}$.

We now come to the two simplest example classes, cycle digraphs and dipaths.

5.2.1 Cycle digraphs

An easy observation is that the cycle digraphs CD_n (Section 3.3) have the minimum deficiency zero among MSDs.

Lemma 5.2.11 *Consider an MSD G with at least two vertices. Then $\delta(G) = |E(G)| - |V(G)| \geq 0$, where $\delta(G) = 0$ if and only if G is a cycle digraph.*

Proof: All vertices in G have in-degree and out-degree at least one, and so there must be at least $|V(G)|$ arcs, i.e., $\delta(G) \geq 0$. A cycle digraph is an MSD with deficiency 0. In the reverse direction, for an MSD G, $\delta(G) = 0$ implies that every vertex has in-degree and out-degree both exactly one, and thus G is a cycle digraph. $\square$

We obtain the basic class $\mathcal{F}_n$, and we can now uncover its underlying graph structure:

Definition 5.2.12 *Let $\boldsymbol{\mathcal{F}_n} := \mathrm{mcs}(\mathrm{CD}_n) \in \mathcal{DFM}$ for $n \geq 2$ (Definition 5.2.7). That is,*

$$\mathcal{F}_n = \{\{-1, 2\}, \dots, \{-(n-1), n\}, \{-n, 1\}, \{1, \dots, n\}, \{-1, \dots, -n\}\} \in \mathcal{DFM}.$$

We have $\mathcal{F}_2 = A_2$ and $\delta(\mathcal{F}_n) = 2$. The DFMs of deficiency 2 are the $\mathcal{F}_n$:

Lemma 5.2.13 *For $F \in \mathcal{DFM}_{\delta=2}$ holds $F \cong \mathcal{F}_{n(F)}$.*

Proof: By Theorem 5.2.9 the positive implication digraph $\mathrm{pdg}(F)$ is an MSD with the deficiency $\delta(F) - 2 = 0$, and thus is a cycle digraph (Lemma 5.2.11) of length $n(F)$. $\square$

It is known that all $\mathcal{F}_n$ are saturated, but for completeness we give a proof:

Lemma 5.2.14 *For every $n \geq 2$, $\mathcal{F}_n$ is saturated.*

Proof: We show that adding a literal x to any clause $C \in \mathcal{F}_n$ introduces a satisfying assignment, i.e., $\mathcal{F}_n$ is saturated. Note that the positive and the negative clauses are already full, and saturation can only touch the mixed binary clauses. Recall $\mathrm{var}(\mathcal{F}_n) = \{1, \dots, n\}$. Due to symmetry assume $C = \{-n, 1\}$, and we add $x \in \{2, \dots, n-1\}$ to C. Let φ be the total assignment where all variables $2, \dots, n$ are set to true while 1 is set to false. Then φ satisfies the monotone clauses and the new clause $\{-n, 1, x\}$. Every literal occurs only once in the core of $\mathcal{F}_n$ and so literal 1 occurs only in C. So φ satisfies every mixed clause in $F \setminus \{C\}$ (which has a positive literal other than 1). $\square$

5.2.2 Ditrees and dipaths

Recall that ditrees are the directed version of trees (Section 3.3), and it is easy to see that for every tree G the digraph $\mathrm{dg}(G)$ is an MSD. Now consider the path digraph $\mathrm{PD}_n = (\{1, \dots, n\}, \{(1, 2), \dots, (n-1, n)\}$ for $n \in \mathbb{N}_0$.

Definition 5.2.15 *Let* $\mathbf{DB}_{\boldsymbol{n}} := \mathrm{mcs}(\mathrm{dg}(\mathrm{PD}_n)) \in \mathcal{DFM}$ *(*$n \in \mathbb{N}_0$*) (Definition 5.2.7). That is,*

$$\begin{aligned} \mathrm{DB}_n \quad = \quad & \{\{-1, 2\}, \{1, -2\}, \dots, \{-(n-1), n\}, \{n-1, -n\}, \\ & \{-1, \dots, -n\}, \{1, \dots, n\}\} \in \mathcal{DFM}. \end{aligned}$$

So $\mathrm{DB}_n = A_n$ for $n \leq 2$, while in general $n(\mathrm{DB}_n) = n$, and for $n \geq 1$ holds $c(\mathrm{DB}_n) = 2 + 2(n-1) = 2n$, and $\delta(\mathrm{DB}_n) = n$.

Example 5.2.16 *By Definition 5.2.15 for the dipath*

$$\mathrm{dg}(\mathrm{PD}_4) = \begin{tikzcd} 1 \arrow[r, bend left] & 2 \arrow[l, bend left] \arrow[r, bend left] & 3 \arrow[l, bend left] \arrow[r, bend left] & 4 \arrow[l, bend left] \end{tikzcd}$$

we obtain $\mathrm{DB}_4 = \mathrm{mcs}(\mathrm{dg}(\mathrm{PD}_4))$ *as follows:*

$$\begin{aligned} \mathrm{DB}_4 = \{ & \{-1, 2\}, \{1, -2\}, \{-2, 3\}, \{2, -3\}, \{-3, 4\}, \{3, -4\}, \\ & \qquad\qquad \{1, 2, 3, 4\}, \{-1, , -2, -3, -4\}\} \in \mathcal{DFM} \end{aligned}$$

DB_n for $n \neq 1$ is nonsingular, and every variable in $\mathrm{var}(\mathrm{DB}_n) \setminus \{1, n\}$ is of degree 6 for $n \geq 2$, while the variables $1, n$ (which are the endpoints of the dipath) have degree 4.

Example 5.2.17 *Consider* DB_n *and obtain* F *by the DP-reduction on a variable* v*. First assume that* $v \in \{1, n\}$ *(the two variables of degree* 4*), and w.l.o.g. consider the DP-reduction on variable* n*. Then* $\mathrm{DP}_n(F)$ *changes only the clauses containing literals* $n, -n$*, namely the full monotone clauses plus* $\{-(n-1), n\}, \{n-1, -n\}$*. These four clauses are replaced by the two monotone clauses* $\{1, \dots, n-1\}, \{-1, \dots, -(n-1)\}$ *and we get* $F = \mathrm{DB}_{n-1}$*. Note that* $\delta(F) = \delta(\mathrm{DB}_n) - 1$*. Otherwise,* v *is of degree* 6 *and occurs in the monotone clauses plus four mixed binary clauses. W.l.o.g. consider variable* 2*. Then DP-reduction replaces these six clauses with two monotone clauses* $\{1, 3, \dots, n-1\}, \{-1, -3, \dots, -(n-1)\}$ *and two mixed clauses* $\{1, -3\}, \{-1, 3\}$*, where* $F \cong \mathrm{DB}_{n-1}$.

Among ditrees, only dipaths can be marginalised to nonsingular 2-uniform MUs, since dipaths are the only ditrees with exactly two linear vertices. The unique marginal MUs obtained from dipaths are as follows:

Definition 5.2.18 *For $n \geq 2$ obtain the uniform $\boldsymbol{\mathcal{B}_n} \in 2\text{-}\mathcal{MU}$ from* DB_n *by replacing the full positive/negative clause with $\{1,n\}$ resp. $\{-1,-n\}$, i.e.,*

$$\mathcal{B}_n := \{\{-1,-n\},\{1,n\},\{-1,2\},\{1,-2\},\dots,\{-(n-1),n\},\{n-1,-n\}\}.$$

We note that $\mathcal{B}_n$ is a complement-invariant marginalisation of DB_n for $n \geq 2$, and every literal of $\mathcal{B}_n$ has degree 2 (occurs exactly twice). Indeed the $\mathcal{B}_n$ are (precisely) the marginalisations of the DB_n.

Lemma 5.2.19 *For $n \geq 2$,* DB_n *has a unique marginalisation, namely $\mathcal{B}_n$.*

Proof: Let F be a marginalisation of DB_n, $n \geq 2$. By Lemma 5.2.10 we know that marginalisation affects only the full positive and the full negative clauses of DB_n, which become $P \in F$ resp. $N \in F$, while the rest stays unchanged. We have to show $P = \{1,n\}$ and $N = \{-1,-n\}$, and for that it suffices to show $1,n \in P$ and $-1,-n \in N$. By symmetry w.l.o.g. we only need to show $1 \in P$, and so assume $1 \notin P$. Now literal 1 only occurs once in F, and variable 1 is singular in F. The occurrences of variable 1 in the (unchanged) mixed binary clauses of F are in $\{-1,2\},\{1,-2\}$. So the resolution of $\{-1,2\},\{1,-2\}$ is tautological, but singular DP-reduction for MUs can not have tautological resolvents (see Section 4.5). □

5.3 Deficiency 2 revisited

We now come to the first main application of the new class $\mathcal{DFM}$, and we give a new and relatively short proof, that MUs $\mathcal{F}_n$ are precisely the nonsingular MUs of deficiency 2. The core combinatorial-logical argument is to show $\mathcal{MU}'_{\delta=2} \subseteq \mathcal{FC}_{\delta=2}$, i.e., every $F \in \mathcal{MU}'_{\delta=2}$ must have two full complementary clauses $C, D \in F$. The connection to the "geometry" then is established by showing $\mathcal{FM}_{\delta=2} \subseteq \mathcal{DFM}_{\delta=2}$, i.e., if an FM F has deficiency 2, then it must be a DFM, i.e., all clauses besides the full monotone clauses are binary. The pure geometrical argument is the characterisation of $\mathcal{DFM}_{\delta=2}$, which has already been done in Lemma 5.2.13.

The proof of the existence of full clauses $C, D = \overline{C}$ in F is based on the Splitting Ansatz (see Section 1.5.2). Since $\mathcal{MU}_{\delta=2}$ is stable under saturation, we can start with a saturated F, and can split on any variable (though later an argument is needed to undo saturation). There must be a variable v occurring at most twice positively as well as negatively (otherwise the basic lemma $\delta(F) \geq 1$ for any MU F would be violated), and due to nonsingularity v occurs exactly twice positively and negatively. The splitting instances F_0, F_1 have deficiency 1. So they have at least one 1-singular variable. There is very little "space" to reduce a nonsingular variable in F to a 1-singular variable in F_0 resp. F_1, and indeed those two clauses whose vanishing in F_0 do this, are included in F_1, and vice versa. Since clauses in $\mathcal{MU}_{\delta=1}$ have at most one clash, F_0, F_1 have exactly one 1-singular variable. And so by the geometry of the structure trees (resp. their Horton-Strahler numbers), both F_0, F_1 are in fact renamable Horn! Thus every variable in F_0, F_1 is singular, and F_0, F_1 must contain a unit-clause. Again considering both sides, it follows that the (two) positive occurrences of v must be a binary clause (yielding the unit-clause) and a full clause C (whose vanishing yields the capping of all variables to singular variables), and the same for the (two) negative occurrences, yielding D. So $F_0, F_1 \in \mathcal{RHO}$ both contain a full clause and we know that the complements of the literals in the full clause occur exactly once in F_0 resp. F_1. Thus in fact C resp. D have the "duty" of removing each others complement, and we get $D = \overline{C}$.

Now consider $F \in \mathcal{FM}_{\delta=2}$ with monotone full clauses $C, D \in F$. Transform the core F' within F into an equivalent F'', by replacing each clause in F' by a contained prime implicate of F', which, since the core means that all variables are equal (semantically), is binary. So we arrive in principle in $\mathcal{DFM}$, but we could have created redundancy, and this can not happen, since an MSD has minimum deficiency 0. The details are as follows:

Theorem 5.3.1 $\mathcal{DFC}_{\delta=2} = \mathcal{FC}_{\delta=2} = \mathcal{MU}'_{\delta=2}$.

Proof: By definition and Lemma 5.1.3 we have $\mathcal{DFC}_{\delta=2} \subseteq \mathcal{FC}_{\delta=2} \subseteq \mathcal{MU}'_{\delta=2}$. First we show $\mathcal{MU}'_{\delta=2} \subseteq \mathcal{FC}_{\delta=2}$, that is, every nonsingular $F \in \mathcal{MU}$ with $\delta(F) = 2$ contains, up to flipping of signs, a full positive and a full negative clause.

MUs with deficiency 2 are non-trivial and so $n(F) \geq 2$. We know that F has a variable $v \in \mathrm{var}(F)$ of degree at most 4 (recall Lemma 4.3.4). Then nonsingularity of F implies that every variable in F is nonsingular with degree at least 4; so v is a nonsingular variable of degree 4. Let $C_1, C_2 \in F$ be the two clauses containing the literal v and $D_1, D_2 \in F$ be the two clauses containing the literal $\overline{v}$.

We assume that F is saturated (note that saturation maintains minimal unsatisfiability and deficiency). By the Splitting Ansatz (recall Lemma 4.4.3), $F_0 := \langle v \to 0 \rangle * F \in \mathcal{MU}_{\delta=1}$ and $F_1 := \langle v \to 1 \rangle * F \in \mathcal{MU}_{\delta=1}$ (due to F being nonsingular, by Corollary 4.4.7 splitting strictly reduces the deficiency). So F_0 removes D_1, D_2 and shortens C_1, C_2, while F_1 removes C_1, C_2 and shortens D_1, D_2 as follows:

$$\begin{aligned} F_0 &= (F \setminus \{C_1, C_2, D_1, D_2\}) \cup \{C_1 \setminus \{v\}, C_2 \setminus \{v\}\} \in \mathcal{MU}_{\delta=1}, \\ F_1 &= (F \setminus \{C_1, C_2, D_1, D_2\}) \cup \{D_1 \setminus \{\overline{v}\}, D_2 \setminus \{\overline{v}\}\} \in \mathcal{MU}_{\delta=1}. \end{aligned}$$

We want to show that $F_0, F_1 \in \mathcal{RHO}$. Both F_0, F_1 contain a 1-singular variable, called a resp. b. We obtain $\{a, \overline{a}\} \subseteq D_1 \cup D_2$, since F has no singular variable and only by removing D_1, D_2 the degree of a decreased to 2. Similarly $\{b, \overline{b}\} \subseteq C_1 \cup C_2$. In $\mathcal{MU}_{\delta=1}$ any two clauses have at most one clash (Corollary 4.6.14), and thus indeed F_0, F_1 have each exactly one 1-singular variable. Now $F_0, F_1 \in \mathcal{MU}_{\delta=1}$ with exactly one 1-singular variable are renamable Horn clause-sets (Lemma 4.6.27).

Now we determine the length of C_1, C_2, D_1, D_2. Since unsatisfiable Horn clause-sets contain unit-clauses (Lemma 4.6.15), which must be created by clause-shortening, one of C_1, C_2 and one of D_1, D_2 are binary (recall that $F \setminus \{C_1, C_2, D_1, D_2\}$ has no unit-clause). W.l.o.g. assume C_1, D_1 are binary. As already shown in Corollary 4.6.24, in an RHO-MU, all variables are singular. F has no singular variable, so in F_0 all singularity is created by the removal of D_1, D_2, and in F_1 all singularity is created by the removal of C_1, C_2. Thus C_2, D_2 must be full clauses.

It remains to show that C_2 is the complement of D_2. For a full clause in an RHO-MU, by Lemma 4.6.23, Part 2 we know that the complement of its literals occur only once. That $F_0, F_1 \in \mathcal{RHO}$ has at least one full clause implies that C_2 and D_2 have the duty of eliminating each others complement, and so we obtain $C_2 = \overline{D_2}$, and indeed $\mathcal{SMU}'_{\delta=2} \subseteq \mathcal{FC}_{\delta=2}$. Furthermore, $C_2 = \overline{D_2}$ implies that every $w \in \mathrm{var}(F) \setminus \{a, b\}$ has degree 4, while a, b have degree 4, and thus every literal occurs exactly twice in F. Therefore, no literal-occurrence can be removed from F without creating a singular variable. That is, all $F \in \mathcal{MU}'_{\delta=2}$ are saturated (the initial saturation did nothing) and so $\mathcal{MU}'_{\delta=2} \subseteq \mathcal{FC}_{\delta=2}$.

We turn to the second part of the proof, showing $\mathcal{FM}_{\delta=2} \subseteq \mathcal{DFM}_{\delta=2}$, that is, the core F' of every $F \in \mathcal{FM}_{\delta=2}$ contains only binary clauses. By the characterisation of FMs (Theorem 5.1.8), F' realises AllEqual over the variables of F. The deficiency of F' is $\delta(F') = \delta(F) - 2 = 0$. Obtain F'' by replacing each $C \in F'$ by a prime implicate $C'' \subseteq C$ of F', where every prime implicate is binary. Now F'' is logically equivalent to F', and we can apply Theorem 5.2.9 to $F'' \mathbin{\dot\cup} \{P, N\}$ (P, N are full positive and negative clauses over $\mathrm{var}(F'')$), obtaining an MSD $G := \mathrm{pdg}(F'' \mathbin{\dot\cup} \{P, N\})$ with $\delta(G) = \delta(F'')$. Due to the functional characterisation of F' we have $\mathrm{var}(F'') = \mathrm{var}(F') = \mathrm{var}(F)$. Using that MSDs have minimal deficiency 0 (Lemma 5.2.11), thus $\delta(G) = 0$, and so by the same lemma G is a cycle of length $n(F)$, and thus $F \mathbin{\dot\cup} \{P, N\}$ is isomorphic to $\mathcal{F}_{n(F)}$ (recall Lemma 5.2.13). Now $\mathcal{F}_{n(F)}$ is saturated (Lemma 5.2.14), and thus indeed $F'' = F'$. □

Now by Theorem 5.3.1 and Lemma 5.2.13 we obtain a new proof of the seminal result of Kleine Büning ([78]).

Corollary 5.3.2 ([78]) *For $F \in \mathcal{MU}'_{\delta=2}$ holds $F \cong \mathcal{F}_{n(F)}$.*

We remark that the approach of [78] is based on splitting $F \in \mathcal{MU}'_{\delta=2}$ on an appropriate variable and analysing the resulting clause-sets. For MUs such a splitting leads to unsatisfiable clause-sets which contain some minimally unsatisfiable sub-formulas. Via a detailed study of the splitting behaviour, they show that splitting instances F_0, F_1 has some MUSs $F'_0, F'_1 \in \mathcal{MU}_{\delta=1}$. Then using so-called "basic matrices" they characterise F'_0, F'_1, and obtain the characterisation of the original clause-set F. Our method (for characterising $F \in \mathcal{MU}'_{\delta=2}$) is based on the Splitting Ansatz (see Section 1.5.2) and the correspondence between DFMs and MSDs. We use splitting on saturated MUS (where the results are always MUs) to show that F is a DFM, up to isomorphism. Then using graph-theoretical reasoning we show that DFMs of deficiency 2 correspond to

MSDs of deficiency 0 which are cycles digraphs, and this yields the characterisation of F.

Now using Corollary 5.3.2, it is easy to see that up to isomorphism the only hitting clause-sets in $\mathcal{MU}'_{\delta=2}$ are $\mathcal{F}_2, \mathcal{F}_3$:

Corollary 5.3.3 ([110]) *There are precisely two elements in $\mathcal{UHIT}_{\delta=2} \cap \mathcal{MU}'$ (up to isomorphism), namely $\mathcal{F}_2$ and $\mathcal{F}_3$.*

We know $\mathrm{nst}(F)$ for $F \in \mathcal{MU}$ is unique (Theorem 4.5.4). So with Corollary 5.3.2, we have confluence modulo isomorphism for F (i.e., all $F' \in \mathrm{sDP}(F)$ are pairwise isomorphic), since $\mathcal{F}_n \cong \mathcal{F}_m$ iff $n = m$. This reveals that each $F \in \mathcal{MU}_{\delta=2}$ contains a unique reason of unsatisfiability, namely a cycle digraph of length $\mathrm{nst}(F)$ over some of its literals (which are not unique in general) together with two clauses stating their non-equivalence.

5.4 Nonsingular 2-MUs

The main goal of this section is to show the confluence modulo isomorphism of singular DP-reduction for 2-MUs (i.e., for $F \in 2\text{–}\mathcal{MU}$ all $F' \in \mathrm{sDP}(F)$ are pairwise isomorphic), and to characterise the nonsingular elements of $2\text{–}\mathcal{MU}$ (i.e., $F \in 2\text{–}\mathcal{MU}'$) using the positive implication digraph (which is enough in this case); while in the next chapter we will use the full implication digraph to characterise all 2-MUs. We start off with non-uniform 2-MUs, and then provide insight for the characterisation of $2\text{–}\mathcal{MU}'$.

5.4.1 The non-uniform cases

Here we study some properties of non-uniform 2-MUs, i.e., 2-MUs with a unit-clause; while in Section 6.3 we will characterise all the isomorphism types of this class.

A well-known fact about 2-MUs with a unit-clause is that they are renamable Horn clause-sets with deficiency 1 ([27, Lemma 7] and [80, Lemma 5.1]), and have at most two unit-clauses ([116, Lemma 8]):

Lemma 5.4.1 *For $F \in 2\text{–}\mathcal{MU}$ with a unit clause holds $F \in \mathcal{MU}_{\delta=1} \cap \mathcal{RHO}$, and has at most two unit-clauses.*

Proof: Unsatisfiable clause-sets in $2\text{–}\mathcal{CLS}$ have a unit-resolution refutation ([90, Lemma 5.6]). By Lemma 4.6.20, for $F \in \mathcal{MU}$ holds: $F \in \mathcal{RHO}$ iff F is refutable by unit-resolution, and so F has deficiency 1 (see the proof of Lemma 4.6.20 for more details). In a 2-MU F with deficiency 1 (i.e., $c(F) = n(F) + 1$) and a unit-clause, the number of literal occurrences in F is less than $2c(F) = 2n(F) + 2$. Recall that in MUs, all variables occur at least once positively and once negatively and so the minimum number of literal occurrences is $2n(F)$. Therefore F can have at most two unit-clauses. $\square$

Regarding the number of literal occurrences in 2-MUs the following upper bound is given in [82, proposition 4]:

Lemma 5.4.2 ([82]) *For $F \in 2\text{-}\mathcal{MU}$, a literal $x \in \mathrm{lit}(F)$ occurs at most twice, i.e., $\mathrm{ld}_F(x) \leq 2$.*

Proof: First consider the case that F has a unit clause. Then by Lemma 5.4.1, F has deficiency 1 (i.e., $c(F) = n(F) + 1$) and there are at most $2c(F) - 1 = 2n(F) + 1$ literals in F. That is, F has at most one literal of degree 2 while every other literal has degree 1.

Now consider the case that F is 2-uniform. Let $v \in \mathrm{var}(F)$ be the variable of x. And let $F' \in \mathcal{MU}$ be a clause-set obtained by locally saturating F on v (Definition 4.1.7). Consider splitting F' on v and let $F_0 := \langle v \to 0 \rangle * F'$ and $F_1 := \langle v \to 1 \rangle * F'$. By Lemma 4.4.4 we have $F_0, F_1 \in \mathcal{MU}$, while by construction holds $F_0, F_1 \in 2\text{–}\mathcal{CLS}$. Since F_0, F_1 have some unit-clauses (which come from the binary clauses in F containing v), by Lemma 5.4.1 we get $F_0, F_1 \in \mathcal{MU}_{\delta=1}$. Now by the first part of this proof, each of F_0, F_1 has at most two unit-clauses. Thus each of v and $\overline{v}$ occurs at most twice in F, since the unit-clauses in F_0 resp. F_1 come precisely from the occurrences of v resp. $\overline{v}$ in F. □

Here we characterise a special case of 2-MUs with a unit-clause while in Section 6.3 we provide full classification of these 2-MUs.

Lemma 5.4.3 *Consider $F = \{\{1\}, \{-1, 2\}, \ldots, \{-(n-1), n\}, \{-n\}\}$ in Lemma 4.6.40. Any $G \in 2\text{-}\mathcal{MU}$ with exactly two unit-clauses is isomorphic to F. Furthermore both unit-clauses can be partially saturated to a full clause (yielding two saturations), and these two full clauses are complementary.*

Proof: Let $n := n(G)$. Since G is uniform except of two unit-clauses, the number of literal occurrences is $2c(G) - 2 = 2n$, and so every literal in G occurs only once and G is marginal (Lemma 4.6.13). By Lemma 4.6.28 we have $G \in \mathcal{RHO}$ and $\delta(G) = 1$. Therefore there exists some $G' \cong \mathrm{S}_n \in \mathcal{SMU}_{\delta=1} \cap \mathcal{RHO}$ where G' is a saturation of G (recall Lemmas 4.6.20 and 4.6.22). Now as S_n has only one unit-clause, there are precisely two possibilities for G' as follows where each has a full monotone clause (recall that by definition of saturation there is a bijection $f : G \to G'$ with $C \subseteq f(C) \in G'$ for all $C \in G$):

1. Either $G' = \mathrm{S}_n \in \mathcal{SMU}_{\delta=1} \cap \mathcal{RHO}$;
2. or $G' = \{\{-n\}, \{n, -(n-1)\}, \ldots, \{n, \ldots, 2, -1\}, \{n, \ldots, 1\}\} \in \mathcal{SMU}_{\delta=1} \cap \mathcal{RHO}$.

□

Before coming to the main results of this section, we comment on a statement from the literature about the shape of 2-MUs:

Example 5.4.4 *In [116, Lemma 19] we find the following general form of 2-MUs F:*

$$F = \{\,\{x, l_1\}, \{-l_1, l_2\}, \ldots, \{-l_m, y\}, \{-y, s_1\}, \ldots, \{-s_m, -y\}\,\} \cup \\ \{\,\{-x, p_1\}, \{-p_1, p_2\}, \ldots, \{-p_m, z\}, \{-z, q_1\}, \ldots, \{-q_m, -z\}\,\}.$$

In general 2-MUs may have up to two unit-clauses, while F does not contain any unit-clause, so we restrict ourselves to the uniform case. F should allow B_n, but F has at least one apparent singular variable, namely x. It follows that the different names might stand for equal variables. But we can not have arbitrary equalities, since every literal in 2-MUs occurs at most twice, and also redundancies can occur. So we see that the above general form of F only shows, that it has a refutation using two input-resolution chains, while the understanding of the possible isomorphism types is a completely different thing.

5.4.2 The uniform cases

We now come to the main results of this section, characterising the nonsingular 2-MUs. We first show that $F \in 2\text{–}\mathcal{MU}'$ can be saturated to a DFM, up to renaming. That is, there exist a positive clause and a negative clause in F which can be partially saturated to full positive and negative clauses. The proof is based on the Splitting Ansatz (Section 1.5.2): we use local saturation on an appropriate variable $v \in \mathrm{var}(F)$ (Definition 4.1.7) to obtain splitting instances $F_0, F_1 \in 2\text{–}\mathcal{MU}$, then we characterise them and lift the information obtained back to F. We show that F_0, F_1 are in $\mathcal{RHO}$ and each has a unit-clause which can be saturated to a full clause. Then we show that these full clauses are complementary and can be lifted to the original F (by adding v resp. $\overline{v}$). So this yields a DFM which is a partial saturation of F. The details are as follows:

Theorem 5.4.5 *Every element of $2\text{–}\mathcal{MU}'$ can be partially saturated to some element of $\mathcal{DFC}$.*

Proof: We show $F \in 2\text{–}\mathcal{MU}'$ contains, up to flipping of signs, exactly one positive and one negative clause, and these can be saturated to full monotone clauses. F has no unit-clause and is 2-uniform (see Lemma 5.4.1). By the upper bound for the literal degree (Lemma 5.4.2), every literal in F has degree 2. Let $F' \in \mathcal{MU}$ be a clause-set obtained from F by locally saturating $v \in \mathrm{var}(F)$. So $F_0 := \langle v \to 0 \rangle * F'$ and $F_1 := \langle v \to 1 \rangle * F'$ are in $2\text{–}\mathcal{MU}$ (Lemma 4.4.4) and each has exactly two unit-clauses (obtained precisely from the clauses in F containing $v, \overline{v}$). So by Lemma 5.4.1 holds $F_0, F_1 \in \mathcal{RHO} \cap \mathcal{MU}_{\delta=1}$. And by Lemma 5.4.3 all variables are 1-singular and in each of F_0, F_1, both unit-clauses can be partially saturated to a full clause. These full clauses can be lifted to the original F (by adding v resp. $\overline{v}$) while maintaining minimal unsatisfiability (if both splitting results are MU, so is the original clause-set; Lemma 4.4.3, Part 1). Now we show that for a full clause in F_0, F_1 adding v or $\overline{v}$ yields a full clause in F, i.e., only v vanished by splitting. All variables in F_0, F_1 are 1-singular, while F has no singular variable. If there would be a variable w in F_0 but not in F_1, then the variable degree of w would be 2 in F, a contradiction. Thus $\mathrm{var}(F_0) \subseteq \mathrm{var}(F_1)$. Similarly we obtain $\mathrm{var}(F_1) \subseteq \mathrm{var}(F_0)$. So $\mathrm{var}(F_0) = \mathrm{var}(F_1) = \mathrm{var}(F) \setminus \{v\}$.

It remains to show that we can lift w.l.o.g. a full positive clause from F_0 and a full negative clause from F_1. Let $C_1, C_2 \in F$ be the clauses containing v and $D_1, D_2 \in F$ be the clauses containing $\overline{v}$. Assume the unit-clause $C_1 \setminus \{v\} \in F_0$ can be saturated to a full positive clause. This implies that every $C \in F \setminus \{C_1\}$ has a negative literal (since $F \setminus \{C_1\}$ is satisfied by setting all variables to false). Then by Lemma 5.4.3 the unit-clause $C_2 \setminus \{v\}$ can be saturated to a full negative clause in F_0. Similarly we obtain that every clause in $F \setminus \{C_2, D_1, D_2\}$ has a positive literal. So F has exactly one positive clause C_1 and all binary clauses in F_0, F_1 are mixed. Since $c(F_1) = n(F_1) + 1 = (n(F) - 1) + 1 = n(F)$ and there are $n(F) - 1$ occurrences of each literal in F_1, w.l.o.g. D_1 is a negative clause and D_2 is mixed. Recall that in $\mathcal{MU}_{\delta=1}$ every two clauses have at most one clash (Corollary 4.6.14), and so $D_1 \setminus \{\overline{v}\} \in F_1$ can be saturated to a full negative clause (otherwise there would be a clause with more than one clash with the full clause). So we obtain a DFM which is a partial saturation of F. □

By [59, Theorem 4], every MSD with at least two vertices has at least two linear vertices (Definition 3.1.5). We need to characterise a special case of MSDs with exactly two linear vertices. This could be derived from the general characterisation by [58, Theorem 7], but proving it directly is useful and not harder than to derive it:

Lemma 5.4.6 *An MSD G with exactly two linear vertices, where every other vertex has in-degree and out-degree both at least 2, is a dipath.*

Proof: We show that G is a dipath by induction on $n := |V(G)|$. For $n = 2$ it is clear that G is MSD iff G is a dipath. So assume $n \geq 3$. Consider one linear vertex $v \in V(G)$ with arcs (w, v) and (v, w'), where $w, w' \in V(G)$. If $w \neq w'$ would be the case, then the MSD obtained by removing v and adding the arc (w, w') had only one linear vertex (since the in-degree and out-degree of other vertices are unchanged). So this case is not possible and we have $w = w'$. Let G' be the MSD obtained by removing v. Now w is a linear vertex in G' (since every MSD has at least two linear vertices). By induction hypothesis G' is a dipath of length $|V(G')|$. The assertion follows now immediately by choosing a linear vertex $u \in V(G')$ and adding a new vertex v with arcs (u, v) and (v, u). □

By definition, for a mixed binary clause-set F, a singular variable of degree 2 (occurring exactly once positively and once negatively) is a linear vertex in the positive implication digraph $\mathrm{pdg}(F)$. So by Theorem 5.2.9, a variable v in a DFM F has degree 4 (i.e., degree 2 in the core) iff v is a linear vertex in $\mathrm{pdg}(F)$.

Theorem 5.4.7 *$F \in \mathcal{DFC}$ can be partially marginalised to some nonsingular element of 2–$\mathcal{MU}$ if and only if $F \cong \mathrm{DB}_{n(F)}$.*

Proof: Since by Lemma 5.2.19 $\mathcal{B}_n$ is a marginalisation of DB_n (obviously then the unique nonsingular one), it remains to show that a DFM F, which can be partially marginalised as in the assertion, is isomorphic to $\mathrm{DB}_{n(F)}$. We show that $\mathrm{pdg}(F)$ has exactly two linear vertices, while all other vertices have in-degree and out-degree at least two, which proves the statement by Theorem 5.2.9 and Lemma 5.4.6. Consider a nonsingular $G \in$ 2–$\mathcal{MU}$ obtained by marginalisation of F. Recall that by Lemma 5.2.10 the mixed clauses are untouched. $\mathrm{pdg}(F)$ has at least two linear vertices, so the mixed clauses in G have at least two 1-singular variables. Indeed the core of F has exactly two 1-singular variables, since these variables must occur in the positive and negative clauses of G, which are of length two. The other vertices have in-degree/out-degree at least two due to nonsingularity. $\square$

By Theorems 5.4.5, 5.4.7 we obtain a new proof for the characterisation of nonsingular 2-MUs:

Corollary 5.4.8 ([82]) *For $F \in$ 2–$\mathcal{MU}'$ with $\delta(F) \geq 2$ holds $F \cong \mathcal{B}_{n(F)}$.*

We remark that the approach of [82] is based on splitting $F \in$ 2–$\mathcal{MU}'$ on a variable and characterising some minimally unsatisfiable sub-formulas of the resulting clause-sets. Since in general the splitting instance F_0, F_1 are not MU, their approach is to remove clauses appropriately in one of splitting instances F_0 in order to obtain an MU $F_0' \subset F_0$. They show that such F_0' has deficiency 1, and they characterise its isomorphism type. Also via induction on the number of variables in F, they show that for any clause $C \in F$ we have $\overline{C} \in F$, which then yields the isomorphism type of F (using the isomorphism type of F_0'). Our method is based on the Splitting Ansatz (Section 1.5.2) and the correspondence between DFMs and MSDs. Since 2–$\mathcal{MU}'$ is not stable under saturation, we use local saturation which only saturates the variable we want to split on. So local saturation uses all clauses, and we obtain 2-MUs $F_0, F_1 \in \mathcal{MU}_{\delta=1}$. Then we connect 2–$\mathcal{MU}'$ to the new class $\mathcal{DFM}$ and use the connection between DFMs and digraphs to obtain the isomorphism types of $F \in$ 2–$\mathcal{MU}'$.

Regarding the number of clauses for $F \in$ 2–$\mathcal{MU}$, in [116, Lemma 19] the upper bound $c(F) \leq 4n(F)$ is given, while a sharper bound $c(F) \leq 4n(F) - 2$ is shown in [112, Proposition 1] (which is also far from being sharp). Here we present the sharp bound for the number of clauses in 2-MUs, attained exactly for the $\mathcal{B}_n$:

Corollary 5.4.9 *For $F \in$ 2–$\mathcal{MU}$, $F \neq \{\bot\}$ holds $c(F) \leq 2n(F)$, where we have equality iff $F \cong \mathcal{B}_{n(F)}$.*

Proof: First we show the upper bound. F with a unit-clause is in $\mathcal{MU}_{\delta=1}$ and so $c(F) = n(F) + 1$ (Lemma 5.4.1). Otherwise, by Lemma 5.4.2, the number of literals is at most $4n(F)$ and so the number of binary clauses is at most $4n(F)/2 = 2n(F)$.

Now turning to the characterisation of equality, for $F \cong \mathcal{B}_{n(F)}$ we have equality. In the reverse direction, $n(F) \geq 2$ implies $\delta(F) \geq 2$, and so F is 2-uniform (Lemma 5.4.1). The upper bound for the literal degree (Lemma 5.4.2) yields that every literal is of degree 2, i.e., $F \in \mathcal{MU}'$ and so $F \cong \mathcal{B}_{n(F)}$. $\square$

Corollary 5.4.8 together with Theorem 4.5.4 imply that singular DP-reduction for $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ with $k \geq 2$ is confluent modulo isomorphism to some $F' \cong \mathcal{B}_k$. This implies that F has a unique reason of unsatisfiability, namely the presence of a dipath of length $\mathrm{nst}(F) = k$ together with two binary monotone clauses containing variables of the endpoints of the dipath.

Example 5.4.10 *Singular DP-reduction is not confluent in general for $F \in 2\text{–}\mathcal{MU}$: Consider $\mathrm{DB}_2 = A_2$. Let F be the clause-set obtained by a singular 2-extension of DB_2, for clauses $\{1,2\},\{1,-2\}$ and a new variable 3 as follows: $F := \{\{1,-3\},\{2,3\},\{-2,3\},\{-1,2\},\{-1,-2\}\}$, where $\mathrm{var}_s(F) = \{1,3\}$. Let $F' := \mathrm{DP}_1(F) \in \mathcal{MU}$. So we get $F' = \{\{2,-3\},\{2,3\},\{-2,3\},\{-2,-3\}\}$, where $F' \cong \mathrm{DB}_2$ but $F' \neq \mathrm{DB}_2$ (so $|\mathrm{sDP}(F)| > 1$).*

Finally since the $\mathcal{B}_n$ are marginal (Section 5.2.2), by Corollary 5.4.8 we immediately obtain:

Corollary 5.4.11 *Every element of $2\text{–}\mathcal{MU}'$ is marginal.*

By contrast every $F \in \mathcal{MU}'_{\delta=2}$ is saturated (Corollary 5.3.2 and Lemma 5.2.14). So we see a potentially interesting duality about the basic classes: every nonsingular MU with deficiency 2 is automatically saturated, while every nonsingular MU with clauses of length at most 2 is automatically marginal.

Chapter 6

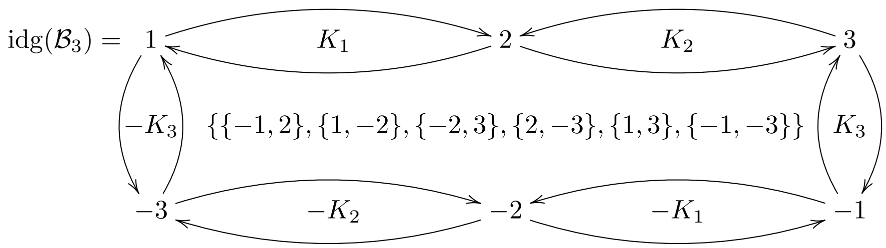
Classification of minimally unsatisfiable 2-CNFs

In Section 5.4 we characterised nonsingular 2-MUs via their positive implication digraph, while in this chapter we provide a full classification all 2-MUs using their (full) implication digraph and give a very precise overview.

After explaining the running example of this chapter in Section 6.1, the important notion of implication digraphs of 2-CNFs are discussed in Section 6.2. Then in Section 6.3 we study 2-MUs of deficiency 1 and characterise their isomorphism types. In Section 6.4 we explain singular DP-reduction in general, including the specialisation 1-singular DP-reduction. We define digraphs WDCs in Section 6.5, and then we characterise their isomorphism types. Section 6.6 provides classification of 2-MUs of deficiency $k \geq 2$. We first provide a generation process for the elements of $2\text{–}\mathcal{MU}_{\delta=k}$, showing that their implication digraphs are $2k$-WDCs. Then we show that for 2-MUs F, F' the set of isomorphisms between F, F' is equal to the set of isomorphisms between their implication digraphs. Finally we obtain a variety of applications in Section 6.7, including a polytime isomorphism decision for 2-MUs, and a bound for the number of their isomorphism types.

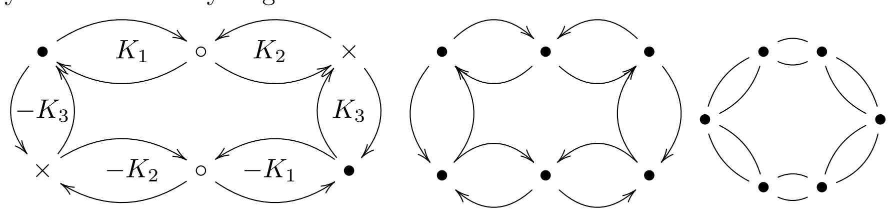
6.1 The running example

The running example of this chapter is based on $\mathcal{B}_3 \in 2\text{–}\mathcal{MU}_{\delta=3}$ (Definition 5.2.18), which has the following implication digraph (3 variables, thus 6 vertices, and 6 clauses, thus $2 \cdot 6 = 12$ arcs):

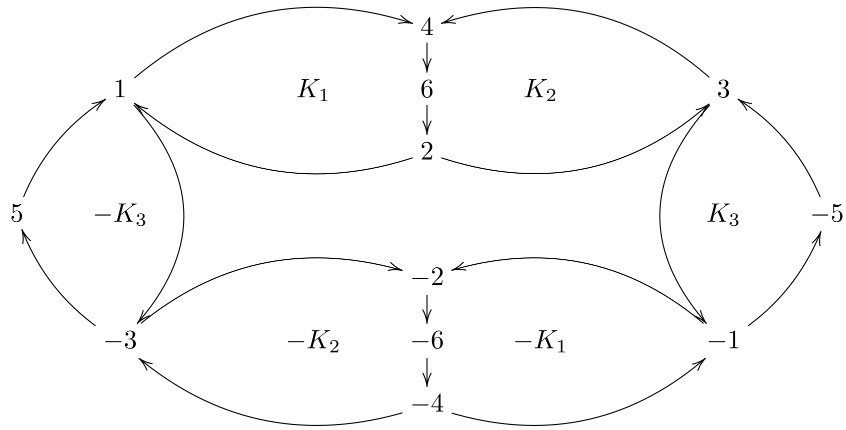


idg($\mathcal{B}_3$) is a double 6-cycle and so has six cycles of length two, the cycles K_1, K_2, K_3, and their contrapositions $-K_1, -K_2, -K_3$. The contraposition of an arc (x, y) is the arc $\overline{(x, y)} := (\overline{y}, \overline{x})$ (we do not use the notation $\overline{K_i}$ here for typographical reasons). We note here, that the contraposition of each *small cycle* is its "antipodal" cycle, on the "opposite side" of the digraph. idg($\mathcal{B}_3$) has also two *big cycles*, namely $K_4 : 1 \to 2 \to 3 \to -1 \to 2 \to -3 \to 1$ and its contraposition $-K_4$, and these two cycles are exactly the contradictory cycles. In general the implication digraph of $\mathcal{B}_n$ is a double $2n$-cycle (see Section 3.3) with $2n$ small cycles (non-contradictory), and two big cycles (contradictory), so that together idg($\mathcal{B}_n$) has exactly $2n + 2$ cycles (a cycle of a digraph is always directed).

To display the unlabelled idg($\mathcal{B}_3$) with complementation, the three pairs of complementary literals (this is all what is needed to know about complementation) in idg($\mathcal{B}_3$) are shown below by three different types $\bullet, \circ, \times$ of vertices (note their antipodal positions). Furthermore we show the abstract implication digraph (the unlabelled idg($\mathcal{B}_3$)), which has lost the information on the complementation. The final abstraction for a 2-MU F is the homeomorphism type of the implication graph (undirected) of F, which here is a cycle of 6 (small) cycles connected by single vertices:



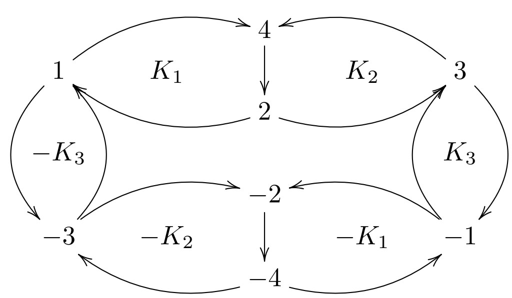
The following implication digraph, which is a WDC, is our running example, obtained from $\mathrm{idg}(\mathcal{B}_3)$ by splitting some vertices and arcs:

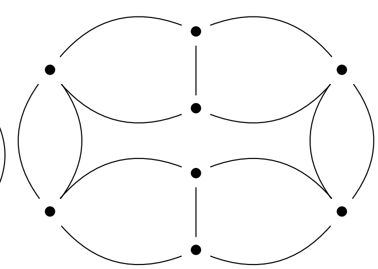


As mentioned before, the implication digraph together with complementation of vertices is essentially the same as the original clause-set. So the underlying clause-set of the above implication digraph is

$$F = \{\{-5,1\},\{-1,4\},\{-4,6\},\{-6,2\},\{-2,3\},\{-3,-1\},\{1,-2\},\\ \{4,-3\},\{3,5\}\} \in 2\text{–}\mathcal{MU}.$$

$\mathrm{idg}(F)$ has six small cycles K_1, K_2, K_3 and their contraposition $-K_1, -K_2, -K_3$, as in $\mathrm{idg}(\mathcal{B}_3)$. Furthermore, $\mathrm{idg}(F)$ has four linear vertices, namely $5, -5, 6, -6$. In order to understand better the structure of F, we consider its non-1-singular normalform, denoted by $1\mathrm{sDP}(F)$, with its implication digraph obtained by removing all the linear vertices. Together with the homeomorphism type of F these graphs are:





$$1\mathrm{sDP}(F) = \{\{1,3\},\{-1,4\},\{2,-4\},\{-2,3\},\\ \{-1,-3\},\{1,-2\},\{-3,4\}\} \in 2\text{–}\mathcal{MU}.$$

6.2 Implication digraphs of 2-CNFs

In Chapter 5 we introduced the positive implication digraph (Definition 5.2.3) which sufficed to characterise the nonsingular elements of $2\text{–}\mathcal{MU}$, while here we need the (full) implication digraph to characterise all 2-MUs. Since in this chapter often the empty clause $\bot$ is just in the way, by an upper-index "$*$" we exclude it: $\boldsymbol{2\text{–}\mathcal{CLS}^*} := \{F \in 2\text{–}\mathcal{CLS} : \bot \notin F\}$, and $\boldsymbol{2\text{–}\mathcal{MU}^*_{\delta=1}} := 2\text{–}\mathcal{MU}_{\delta=1} \setminus \{\{\bot\}\}$.

Definition 6.2.1 *For $F \in 2\text{–}\mathcal{CLS}^*$ the **implication digraph** $\mathbf{idg}(\boldsymbol{F})$ is defined as follows:*

- $V(\mathrm{idg}(F)) := \mathrm{lit}(F)$,
- $E(\mathrm{idg}(F)) := \{(\overline{x}, x) | \{x\} \in F\} \cup \{(\overline{x}, y), (\overline{y}, x) | \{x, y\} \in F \text{ and } x \neq y\}$.

*The **implication graph** is $\mathbf{ig}(\boldsymbol{F}) := \mathrm{ug}(\mathrm{idg}(F))$.*

For a literal $x \in \mathrm{lit}(F)$ its degree $\mathrm{ld}_F(x)$ is the in-degree of vertex x in $\mathrm{idg}(F)$, and the out-degree of vertex $\overline{x}$, while for a variable $v \in \mathrm{var}(F)$ its degree $\mathrm{vd}_F(v)$ is the degree of vertex v as well as the degree of vertex $\overline{v}$ in $\mathrm{idg}(F)$. The arc $\overline{y} \to x$ is the **contraposition** of the arc $\overline{x} \to y$. In an implication digraph, a cycle with two clashing literals (i.e., a literal and its complement) is called **contradictory**. The fundamental property, first observed in [8], is that a clause-set $F \in 2\text{–}\mathcal{CLS}^*$ is unsatisfiable iff there exists a contradictory cycle in $\mathrm{idg}(F)$. By forgetting complementation and translating clauses into arcs we have:

Lemma 6.2.2 *For $F_1, F_2 \in 2\text{–}\mathcal{CLS}^*$ holds: if $F_1 \cong F_2$ then $\mathrm{idg}(F_1) \cong \mathrm{idg}(F_2)$. More precise, if $f : F_1 \to F_2$ is an isomorphism, then also $f : \mathrm{idg}(F_1) \to \mathrm{idg}(F_2)$ is an isomorphism.*

The reverse direction of Lemma 6.2.2 does not hold in general, and so the isomorphism type of implication digraphs is not a "complete isomorphism invariant" for 2-CNFs, as the following example shows:

Example 6.2.3 *Consider any digraph G which is the disjoint union of two (directed) cycles, and assume that G is the implication digraph of some 2-CNFs. If the cycles have* different *lengths, then they can not be the contraposition of each other, and so each cycle must be a contradictory cycle (as for every literal its complement must be in the same cycle). That is, every $F \in 2\text{–}\mathcal{CLS}^*$ with $\mathrm{idg}(F) \cong G$ is unsatisfiable.*

Now assume that the cycles have equal *length. So we have two possibilities, namely that the cycles are the contraposition of each other, or they both are contradictory. The first case corresponds to a satisfiable 2-CNF, while the second case yields an unsatisfiable 2-CNF as before, e.g., consider*

$$\begin{aligned} F &= \{\{-1, 2\}, \{-2, 3\}, \{-3, 4\}, \{1, -4\}\}, \\ F' &= \{\{-1\}, \{1, 2\}, \{-2\}, \{-3\}, \{3, 4\}, \{-4\}\}. \end{aligned}$$

The implication digraphs are

$$\mathrm{idg}(F) = \begin{tikzcd} 1 \arrow[r] & 2 \arrow[d] \\ 4 \arrow[u] & 3 \arrow[l] \end{tikzcd} \quad \begin{tikzcd} -1 \arrow[r] & -4 \arrow[d] \\ -2 \arrow[u] & -3 \arrow[l] \end{tikzcd} \quad \mathrm{idg}(F') = \begin{tikzcd} 1 \arrow[r] & -1 \arrow[d] \\ -2 \arrow[u] & 2 \arrow[l] \end{tikzcd} \quad \begin{tikzcd} 3 \arrow[r] & -3 \arrow[d] \\ -4 \arrow[u] & 4 \arrow[l] \end{tikzcd}$$

$\mathrm{idg}(F')$ *has two contradictory cycles, and so* F' *is unsatisfiable, while* F *is satisfiable. Therefore* $F \not\cong F'$*, while* $\mathrm{idg}(F) \cong \mathrm{idg}(F')$.

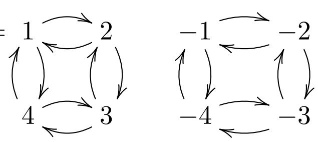
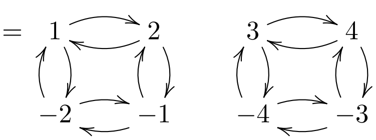
Now consider any implication digraph with precisely two components, each isomorphic to $\mathrm{dg}(\mathrm{CG}_4)$ *(recall Example 3.3.1). One possibility is that the two components are contraposition of each other, which yields a satisfiable 2-CNF, e.g.,*

$$F = \{\{-1, 2\}, \{1, -2\}, \{-2, 3\}, \{2, -3\}, \{-3, 4\}, \{3, -4\}, \{-1, 4\}, \{1, -4\}\}.$$

Another possibility is that each of the components has a contradictory cycle, and so this case corresponds to an unsatisfiable 2-CNF, e.g.,

$$F' = \{\{-1, 2\}, \{1, -2\}, \{1, 2\}, \{-1, -2\}, \{-3, 4\}, \{3, -4\}, \{3, 4\}, \{-3, -4\}\}.$$

The implication digraphs of F, F' *are as follows, where we see* $\mathrm{idg}(F) \cong \mathrm{idg}(F')$ *but* $F \not\cong F'$.

$\mathrm{idg}(F) =$  $\mathrm{idg}(F') =$ 

When adding a notion of complementation to digraphs, then we obtain basically the same as 2-CNFs, as we now explain:

Definition 6.2.4 ([64]) *A* ***skew-symmetry*** *of a digraph* G *is a bijection* $\sigma : V(G) \to V(G)$ *with the following properties:*

1. σ *is its own inverse, i.e.,* $\forall\, v \in V(G) : \sigma(\sigma(v)) = v$ *(i.e.,* σ *is an involution);*

2. *for every vertex* $v \in V(G)$ *we have* $\sigma(v) \neq v$ *(i.e.,* σ *has no fixed-point);*

3. *for every arc* $(a, b) \in E(G)$ *holds* $\sigma(a, b) := (\sigma(b), \sigma(a)) \in E(G)$*, and the induced map* $\sigma : E(G) \to E(G)$ *is a bijection (i.e., transposition by* σ *is an automorphism of* G*).*

A digraph G *is called* ***skew-symmetric****, if there exists a skew-symmetry for* G*. A* ***digraph with skew-symmetry*** *is a pair* (G, σ)*, where* G *is a digraph* G *and* σ *is a skew-symmetry of* G*.*

For a digraph G, the transposed digraph, obtained by reversing the direction of all arcs, is denoted by $\boldsymbol{G^{\mathrm{t}}}$. A skew-symmetry for G is an isomorphism $f : G \to G^{\mathrm{t}}$, where f as a map (from $V(G)$ to itself) is an involution and fixed-point free. For $F \in 2\text{–}\mathcal{CLS}^*$ the digraph $\mathrm{idg}(F)$ has a natural skew-symmetry, namely the complementation of literals, and the corresponding digraph with skew-symmetry is denoted by $\mathbf{sidg}(\boldsymbol{F}) := (\mathrm{idg}(F), (\overline{x})_{x \in \mathrm{lit}(F)})$.

An isomorphism from (G_1, σ_1) to (G_2, σ_2) is a digraph-isomorphism $f : G_1 \to G_2$ which is compatible with the skew-symmetries, i.e., for all $v \in$

$V(G_1)$ holds $\sigma_2(f(v)) = f(\sigma_1(v))$. For all $F, F' \in 2\text{–}\mathcal{CLS}^*$ holds $F \cong F'$ iff $\mathrm{sidg}(F) \cong \mathrm{sidg}(F')$. For any digraph with skew-symmetry (G, σ) we can assume w.l.o.g. that $V(G)$ is a set of literals closed under complementation, and that $\sigma(x) = \overline{x}$ for all $x \in V(G)$ holds, and then there is a unique $F \in 2\text{–}\mathcal{CLS}^*$ with $\mathrm{sidg}(F) = (G, \sigma)$. So in this sense digraphs with skew-symmetry are "the same" of 2-CNFs (however without variables, but just based on literals and their complements), and we can call skew-symmetries of digraphs just "complementations". A digraph may have no complementation (e.g., digraphs with an odd number of vertices) or multiple complementations.

Example 6.2.5 *For an example of a digraph with multiple complementations, we continue Example 6.2.3, by considering the underlying unlabelled digraph G consisting of two disjoint cycles $v_1 \to \ldots \to v_4 \to v_1$ and $w_1 \to \ldots \to w_4 \to w_1$ of length four, i.e.,*

$$G := (\{v_1, \ldots, v_4, w_1, \ldots, w_4\}, \{(v_1, v_2), \ldots, (v_4, v_1), (w_1, w_2), \ldots, (w_4, w_1)\}).$$

We have seen two complementations, given by $G \cong \mathrm{idg}(F)$ and $G \cong \mathrm{idg}(F')$. Now what are all complementations?

There are four complementations yielding a digraph with skew-symmetry isomorphic to $\mathrm{sidg}(F)$, namely one can choose $\overline{v_1} = w_i$ for any i, and then the other complementations are determined. And there are $2 \cdot 2 = 4$ complementations yielding $\mathrm{sidg}(F')$, namely one can say $\overline{v_1} = v_2$ or $\overline{v_2} = v_3$ for the first cycle (which determines the complementations in this cycle), and the same for the second cycle. Altogether G has exactly $4 + 4 = 8$ complementations, which yield exactly two isomorphism-types of digraphs with skew-symmetry.

An arc (x, y) is mapped by complementation to itself, i.e., $(\overline{y}, \overline{x}) = (x, y)$, iff $x = \overline{y}$, iff the arc corresponds to the unit-clause $\{y\}$.

6.3 2-MUs of deficiency one

In this section we characterise of the isomorphism types of $F \in 2\text{–}\mathcal{MU}_{\delta=1}$. By Lemma 4.6.1 1-singular DP-reduction applied to any MU F results in $\{\bot\}$ iff $\delta(F) = 1$. So we can generate (exactly) all of $2\text{–}\mathcal{MU}_{\delta=1}$ by inverse 1-singular DP-reduction. That is, we start with the empty clause, and repeatedly replace a single clause C already generated by two clauses $C' \cup \{v\}, C'' \cup \{\overline{v}\}$ for $C' \cup C'' = C$, $|C'|, |C''| \leq 1$, and a new variable v (a special case of the creation process in Theorem 4.6.2). The clause-sets generated this way, starting with $\{\{\bot\}\}$, together exactly yield $2\text{–}\mathcal{MU}_{\delta=1}$.

Here we consider generating the elements of $2\text{–}\mathcal{MU}^*_{\delta=1}$ (without $\{\bot\}$), and so the starting point are the 2-MUs with precisely one variable, namely $\{\{v\}, \{\overline{v}\}\}$. We need indeed not to create all of $2\text{–}\mathcal{MU}^*_{\delta=1}$, but only up to isomorphism. To make a start, we have w.l.o.g. the following cases, for $|C| = 1, 2$:

(i) $C = \{x\}$:

A: $C' = \{x, v\}$, $C'' = \{\overline{v}\}$.

B: $C'' = \{x, v\}$, $C'' = \{x, \overline{v}\}$.

(ii) $C = \{x, y\}$, $x \neq y$: $C' = \{x, v\}$, $C'' = \{y, \overline{v}\}$.

The above "w.l.o.g." here just means that all additional unit-clauses are negative. We note that there are always at most two unit-clauses (Lemma 5.4.1). We now (further) standardise the process, to minimise the number of case distinctions needed. It is possible to start only with variable $v = 1$, that is, with $\{\{1\}, \{-1\}\}$, and for each new variable to choose the next natural number.

Lemma 6.3.1 *We can generate up to isomorphism the elements of* $2\text{-}\mathcal{MU}^*_{\delta=1}$ *by a sequence of applications of Rules A, B, (ii), with the following restrictions: First at most one application of A, then at most two applications of B, and finally arbitrarily many application of (ii) (if at least one application of Rules (ia) or (ib) took place). If we have one A and at least one B, then as main clause of the first B the new unit-clause is used.*

Proof: Rule (ii) can not be used at the start, and Rule B can be applied at most twice. Indeed the generation process can be restricted w.l.o.g. to have two phases, where Rules A, B are only used in the first phase, and Rule (ii) only in the second phase. For each rule, the clause we choose (C above) is the main clause, while the side clauses are the replacement clauses (C', C'' above). If Rule (ii) is followed by Rule A or B, then we can swap the applications, as the side clauses for Rule (ii) are binary and thus disjoint with the main clause for Rule A or B. Also if Rule B is followed by Rule A, then because of disjointness of the side clauses of B and the main clause of A we can swap the rules. So we can assume that a generation process has first applications of Rule A, then at most two applications of B, and then applications of (ii).

Furthermore, two consecutive applications of Rule A can be replaced by one application of Rule A, followed by one application of Rule (ii); this is shown by considering the very first applications, w.l.o.g. first applied to $\{1\}$, then to $\{-2\}$, yielding

$$\{\{1\}, \{-1\}\} \rightsquigarrow \{\{1, 2\}, \{-2\}, \{-1\}\} \rightsquigarrow \{\{1, 2\}, \{-2, 3\}, \{-3\}, \{-1\}\}.$$

The simulation is

$$\{\{1\}, \{-1\}\} \rightsquigarrow \{\{1, 3\}, \{-3\}, \{-1\}\} \rightsquigarrow \{\{1, 2\}, \{-2, 3\}, \{-3\}, \{-1\}\}.$$

The last point in this standardisation process is to consider exactly one application of A, followed by at least one application of B. Here it does not matter, whether the first application of B uses as main clause the original clause or the new unit-clause produced by A, while we note that the unit-clause for a second application of B is unique. The reason is that both clause-sets are isomorphic: the first case yields

$$\{\{1\}, \{-1\}\} \rightsquigarrow \{\{1, 2\}, \{-2\}, \{-1\}\} \rightsquigarrow \{\{1, 2\}, \{-2\}, \{-1, 3\}, \{-1, -3\}\},$$

the second case yields

$$\{\{1,2\},\{-2\},\{-1\}\} \rightsquigarrow \{\{1,2\},\{-2,3\},\{-2,-3\},\{-1\}\},$$

and the isomorphism swaps variables 1 and 2. $\square$

Concerning Rule (ii), it is easy to see that it produces just a chain as follows:

Lemma 6.3.2 *Applying Rule (ii) $n \geq 1$ times to $\{x,y\}$ yields a clause-set isomorphic to $\{x,1\},\{-1,2\},\dots,\{-(n-1),n\},\{-n,y\}$.*

Important to note here that when there are several binary clauses to start with, the applications of Rule (ii) do not interfere, and so for each of the starting binary clauses we can apply Lemma 6.3.2, with the new variables made disjoint.

We are now ready to derive the basic types of $F \in 2\text{-}\mathcal{MU}^*_{\delta=1}$, according to the number of applications of Rules A, B (while Rule (ii) is applied arbitrarily often). By Lemma 6.3.1 we have:

Corollary 6.3.3 *The five starting points for the applications of Lemma 6.3.2 (and Rule (ii)) are as follows, showing the sequence of applications of Rules A, B, and after the colon the number of unit-clauses:*

(A) $\{\{1,2\},\{-2\},\{-1\}\}$ *: 2.*

(B) $\{\{1,2\},\{1,-2\},\{-1\}\}$ *: 1.*

(AB) $\{\{1,2\},\{-2,3\},\{-2,-3\},\{-1\}\}$ *: 1.*

(BB) $\{\{1,2\}.\{1,-2\},\{-1,3\},\{-1,-3\}\}$ *: 0.*

(ABB) $\{\{1,2\},\{-2,3\},\{-2,-3\},\{-1,4\},\{-1,-4\}\}$ *: 0.*

Definition 6.3.4 *Let $M := \{\{-1,2\},\dots,\{-(n-1),n\}\}$ for $n \in \mathbb{N}$ (and with $n-1$ clauses) be the invariant "middle part" (compare Lemma 6.3.2). We define the following clause-sets where with "U" denotes "unit":*

1. $\mathbf{U^2_{\boldsymbol{n}}} := M \cup \{\{1\},\{-n\}\}$ *for $n \geq 1$ (occurs in [27] and [80]).*
2. $\mathbf{U^1_{\boldsymbol{n,i}}} := M \cup \{\{1\},\{-n,-i\}\}$ *for $n \geq 2$, $1 \leq i \leq n-1$ (introduced in [27]).*
3. $\mathbf{U^0_{\boldsymbol{n,i}}} := M \cup \{\{1,i\},\{-n,-i\}\}$ *for $n \geq 3$, $2 \leq i \leq \frac{n+1}{2}$ (occurred in [35]).*
4. $\mathbf{U^0_{\boldsymbol{n,x,y}}} = M \cup \{\{1,x\},\{-n,-y\}\}$ *for $n \geq 4$, $2 \leq x < y \leq n-1$, $x+y \leq n+1$.*

We now characterise the isomorphism types of $F \in 2\text{-}\mathcal{MU}^*_{\delta=1}$. First consider 2-MUs with two unit-clauses:

Lemma 6.3.5 *For $F \in 2\text{-}\mathcal{MU}_{\delta=1}$ holds $F \cong \mathrm{U}^2_{n(F)}$ iff F has two unit-clauses.*

Proof: The only starting point in Corollary 6.3.3 with two unit-clauses is case (A). Replacing $\{1, 2\}$ according to Lemma 6.3.2, using new variables $3, \dots, n$, yields the clauses $\{1, 2\}, \{-2, 3\}, \dots, \{-(n-1), n\}, \{-n, 2\}, \{-2\}, \{-1\}$. For better formatting we swap variables n and 2, and flip variable 1. This yields U^2_n (Definition 6.3.4), which makes sense for $n \geq 1$, and thus covers all cases with exactly two unit-clauses. □

The implication digraph of U^2_n is a cycle digraph with $2n$ vertices and $2n$ edges (where all vertices have degree 2). The labelled digraph, actually a digraph with skew-symmetry, is shown as follows (see Lemma 6.6.7 for more details). Here arcs from unit-clauses are drawn as double-arcs (if multigraphs would be used, then unit-clauses indeed would yield two parallel arcs):

$$\mathrm{idg}(\mathrm{U}^2_n) = \begin{tikzcd}
1 \arrow[r] & 2 \arrow[r] & \cdots \arrow[r] & n-1 \arrow[r] & n \arrow[d, Rightarrow] \\
-1 \arrow[u, Rightarrow] & -2 \arrow[l] & \cdots \arrow[l] & -(n-1) \arrow[l] & -n \arrow[l]
\end{tikzcd}$$

We now characterise 2-MUs with precisely one unit-clause. First we note that we can merge chains based on two binary clauses $\{x, z\}, \{y, \overline{z}\}$, using z to connect the chains:

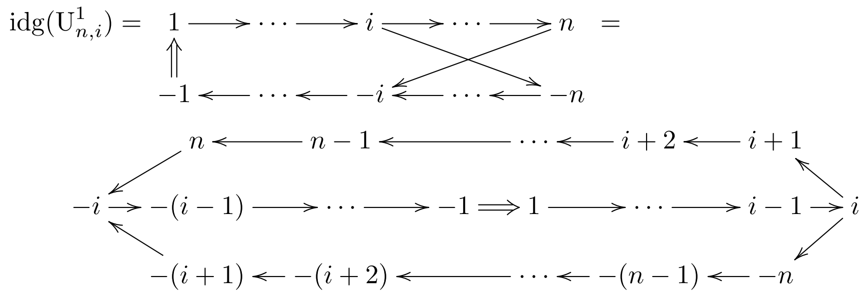
Lemma 6.3.6 *Applying Rule (ii) $n \geq 1$ times to $\{\{x, z\}, \{y, \overline{z}\}\}$ yields a clause-set isomorphic to $\{x, 1\}, \{-1, 2\}, \dots, \{-(n-1), n\}, \{-n, y\}$.*

Lemma 6.3.7 *For $F \in 2\text{-}\mathcal{MU}_{\delta=1}$ holds $F \cong \mathrm{U}^1_{n(F),i}$ for some $1 \leq i < n(F)$ iff F has exactly one unit-clause.*

Proof: In Corollary 6.3.3 cases with precisely one unit-clause are cases (B) and (AB). From Case (B) we obtain $\{\{-1\}, \{1, 2\}, \{-2, 3\}, \dots, \{-(n-1), n\}, \{-n, 1\}\}$ for $n \geq 2$. We note that after flipping literal 1, this is U^2_n, when adding to the last clause the literal -1, i.e., we get $\mathrm{U}^1_{n,1}$ (Definition 6.3.4).

For Case (AB), we rename variable 2 to some x not used as new variable. First from $\{1, x\}$ we obtain either $\{1, x\}$ or $\{1, 2\}, \dots, \{-p, x\}$ for some $p \geq 2$. And from $\{-x, 3\}, \{-x, -3\}$, renamed to $\{-x, p+1\}, \{-x, -(p+1)\}$, we obtain $\{-x, p+1\}, \dots, \{-q, -x\}$ for some $q \geq p+1$. Appending these chains yields with the original $\{-1\}$ a clause-set isomorphic to $\{-1\}, \{1, 2\}, \dots, \{-(n-1), n\}, \{n, -i\}$ for some $2 \leq i < n$ and $n \geq 2$. After flipping literal 1, this is U^2_n, when adding to the last clause the literal $-i$, i.e., we get $\mathrm{U}^1_{n,1}$ (Definition 6.3.4).
□

We note that $\mathrm{U}^2_n = \mathrm{U}^1_{n,n}$ (allowing this degeneration for the moment). The implication digraph of $\mathrm{U}^1_{n,i}$ has $2n$ vertices and $2n+1$ edges, and consists of two cycle digraphs of length $n+i$, which overlap in a path of length $2i - 1 \geq 1$ (we note $2(n+i) - (2i-1) = 2n+1$); two vertices have degree 3, all other vertices have degree 2:



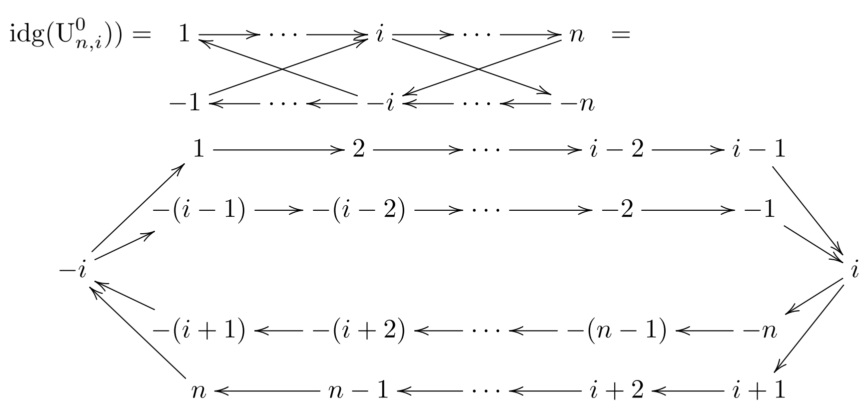
We note here that the digraphs $\mathrm{idg}(\mathrm{U}^1_{n,i})$ have a unique skew-symmetry (we do not prove that here, but the basic fact used is that a skew-symmetry is an "anti-automorphism", and has to pair vertices of identical degree).

Finally we characterise 2-uniform (without unit-clauses) elements of $2\text{–}\mathcal{MU}_{\delta=1}$:

Lemma 6.3.8 *For 2-uniform $F \in 2\text{–}\mathcal{MU}_{\delta=1}$ with $n := n(F)$ holds:*

- *If F has a variable of degree 4 (occurring twice positively and twice negatively), then $n \geq 3$ and $F \cong \mathrm{U}^0_{n,i}$ for some $i \in \{2, \ldots, n-1\}$.*
- *Otherwise $n \geq 4$ and $F \cong \mathrm{U}^0_{n,x,y}$ for some $x, y \in \{2, \ldots, n-1\}$ with $x < y$.*

Proof: The 2-uniform starting points in Corollary 6.3.3 are cases cases (BB) and (ABB). For case (BB), we apply Lemma 6.3.6 twice, and similar to above, we obtain $\mathrm{U}^0_{n,i}$ (Definition 6.3.4), for the moment allowing all $2 \leq i \leq n-1$. The implication digraph of $\mathrm{U}^0_{n,i}$ has $2n$ vertices and $2n+2$ edges, and two vertices have degree 4, while all other vertices have degree 2:



The two paths from $-i$ to i have length i, while the two paths from i to $-i$ have length $n-i+1$. If $i > n-i+1$, then we reverse the direction of all arcs in this digraph, which corresponds to flipping all literals in $\mathrm{U}^0_{n,i}$. So then we obtained an isomorphic clause-set, where the upper two paths are swapped with the lower two paths, and thus w.l.o.g. one can assume $i \le n - i + 1$, that is, $i \le \frac{n+1}{2}$. We note here that the digraphs $\mathrm{idg}(\mathrm{U}^0_{n,i})$) again have a unique skew-symmetry.

Similarly, for case (ABB), in a sense the most general case, we obtain $\mathrm{U}^0_{n,x,y}$ (Definition 6.3.4), allowing for the moment all $2 \le x, y \le n-1$, where $x < y$ (this comes from the chaining-order). Allowing degenerations, we have $\mathrm{U}^2_n = \mathrm{U}^0_{n,1,n}$, $\mathrm{U}^1_{n,i} = \mathrm{U}^0_{n,1,i}$ and $\mathrm{U}^0_{n,i} = \mathrm{U}^0_{n,i,i}$. The implication digraph of $\mathrm{U}^0_{n,x,y}$ has $2n$ vertices and $2n+2$ edges, and four vertices have degree 3, while all other vertices have degree 2:

$$\mathrm{idg}(\mathrm{U}^0_{n,x,y}) = \begin{tikzcd}
1 \arrow[r] & \cdots \arrow[r] & x \arrow[r] & \cdots \arrow[r] & y \arrow[r] \arrow[drr] & \cdots \arrow[r] & n \arrow[dll] \\
-1 \arrow[urr] & \cdots \arrow[l] & -x \arrow[l] \arrow[ull] & \cdots \arrow[l] & -y \arrow[l] & \cdots \arrow[l] & -n \arrow[l]
\end{tikzcd} =$$

$$\begin{tikzcd}
 & 1 \arrow[r] & 2 \arrow[r] & \cdots \arrow[r] & x-2 \arrow[r] & x-1 \arrow[dr] & \\
-x \arrow[ur] \arrow[r] & -(x-1) \arrow[r] & -(x-2) \arrow[r] & \cdots \arrow[r] & -2 \arrow[r] & -1 \arrow[r] & x \arrow[d] \\
-(x+1) \arrow[u] & & & & & & x+1 \arrow[d] \\
\cdots \arrow[u] & & & & & & \cdots \arrow[d] \\
-(y-1) \arrow[u] & & & & & & y-1 \arrow[d] \\
-y \arrow[u] & -(y+1) \arrow[l] & -(y+2) \arrow[l] & \cdots \arrow[l] & -(n-1) \arrow[l] & -n \arrow[l] & y \arrow[l] \arrow[dl] \\
 & n \arrow[ul] & n-1 \arrow[l] & \cdots \arrow[l] & y+2 \arrow[l] & y+1 \arrow[l] &
\end{tikzcd}$$

The two paths from $-x$ to x have length x, the two paths from y to $-y$ have length $n-y+1$. As above, w.l.o.g. we can assume $x \le n-y+1$, i.e., $x+y \le n+1$. □

Example 6.3.9 *The "snakes" clause-sets introduced in [35] are isomorphic to* $\mathrm{U}^0_{n,\frac{n+1}{2}} \in 2\text{–}\mathcal{MU}_{\delta=1}$ *for odd* n.

Also "bicycle" 2-CNFs in [35] are a more general form of $\mathrm{U}^0_{n,x,y}$*, namely that a bicycle can be obtained from* U^2_n *for some* $n \ge 2$ *by choosing literals* $x, y \in \mathrm{lit}(\{2, \dots, n-1\})$*, and adding literal* x *to the unit-clause* $\{1\}$ *and adding literal* y *to the unit-clause* $\{-n\}$*. So these bicycle 2-CNFs in general might be satisfiable or unsatisfiable.*

Altogether we achieved the classification of $2\text{–}\mathcal{MU}^*_{\delta=1}$:

Theorem 6.3.10 *For input $F \in 2\text{-}\mathcal{MU}^*_{\delta=1}$ exactly one of the four cases in Lemmas 6.3.5, 6.3.7, and 6.3.8 applies. Let $u(F) \in \{0,1,2\}$ be the number of unit-clauses in F. Then in polynomial time the unique parameter-list $L(F)$ of length $0,1,2$, according to the applicable case can be computed, such that $F \cong \mathrm{U}^{u(F)}_{n(F),L(F)}$ holds. This map* $\mathrm{canon} : 2\text{-}\mathcal{MU}^*_{\delta=1} \to 2\text{-}\mathcal{MU}^*_{\delta=1}$ *given by* $\mathrm{canon}(F) := \mathrm{U}^{u(F)}_{n(F),L(F)}$, *is a polytime computable clause-set-canonisation, that is, for $F, F' \in 2\text{-}\mathcal{MU}^*_{\delta=1}$ holds $F \cong F'$ iff* $\mathrm{canon}(F) = \mathrm{canon}(F')$.

Furthermore the map $F \in 2\text{-}\mathcal{MU}^*_{\delta=1} \mapsto \mathrm{canon}'(F) := \mathrm{ig}(\mathrm{canon}(F))$ to the class of graphs is a polytime computable graph-canonisation, that is, for $F, F' \in 2\text{-}\mathcal{MU}^*_{\delta=1}$ holds $F \cong F'$ iff $\mathrm{canon}'(F) = \mathrm{canon}'(F')$. From $\mathrm{canon}'(F)$ in polytime F can be reconstructed up to isomorphism.

Proof: A contraction of two arcs into one edge, when transitioning from the implication digraph $\mathrm{idg}(F)$ to the implication graph $\mathrm{ig}(F)$, happens exactly for the two following cases:

1. For complementary unit-clauses in F, $\mathrm{idg}(F)$ is the cycle digraph of length 2, while $\mathrm{ig}(F)$ is the complete graph with two vertices.

2. Equivalence-clauses $\{x, y\}, \{\overline{x}, \overline{y}\} \in F$ yield a cycle digraph of length 2 in $\mathrm{idg}(F)$, and a single edge in $\mathrm{ig}(F)$.

We see from the implication digraphs, that here only the first case happens, i.e., when $n(F) = 1$. This is the case U^2_1, which does not pose any problems. For the sequel of the proof we assume $n(F) \geq 2$, and thus no contractions happen when transitioning from $\mathrm{idg}(F)$ to $\mathrm{ig}(F)$.

The four cases $\mathrm{U}^2_n, \mathrm{U}^1_{n,i}, \mathrm{U}^0_{n,i}, \mathrm{U}^0_{n,x,y}$ are separated by vertex-degrees in the implication graph, since their degree-spectra as triples in $(\mathbb{N}_0 \cup \{+\infty\}^3)$ for the numbers of degree-2/3/4-vertices, with "inf" meaning "unbounded", are resp. $(\mathrm{inf}, 0, 0)$, $(\mathrm{inf}, 2, 0)$, $(\mathrm{inf}, 0, 4)$ and $(\mathrm{inf}, 4, 0)$. Some parameter-list $L(F)$ can be computed in polytime by performing the standardisation of the chain of 1-singular-DP-reductions leading to $\bot$, as in the proofs of Lemmas 6.3.5, 6.3.7, and 6.3.8, or they are determined from the implication graph, as in the following uniqueness argument. Namely that the parameters are uniquely determined, is read off the unlabelled implications graphs (i.e., vertices are "anonymised") as follows:

- The parameter i in $\mathrm{ig}(\mathrm{U}^1_{n,i})$ can be computed from the length of the shared path $2i - 1$ of the two cycles.

- For $\mathrm{ig}(\mathrm{U}^0_{n,i})$ there are exactly two vertices of degree 4, and there are two paths of length i and two paths of length $n - i + 1$ between them (no more). Since $i \leq n - i + 1$, we can compute i.

- For $\mathrm{ig}(\mathrm{U}^0_{n,x,y})$ there are exactly four vertices a, b, c, d of degree 3, which we can identify in such a way that $\mathrm{ig}(\mathrm{U}^0_{n,x,y})$ consists of a cycle running through these vertices in the given order, and where between a, b and c, d

there are parallel paths to the path between a, b resp. c, d on that cycle, such that this is all of the graph. These parallel paths have the same length p resp. q. W.l.o.g. $p \le q$, and now $x := p$ and $y := n - q + 1$.

This shows all the statements in the theorem. $\square$

By adding up the contributions we obtain the exact number of isomorphism types of 2-MUs of deficiency 1 as follows:

Theorem 6.3.11 *The exact number of isomorphism types of $F \in 2\text{–}\mathcal{MU}_{\delta=1}$ with $n(F) = n \in \mathbb{N}_0$ is*

$$\begin{cases} 1 & \text{if } n = 0 \\ \frac{1}{4}n(n+2) & \text{if } n \text{ is even} \\ \frac{1}{4}(n+1)^2 & \text{if } n \text{ is odd.} \end{cases}$$

This is the sequence A076921 in the OEIS ([139]), where that sequence starts with index 1*.*

Proof: We have $n = 0$ iff $F = A_0 = \{\bot\}$. For $n \ge 1$ by Theorem 6.3.10 there are precisely four cases as follows:

2 unit-clauses: $F \cong \mathrm{U}^2_{n(F)}$, with precisely 1 isomorphism type.

1 unit-clause: $F \cong \mathrm{U}^1_{n,i}$ and the number of isomorphism types is 0 for $n = 1$, and $n - 1$ for $n \ge 2$.

0 unit-clauses, 1 non-1-singulars: $F \cong \mathrm{U}^0_{n,i}$ with number of isomorphism types 0 for $n \le 2$, otherwise $\frac{n-1}{2}$ for odd n, and $\frac{n-2}{2}$ for even n.

0 unit-clauses, 4 non-1-singulars: $F \cong \mathrm{U}^0_{n,x,y}$ with the number of isomorphism types 0 for $n \le 3$, otherwise $\frac{1}{4}(n-2)^2$ for even n, and $\frac{1}{4}(n-1)(n-3)$ for odd n.

The sum for $n \ge 4$ yields $\frac{1}{4}n(n+2)$ for even n, and $\frac{1}{4}(n+1)^2$ for odd n; these formulas hold indeed for $n \ge 1$. $\square$

6.4 Singular DP-reduction and smoothing

As already discussed, a fundamental tool for the analysis of MUs is singular DP-reduction, where $\mathrm{DP}_v(F) \in \mathcal{MU}$ is guaranteed (see Section 4.5). For $F \in 2\text{–}\mathcal{MU}^*_{\delta=1}$, a variable v is singular, iff vertex v or $\overline{v}$ in $\mathrm{idg}(F)$ is linear. The classes of MUs with fixed deficiency $k \ge 1$ are stable under singular DP-reduction (Corollary 4.5.3), and since $2\text{–}\mathcal{CLS}$ is stable under resolution, also the classes $2\text{–}\mathcal{MU}_{\delta=k}$ are stable under singular DP-reduction. The basic result for this chapter, already shown in Corollary 5.4.8, is that for $F \in 2\text{–}\mathcal{MU}$ and $F' \in$

sDP(F) holds: If $\delta(F) = 1$ then $F' = \{\bot\}$, if $\delta(F) \geq 2$ then $F' \cong \mathcal{B}_{\delta(F)}$. In Section 6.3 we characterised all 2-MUs with deficiency 1, based on reversal of 1-singular DP-reduction (in a special setting; this will be taken up again in Subsection 6.4.1). To generate all 2-MUs for higher deficiencies, general singular DP-reduction (for 2-CNFs) has to be reversed.

6.4.1 1-singular DP-reduction

The nicest case of singular DP-reduction is when we have confluence, that is, $|\mathrm{sDP}(F)| = 1$. By [108, Section 5] we indeed have confluence, when performing only 1-singular DP-reduction. And furthermore it follows by the general results there, that once all 1-singular variables are eliminated, none are being reintroduced. The simple reason for confluence is that 1-singular DP-reduction does not remove 1-singular variables other than the eliminated variable (but we note that new 1-singular variables in general are created). We denote by $\boldsymbol{\mathcal{MU}^+} \subset \mathcal{MU}$ the set of **non-1-singular** $F \in \mathcal{MU}$, i.e., where every variable of F has degree at least 3 (while for nonsingular $F \in \mathcal{MU}'$ every variable has degree at least 4). For $\mathcal{C} \subseteq \mathcal{MU}$ we use $\boldsymbol{\mathcal{C}^+} := \mathcal{C} \cap \mathcal{MU}^+$. We use **1sDP($\boldsymbol{F}$)** $\in \mathcal{MU}^+$ for $F \in \mathcal{MU}$ to denote the (unique) non-1-singular MU obtained by 1-singular DP-reduction from F. The basis for Section 6.3 is that for all $F \in \mathcal{MU}$ holds $\mathrm{1sDP}(F) = \{\bot\}$ iff $\delta(F) = 1$ (Lemma 4.6.1). As mentioned, once we have removed all 1-singular variables, singular-DP-reduction never reintroduces them:

Lemma 6.4.1 *$\mathcal{MU}^+$ is stable under singular-DP-reduction.*

Proof: The only possibility of a singular DP-reduction on v with main clause $v \in C \in F \in \mathcal{MU}$ and side-clauses $\overline{v} \in D_1, \dots, D_m \in F$ decreasing the degree of a literal $x \in \mathrm{lit}(F)$ is that $x \in C \cap D_1 \cap \cdots \cap D_m$, but since $F \in \mathcal{MU}^+$, we have $m \geq 2$, and thus the literal-degree of x in $\mathrm{DP}_v(F)$ is at least two. $\square$

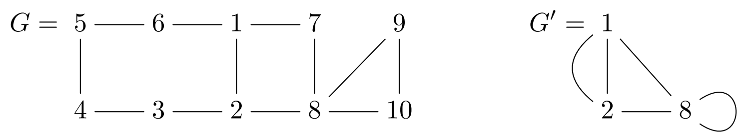
The analysis of singular DP-reduction for a class $\mathcal{C} \subseteq \mathcal{MU}$, where always stability of $\mathcal{C}$ under singular DP-reduction is assumed, now can proceed by first considering the simple confluent reduction $F \in \mathcal{C} \leadsto \mathrm{1sDP}(F) \in \mathcal{C}^+$ and characterising the elements of $\mathcal{C}^+$. The second stage then can start with $\mathcal{C}^+ \subseteq \mathcal{C}$, and need only to consider non-1-singular DP-reductions to arrive at $\mathcal{C}' = \mathcal{C} \cap \mathcal{MU}'$.

6.4.2 Smoothing of (multi-)graphs

We will now see that a general reduction operation for graphs, strongly related to the concept of "homeomorphism" of graphs, covers most cases of 1-singular DP-reduction. Indeed it is essential to consider *multi*graphs here, which allow loops and parallel edges.

Following [66, Section 7.2.4, D37], a **smoothing step** for a multigraph G chooses a linear vertex $v \in V(G)$ (Definition 3.1.5) and with $v \notin \mathrm{N}_G(v)$ (the set of neighbours of v), removes the vertex v and the two edges from G incident with v, and for the vertices u, w with $\mathrm{N}_G(v) = \{u, w\}$ (note that possibly $u = w$) adds an edge connecting u and w; thus the obtained multigraph G' has $V(G') = V(G) \setminus \{v\}$ and $E(G')(\{u,w\}) = E(G)(\{u,w\}) + 1$. We note that the degree of the remaining vertices is not changed except for the case $u = w$, in which case the degree of u decreases by one. Especially linear vertices in G different from v stay linear vertices in G', except for the case when we have two linear vertices $v \neq v'$ forming a 2-cycle (i.e., $E(G)(\{v,v'\}) = 2$), in which case the degree of v' in G' is 1 (namely $E(G')(\{v\}) = 1$).

Example 6.4.2 *Smoothing of the graph G yields the multigraph G':*



Smoothing of the cycle multigraph CG_n, $n \geq 1$, yields exactly one of the multigraphs with one vertex i, $1 \leq i \leq n$, with exactly one edge.

So performing smoothing steps on a multigraph G as long as possible results in a multigraph G' (with $V(G') \subseteq V(G)$), where G' is *uniquely determined* except for isolated cycles $C \subseteq V(G)$ (all vertices of C are linear in G) of length at least two, where exactly one $v \in C$ is chosen, and the whole cycle C is replaced by a loop at v. For 1-singular DP-reduction this choice does not happen, since the result of this situation is the (unique) empty clause.

Example 6.4.3 *Consider $\mathrm{U}_2^2 = \{\{1\}, \{-1,2\}, \{-2\}\}$: we can perform 1-singular DP-reduction on variables 1 or 2, obtaining $\{\{2\},\{-2\}\}$ or $\{\{1\},\{-1\}\}$, which corresponds to isolated cycles of length 2, but resolution in both cases yields $\perp$, while for the corresponding smoothing-operation one of the two literals is selected to label the remaining loop.*

Now assume that some linear order on the universe of vertices is given. For a (finite) digraph G by $\boldsymbol{\mathrm{sm}(G)}$ we denote the multigraph obtained from G by performing smoothing steps as long as possible, where in case of a choice the first element in the given linear order is chosen. We have shown:

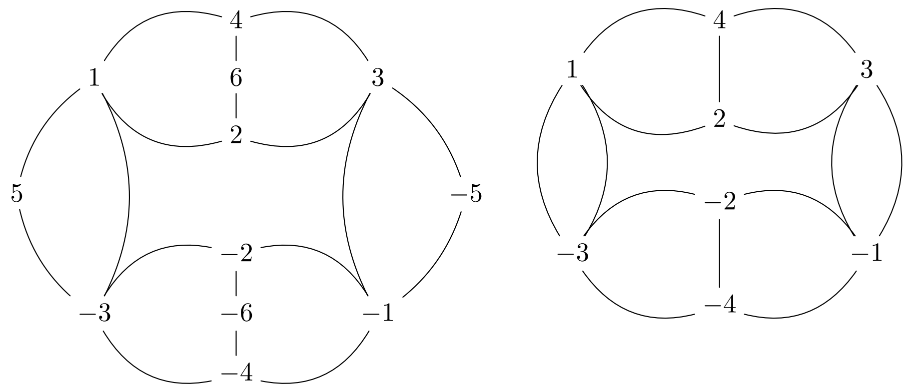
Lemma 6.4.4 *For a digraph G we have precisely two cases for $\mathrm{sm}(G)$ as follows:*

1. *If G has no isolated cycles, then smoothing is confluent (i.e., $\mathrm{sm}(G)$ does not depend on the linear order on vertices), and the vertices of $\mathrm{sm}(G)$ are the nonlinear vertices of G.*
2. *Otherwise, G has some isolated cycles $C \subseteq V(G)$ of length $|C| \geq 2$, where for the last element $v \in C$ according to the linear order we have $V(\mathrm{sm}(G)) \cap C = \{v\}$, with $\deg_{\mathrm{sm}(G)}(v) = 1$ and $\mathrm{N}_{\mathrm{sm}(G)}(v) = \{v\}$. Furthermore the vertices of $\mathrm{sm}(G)$ are the nonlinear vertices of G plus these selected vertices for each isolated cycle of G, and the results obtained for different linear orders are isomorphic.*

Corollary 6.4.5 *For two multigraphs G, G', if $G \cong G'$ then* $\mathrm{sm}(G) \cong \mathrm{sm}(G')$.

Following [66, Section 7.2.4, D38], two multigraphs G, G' are **homeomorphic**, if $\mathrm{sm}(G) \cong \mathrm{sm}(G')$. So two isomorphic multigraphs are homeomorphic, but not vice versa.

Example 6.4.6 *Consider our running example F and its non-1-singular normalform* $\mathrm{1sDP}(F)$ *(Section 6.1). The multigraph* $\mathrm{mg}(\mathrm{idg}(F))$*, shown below, has four linear vertices* $5, -5, 6, -6$*, which are removed by smoothing as follows.*



The result $\mathrm{sm}(\mathrm{idg}(F))$ *is isomorphic to the multigraph* $\mathrm{mg}(\mathrm{idg}(\mathrm{1sDP}(F)))$ *which has no linear vertices (see Section 6.1 for the implication digraph of* $\mathrm{1sDP}(F)$*). That is, the results of smoothing for the implication digraph of F and* $\mathrm{1sDP}(F)$ *are isomorphic, and so their multigraphs are homeomorphic.*

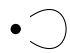
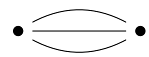
In order to present the connection between 1-singular DP-reduction for 2-MUs and smoothing of the implication graphs, we need indeed to introduce the **implication multigraph img($\boldsymbol{F}$)** for $F \in 2\text{–}\mathcal{CLS}$, which is obtained from $\mathrm{idg}(F)$ by conversion in case of $\perp \notin F$, while for $\perp \in F$ we add a new vertex $v_\perp$ to $\mathrm{img}(F \setminus \{\perp\})$, and add a loop at $v_\perp$ (of multiplicity 1). So for a variable v its variable-degree $\mathrm{vd}_F(v)$ equals the degree of vertex v as well as the degree of vertex $\overline{v}$ in $\mathrm{img}(F)$. Thus a vertex $x \in V(\mathrm{img}(F))$ is linear iff $\overline{x}$ is linear, while a variable v is 1-singular in F iff vertices $v, \overline{v}$ are linear in $\mathrm{img}(F)$. If $f : F \to F'$ is an isomorphism between $F, F' \in 2\text{–}\mathcal{CLS}$, then $f' : \mathrm{img}(F) \to \mathrm{img}(F')$ is also an isomorphism, where f' just extends the map f by mapping $f'(v_\perp) = v_\perp$.

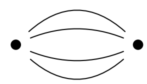
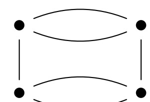
Now smoothing of $\mathrm{img}(F)$ for $F \in 2\text{–}\mathcal{MU}$ corresponds exactly to 1-singular DP-reduction for F, *except* that unit-clauses $\{x\}$ obtained by *contraction*, i.e., from $\{v, x\}, \{\overline{v}, x\}$, with v being 1-singular, never participate in the reduction process, since the multiplicity of the edge between $-x$ and x here is increased by *two*. And *except* that the clause-set-process does not record multiplicities of edges, of course:

Lemma 6.4.7 *Consider $F \in 2\text{–}\mathcal{MU}$. Then the underlying graph of* $\mathrm{sm}(F) := \mathrm{sm}(\mathrm{img}(F))$ *is* $\mathrm{ig}(F')$*, where F' is obtained from F by any series of 1-singular*

DP-reductions without using new unit-clauses obtained by contraction, where the series is maximal, and where in case $F' = \{\bot\}$ we let the single vertex $v_\bot$ of $\mathrm{img}(F')$ be the final vertex in the smoothing-sequence of $\mathrm{img}(F)$.

We call $\mathrm{sm}(F)$ the **homeomorphism type** of F. By Theorem 6.3.10 we obtain that $\mathrm{sm}(F)$ for $F \in 2\text{–}\mathcal{MU}_{\delta=1}$ is equal to exactly one of the homeomorphism types of the four cases $\mathrm{U}^2_n, \mathrm{U}^1_{n,i}, \mathrm{U}^0_{n,i}, \mathrm{U}^0_{n,x,y}$ (see Section 6.3 for the implication digraphs of these cases):

$\mathrm{sm}(\mathrm{U}^2_n) =$ , $\quad \mathrm{sm}(\mathrm{U}^1_{n,i}) =$ ,

$\mathrm{sm}(\mathrm{U}^0_{n,i}) =$ , $\quad \mathrm{sm}(\mathrm{U}^0_{n,x,y}) =$ 

Before coming to the main results of this chapter, we comment on a different study of graphs related to 2-CNFs from the literature about distinguishing satisfiability and unsatisfiability of 2-CNFs:

Example 6.4.8 *In [74] the multigraphs considered for $F \in 2\text{–}\mathcal{CLS}$, call it $\mathrm{ig}'(F)$, can be obtained from the implication graphs $\mathrm{ig}(F)$ by identifying complementary vertices (this information comes from F itself). The paper only studies the graph-cases (no parallel edges), by first applying an arbitrary preprocessing of F to remove $C, D \in F$, with $C \neq D$ but $\mathrm{var}(C) = \mathrm{var}(D)$ (these yield the parallel edges). For example the isomorphism types of $2\text{–}\mathcal{MU}_{\delta=1}$ (Theorem 6.3.10) we obtain:*

$$\mathrm{ig}'(\mathrm{U}^2_n):\ 1 - 2 - \ldots - n\ , \qquad \mathrm{ig}'(\mathrm{U}^1_{n,i}):\ \begin{tikzcd} 1 \arrow[r, dash] & 2 \arrow[r, dash] & \ldots \arrow[r, dash] & i \arrow[r, dash] \arrow[drr, dash] & \ldots \arrow[r, dash] & n-1 \arrow[d, dash] \\ & & & & & n \end{tikzcd}$$

$$\mathrm{ig}'(\mathrm{U}^0_{n,i}):\ \begin{tikzcd} 2 \arrow[r, dash] \arrow[d, dash] & \ldots \arrow[r, dash] & i \arrow[r, dash] \arrow[dll, dash] \arrow[drr, dash] & \ldots \arrow[r] & n-1 \arrow[d, dash] \\ 1 & & & & n \end{tikzcd}$$

$$\mathrm{ig}'(\mathrm{U}^0_{n,x,y}):\ \begin{tikzcd} 2 \arrow[r, dash] \arrow[d, dash] & \ldots \arrow[r, dash] & x \arrow[r, dash] \arrow[dll, dash] & \ldots \arrow[r, dash] & y \arrow[r, dash] \arrow[drr, dash] & \ldots \arrow[r, dash] & n-1 \arrow[d, dash] \\ 1 & & & & & & n \end{tikzcd}$$

Each of $\mathrm{ig}'(\mathrm{U}^0_{n,x,y})$ and $\mathrm{ig}'(\mathrm{U}^0_{n,i})$ is already homeomorphic to one of the graphs in [74, Theorem 18]. However based on the paper, 2-MUs $\mathrm{U}^2_n, \mathrm{U}^1_{n,i}$ are not "simple" as they have some unit-clauses or two clauses with same variable-set. Then the preprocessing for these clause-sets (as explained in [74, Remark 1]) yields $\{\bot\}$. Therefore this preprocessing destroys information on isomorphism types, and thus is not suitable for our investigations. An interesting aspect is that at least for the graph cases, $\mathrm{ig}'(F)$ can distinguish satisfiable and unsatisfiable F.

6.5 Weak-double-cycles for 2-MUs of higher deficiency

The operation of **splitting a vertex** x in a digraph G consists of replacing x by two new vertices u, v and an arc (u, v), such that all arcs coming into x come into u, and all arcs going out of x go out of v.

Splitting an arc (also called subdividing an arc) in a digraph G replaces an arc $(x, y) \in E(G)$ by $(x, v), (v, y)$ for a new (linear) vertex v.

$$x \longrightarrow y \qquad \rightsquigarrow \qquad x \longrightarrow v \longrightarrow y$$

A **double m-cycle** for $m \geq 3$ is the digraph obtained from some cycle graph, that is, a digraph isomorphic to $\mathrm{dg}(\mathrm{CG}_m)$ (see Section 3.3). A double cycle G is strongly connected, and every vertex has in- and out-degree 2. Furthermore for every arc in G also the reverse arc exists (this characterises the class of double cycles) and so $|E(G)| = 2 \cdot |V(G)|$ and $\delta(G) = |V(G)|$. The main class of digraphs studied here is the "m-weak-double-cycle", introduced in [136] in the context of the Even-Cycles-problem (called "weak m-double-cycles" there), while we use the terminology of [13]. Weak-double-cycles (WDCs) are obtained from double cycles by splitting some vertices or some arcs.

Definition 6.5.1 *An m-**weak-double-cycle** (m-WDC) is a digraph obtained from some double m-cycle ($m \geq 3$) by splitting some vertices or arcs.*

A double m-cycle G has m (small) cycle digraphs of length two, and two (big) cycle digraphs of length m. These $m + 2$ cycles are precisely all the cycle digraphs in G. It is easy to see that splitting arcs or vertices in any digraph G maintains the number the cycles of G (just enlarges some of them). So an m-WDC G has precisely $m + 2$ cycle digraphs. The small cycles in G are characterised by having at most 4 nonlinear vertices (in G), with at most two of them of degree 4 (in G). The big cycles ("clockwise" and "anticlockwise") contain all the overlapping vertices between small cycles, and alternately choose the "outer section" and "the inner section" of a small cycle, using the natural planar drawing of G.

Example 6.5.2 *Examples of WDCs are the implication digraph of F and its non-1-singular normalform $\mathrm{1sDP}(F)$ in Section 6.1.*

To get a better grasp on the cycle digraphs in WDCs, we introduce the general concept of the **cycle-multigraph $\mathbf{cmg}(G)$** for a graph/multigraph/digraph G, which is a multigraph with vertex-set the cycles of G (recall these are sub-graph/multigraph/digraphs), and for any two vertices g, g' the number of edges between them is $|V(g) \cap V(g')|$, i.e., the number of common vertices. For an isomorphism $f : G \to G'$, we obtain an induced isomorphism $f' : \mathrm{cmg}(G) \to \mathrm{cmg}(G')$ in the obvious way (just mapping via f the vertices of G inside the structure $\mathrm{cmg}(G)$). For an m-WDC G the cycle-multigraph $\mathrm{cmg}(G)$ is as follows:

1. There are $m + 2$ vertices, m of them in an m-cycle (the "small cycles"), and two central vertices connected to every other vertex (the "big cycles").
2. Every vertex of G has a loop, with multiplicity the size of the sub-graph ($|V(g)|$).
3. Every small-cycle-vertex connects with its neighbouring small cycles, where multiplicity of the connecting is being the number of vertices in the overlap.
4. The multiplicity of the edge between the two central vertices is the sum of these overlaps.
5. The multiplicity of the edge connecting one central vertex g with a small-cycle g' is the sum of the overlaps of g' with its small-cycle-neighbours plus the number of vertices in the "outer-/inner-section" of g' as chosen by g.

An isomorphism $f : G \to G'$ of WDCs G, G' maps small cycles of G to small cycles of G' (by the above invariant characterisation of small cycles in WDCs), and the appropriate restrictions of f yield isomorphisms of these small cycles (as sub-digraphs). Let S, S' be the induced sub-multigraphs of $\mathrm{cmg}(G), \mathrm{cmg}(G')$ given by the small-cycle-vertices. As stated above, we have the induced multigraph-isomorphism $f' : \mathrm{cmg}(G) \to \mathrm{cmg}(G')$, which induces a multigraph-isomorphism $f'' : S \to S'$, since small cycles are mapped by f' to small cycles. Furthermore, from f'' one can reconstruct f in polynomial time: f must respect the overlaps of the cycles, and then the map is fixed also on the interior vertices of the cycles. The underlying graph $\mathrm{ug}(S)$ is an m-cycle graph. The automorphism group (the self-isomorphisms together with the composition of maps) of CG_m (the cycle graph with m vertices) is the Dihedral group with $2m$ elements (m rotations and m reflections). We have arrived at an efficient process for computing the isomorphisms between WDCs:

Lemma 6.5.3 *Consider WDCs G, G'. The isomorphisms $f : G \to G'$ can be determined in polynomial time as follows, where we assume that both G, G' are m-WDCs for some $m \geq 3$ (otherwise $G \ncong G'$):*

1. *Choose any isomorphisms α, β between the cycles S, S' of small cycles in $\mathrm{cmg}(G), \mathrm{cmg}(G')$, as graphs, with CG_m.*
2. *Run through the $2m$ automorphisms of the Dihedral group, as permutations of $\{1, \ldots, m\}$, considered via α, β as an isomorphism $f'' : S \to S'$.*

3. *Keep those $f : V(G) \to V(G')$, where the extension process from f'' succeeds.*

Proof: As explained before, any isomorphism $f : G \to G'$ induces a unique isomorphism $f'' : S \to S'$ where S, S' are the induced sub-multigraphs of $\mathrm{cmg}(G), \mathrm{cmg}(G')$, respectively. On the other hand, from any f'' we can reconstruct a potential f and check whether f is an isomorphism between G, G' in polynomial time. Therefore in order to obtain all isomorphisms between G, G' we just need to run through all isomorphisms between S, S', where the details are as follows: We recall that the underlying graphs for S, S' are isomorphic to the cycle graph CG_m, and that the automorphism group of CG_m is the Dihedral group with $2m$ elements. Now consider any isomorphism $\alpha : S \to \mathrm{CG}_m$ and $\beta : S' \to \mathrm{CG}_m$ (step 1). For any arbitrary automorphism γ of CG_m we have $f'' = \beta^{-1} \circ \gamma \circ \alpha$, and in this way we can run through all the $2m$ automorphisms γ of CG_m and transfer them to isomorphisms f'' between S, S' (step 2). Finally because of the relation between f, f'', we transfer f'' to f and keep those f which are a bijection $f : V(G) \to V(G')$ with $f(E(G)) = \{(f(a), f(b)) : (a, b) \in E(G)\} = E(G')$ (Definition 3.2.1), i.e., $f : G \to G'$ (step 3). $\square$

We see that the automorphism groups of m-WDCs are subgroups of the Dihedral group with $2m$ elements (obtained in Lemma 6.5.3 by a natural filtering process). We also obtain a reasonably direct procedure for deciding isomorphism of WDCs (which indeed follows immediately from [121] by the fact that the maximum degrees of WDCs is 4):

Corollary 6.5.4 *The class of WDCs has polytime isomorphism decision.*

In general from the (unlabelled) $\mathrm{cmg}(G)$ one can not reconstruct G (up to isomorphism), but for a WDC G this is possible, and this even from $\mathrm{cmg}(\mathrm{ug}(G))$. This implies that WDCs can be reconstructed up to isomorphism from their underlying graphs. We prove this however in a more direct way, avoiding to unfold here the "full cycle-picture" (we note that $\mathrm{cmg}(\mathrm{ug}(G))$ has more elements than $\mathrm{cmg}(G)$, which corresponds to the wlog's in the direct proofs).

Between the base level of double cycles and the general level of WDCs there is the middle level of **nonlinear WDCs**, i.e., WDCs without linear vertices. Once one linear vertex has been produced via splitting of arcs, we will always keep one, and thus nonlinear WDCs are exactly generated from double cycles by (only) splitting vertices. Nonlinear m-WDCs arise from from 0 to m splittings of vertices, where splitting a degree-4-vertex yields two degree-3-vertices, and the new arc is an overlap between the two neighbouring cycles involved.

Lemma 6.5.5 *For any nonlinear WDC G, from the unlabelled* $\mathrm{mg}(G)$ *we can reconstruct G up to isomorphism (in polynomial time).*

Proof: We need to give directions to the arcs of $\mathrm{mg}(G)$, which is a big cycle of small cycles (each of length $2, 3, 4$). We just choose one of the small cycles, and choose one direction for it (does not matter which). Now those neighbouring cycles, which have a nontrivial overlap with that cycle, obtain their direction from the one arc in them, and so on. If we come to a one-point-connection between cycles, then we are free to choose a direction for the new cycle, and we force again the neighbouring cycles with an overlap. In this way we necessarily can give all edges a direction, and we obtain a digraph isomorphic to G. $\square$

It is easy to see that general WDCs G are produced by first producing some nonlinear WDC G', and then splitting arcs in G', obtaining G. In other words, using the smoothing operation (recall Definition 6.4.2), an arbitrary digraph G is a WDC iff $\mathrm{sm}(G)$ is a nonlinear WDC. Adding linear vertices still allows to apply the proof of Lemma 6.5.5, and so we obtain

Corollary 6.5.6 *For any WDC G, from the unlabelled $\mathrm{mg}(G)$ we can reconstruct G up to isomorphism (in polynomial time).*

So transpositions of WDCs are isomorphic WDCs (which allows Lemma 6.5.3 to be applied in the determination of skew-symmetries for WDCs). We can even forget the multiplicity of edges:

Corollary 6.5.7 *For any WDC G, from the unlabelled underlying graph of G we can reconstruct G up to isomorphism (in polynomial time).*

Proof: If in the big cycle there are "single edges", not part of a small cycle, then these edges are replaced by a pair of parallel edges. To the obtained multigraph, Corollary 6.5.6 is applied. $\square$

So for any WDCs G, G' we have $G \cong G'$ iff $\mathrm{ug}(G) \cong \mathrm{ug}(G')$.

Now that we know that $\mathrm{mg}(G)$ for WDCs G contains the essential information of G, we can consider the homeomorphism type of G, i.e., the homeomorphism type of $\mathrm{mg}(G)$. So we reduce $\mathrm{mg}(G)$ to $\mathrm{sm}(\mathrm{mg}(G))$ according to Lemma 6.4.4. We have $\mathrm{sm}(\mathrm{mg}(G)) = \mathrm{mg}(G')$, where G' is the nonlinear WDC which is obtained in the first phase of generating G (only splitting vertices in double cycles). So the homeomorphism type of G (all digraphs homeomorphic to G) is the set of all WDCs H with $\mathrm{sm}(\mathrm{mg}(G)) \cong \mathrm{sm}(\mathrm{mg}(G))$, which is equivalent to $G' \cong H'$, where G', H' are the nonlinear WDCs obtained of the first phase of generation. We will conclude this section on WDCs by giving a concrete description of the isomorphism type of nonlinear WDCs, in terms of "binary bracelets" (instead of "bracelet" also "turnover necklace" is used). Since formulas for counting bracelets are known ([61]), this yields an explicit formula for the number of isomorphism types of nonlinear m-WDCs.

Definition 6.5.8 *A **binary bracelet** of length $m \in \mathbb{N}$ is a binary string of length m (i.e., a m-tuple $(a_1, \ldots, a_m) \in \{0,1\}^m$). Two binary bracelets are **equivalent** if one can be obtained from the other by rotation or reflection.*

Example 6.5.9 *Numerical data on the number of equivalence classes of binary bracelets of length m is given in the OEIS ([139, Sequence A000029]). For example for $m = 3$ one has 4 equivalence classes as*

$$000, 100, 110, 111,$$

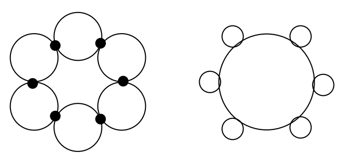
and for $m = 4$ there are 6 classes as

$$0000, 1000, 1100, 1010, 1110, 1111.$$

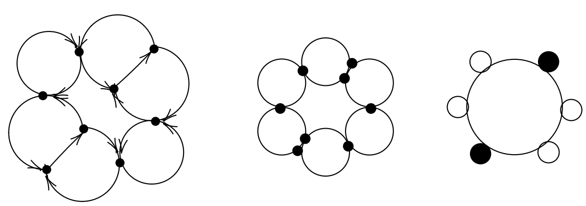
For the graphical illustration of these examples see [41].

A nonlinear m-WDC G is a big cycle of m small cycles, where the overlap of two neighbouring cycles is either a vertex or a single edge. Such multigraphs are equivalent to a binary bracelet of length m, where the one- resp. the two-vertex overlap is represented by 0 resp. 1. This can be seen by considering $\mathrm{cmg}(G)$ of a nonlinear m-WDC G, and the multi-cycle S of m small cycles in $\mathrm{cmg}(G)$ (recall the discussion before Lemma 6.5.3).

Example 6.5.10 *The implication digraph $\mathrm{idg}(\mathcal{B}_3)$, shown is Section 6.1, is a 6-WDC. So $\mathrm{idg}(\mathcal{B}_3)$, as a big cycle of 6 small cycle, and the graphical illustration of its corresponding bracelet (i.e., 000000) are as follows:*



Consider the non-1-singular normalform $\mathrm{1sDP}(F)$ of our running example F (Section 6.1). The implication digraph of $\mathrm{1sDP}(F)$ is a nonlinear 6-WDC, and its representation as a cycle of small cycles, and the graphical illustration of its corresponding bracelet (i.e., 010010 when starting from top-left and moving clockwise) are as



Since due to nonlinearity the small cycles have no internal structure other than given in the overlaps, G can be reconstructed up to isomorphism from S. The information we are seeking is contained in the multiplicity of connecting edges in the multi-cycle, and we can drop the loops at the vertices. So the isomorphism type of G is represented by the cycle-multigraph S' of length m obtained from S by removing all loops: neighbouring vertices in S' are connected by one/two edges if they intersect in one/two vertices. Translating "one edge" into 0 and "two edges" into 1, we obtain the derived bracelet (up to equivalence of bracelets; we pick an arbitrary starting point in the cycle, and pick one of the two directions). We have shown:

Lemma 6.5.11 *For two nonlinear WDCs G, G' we have $G \cong G'$ iff the derived binary bracelets are equivalent.*

Thus the number of isomorphism types of nonlinear m-WDCs is the number of equivalence classes of binary bracelets. Using the bounds in [133], an asymptotic formula for the number of binary bracelets of length m is $2^m m^{-1} f(m)$ where $2^{-1} \le f(m) \le 2$.

6.6 Classifying 2-MUs of higher deficiency

We fix now $k \ge 2$ and consider $F \in 2\text{–}\mathcal{MU}_{\delta=k}$. Two reminders: F does not have unit-clauses (but is 2-uniform; Lemma 5.4.1), and every literal occurs in F at most twice (Lemma 5.4.2). From not having unit-clauses we get:

Lemma 6.6.1 *From $2\text{–}\mathcal{MU}^+_{\delta=k}$, $k \ge 2$ we obtain $2\text{–}\mathcal{MU}_{\delta=k}$ by repeated applications of 1-singular extension, which means that for $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ one chooses $\{x, y\} \in F$ ($x \ne y$ holds) and a new variable v, and replaces $\{x, y\}$ by $\{v, x\}, \{\overline{v}, y\} \in F$.*

Proof: 1-singular DP-reduction for any $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ with $k \ge 2$ does not create any unit-clause since singular DP-reduction preserves deficiency and 2-MUs with a unit-clause have deficiency 1 (Lemma 5.4.1). Thus DP-reduction of F on any 1-singular variable, removes only one variable, and leaves other literal-degrees unchanged (i.e., does not create any new 1-singular variable). Therefore complete 1-singular DP-reduction of F is confluent to $F' := \mathrm{1sDP}(F) \in 2\text{–}\mathcal{MU}^+_{\delta=k}$ (Section 4.5), and by performing reverse of this process (i.e., 1-singular extension) we create all elements of $2\text{–}\mathcal{MU}_{\delta=k}$ from the elements of $2\text{–}\mathcal{MU}^+_{\delta=k}$. □

The related reduction by 1-singular DP-reduction reduces F to $F' := \mathrm{1sDP}(F) \in 2\text{–}\mathcal{MU}^+_{\delta=k}$. For the implication digraphs $G := \mathrm{idg}(F)$ and $G' := \mathrm{idg}(F')$ we have now $\mathrm{mg}(G') = \mathrm{sm}(\mathrm{mg}(G))$ by Lemma 6.4.7 (contractions are impossible).

Our new starting point is now F', and we perform singular DP-reductions, which by Lemma 6.4.1 are necessarily non-1-singular, that is, eliminate variables of degree 3 (also called "2-singular variables", since there are two side-clauses). So consider a variable v of degree 3 in F, with occurrences $\{v, x\}, \{\overline{v}, y\}, \{\overline{v}, z\} \in F$. Again we do not have contraction here, that is $x \ne y$ and $x \ne z$ (otherwise a unit-clause would be created but recall that singular DP-reduction preserves deficiency and 2-MUs with a unit-clause have deficiency 1), thus DP-reduction for v increases the literal-degree of x by one, and thus not only literal v occurs only once in F, but also literal x (and $\mathrm{var}(x)$ is also 2-singular). We have shown that a singular DP-reduction for any $F' \in 2\text{–}\mathcal{MU}^+_{\delta=k}$ removes one 2-singular variable (degree-3-variable), transforms one 2-singular variable (degree-3-variable) into a nonsingular variable (degree-4-variable), and leaves other literal-degrees unchanged. Since this reduction process for F' ends with a clause-set isomorphic to $\mathcal{B}_k$, which has k variables, all of degree 4 (nonsingular), there can be at most k singular DP-reductions for F'. So the reverse of this process generates the elements of $2\text{–}\mathcal{MU}^+_{\delta=k}$:

Lemma 6.6.2 *We obtain* $2\text{–}\mathcal{MU}^{+}_{\delta=k}$ *for* $k \geq 2$ *by starting with any clause-set isomorphic to* $\mathcal{B}_k$*, and then repeatedly applying up to* k *times the following process for* $F \in 2\text{–}\mathcal{MU}^{+}_{\delta=k}$*: choose some literal* x *which occurs positively and negatively twice in* F*, and for the occurrences* $\{x,a\},\{x,b\},\{\overline{x},c\},\{\overline{x},d\} \in F$ *and a literal* y *with underlying new variable* $\mathrm{var}(y)$*, replace the two* x*-clauses by* $\{y,x\},\{\overline{y},a\},\{\overline{y},b\}$.

Proof: Lemma 6.4.1 together with Lemmas 5.4.1 and 5.4.2 imply that non-1-singular DP-reduction for any $F \in 2\text{–}\mathcal{MU}^{+}_{\delta=k}$ removes exactly one 2-singular variable, transforms one 2-singular variable into a nonsingular variable, and does not change other literal-degrees (so does not create any new singular variable). Since complete non-1-singular DP-reduction of F is confluent to some clause-set isomorphic to $\mathcal{B}_k$ (Corollary 5.4.8), the reverse process creates the elements of $2\text{–}\mathcal{MU}^{+}_{\delta=k}$ starting from some clause-set isomorphic to $\mathcal{B}_k$. Furthermore since $\mathcal{B}_k$ has precisely k nonsingular variables of degree 4 (Definition 5.2.18) and the creation process does not introduce new nonsingular variable, there can be at most k applications of singular extension in the creation process. $\square$

So altogether we obtain all $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ by first applying Lemma 6.6.2, obtaining $F' \in 2\text{–}\mathcal{MU}^{+}_{\delta=k}$, which is taken as starting point for applying Lemma 6.6.1. Such a generation sequence can be computed in polynomial time for F, by first computing $1\mathrm{sDP}(F)$, which in turn is reduced by singular DP-reduction, and then reversing the whole reduction sequence.

We now show that the implication digraphs of $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ are $2k$-WDCs. To start, we have $\mathrm{idg}(\mathcal{B}_k) \cong \mathrm{dg}(\mathrm{CG}_{2k})$:

$$\mathrm{idg}(\mathcal{B}_k) = \begin{tikzcd}
1 \arrow[r, bend left] \arrow[d, bend left] & 2 \arrow[l, bend left] \arrow[r, bend left] & \cdots \arrow[l, bend left] \arrow[r, bend left] & k-1 \arrow[l, bend left] \arrow[r, bend left] & k \arrow[l, bend left] \arrow[d, bend left] \\
-k \arrow[u, bend left] \arrow[r, bend left] & -(k-1) \arrow[l, bend left] \arrow[r, bend left] & \cdots \arrow[l, bend left] \arrow[r, bend left] & -2 \arrow[l, bend left] \arrow[r, bend left] & -1 \arrow[l, bend left] \arrow[u, bend left]
\end{tikzcd} .$$

Now consider Lemma 6.6.2. We replace two clauses $\{x,a\},\{x,b\}$ by three clauses $\{y,x\},\{\overline{y},a\},\{\overline{y},b\}$. For the implication digraph this means the transition:

$$\begin{tikzcd}
\overline{a} \arrow[dr] & & c \\
 & x \arrow[ur] \arrow[dr] & \\
\overline{b} \arrow[ur] & & d
\end{tikzcd}
\rightsquigarrow
\begin{tikzcd}
\overline{a} \arrow[dr] & & & c \\
 & \overline{y} \arrow[r] & x \arrow[ur] \arrow[dr] & \\
\overline{b} \arrow[ur] & & & d
\end{tikzcd}$$

$$\begin{tikzcd}
\overline{c} \arrow[dr] & & a \\
 & \overline{x} \arrow[ur] \arrow[dr] & \\
\overline{d} \arrow[ur] & & b
\end{tikzcd}
\rightsquigarrow
\begin{tikzcd}
\overline{c} \arrow[dr] & & & a \\
 & \overline{x} \arrow[r] & y \arrow[ur] \arrow[dr] & \\
\overline{d} \arrow[ur] & & & b
\end{tikzcd}$$

We see that this can be obtained up to isomorphism of digraphs by first, say, splitting vertex x, and then splitting vertex $\overline{x}$ (recall that the vertices in implication digraphs are just placeholders, and also do not know about complementation). We have shown:

Lemma 6.6.3 *The implication digraph of $F \in 2\text{-}\mathcal{MU}^{+}_{\delta=k}$ for $k \geq 2$ is a non-linear $2k$-WDC.*

Proof: As shown before $\mathrm{idg}(\mathcal{B}_k) \cong \mathrm{dg}(\mathrm{CG}_{2k})$ is a $2k$-WDC. Then the creation process for $F \in 2\text{-}\mathcal{MU}^{+}_{\delta=k}$ in Lemma 6.6.2 and the correspondence between performing reverse non-1-singular DP-reduction for a 2-MU F and splitting vertices in $\mathrm{idg}(F)$ implies that $\mathrm{idg}(F)$ can be created from some $G \cong \mathrm{idg}(\mathcal{B}_k)$ by splitting some vertices. That is, $\mathrm{idg}(F)$ is a non-linear $2k$-WDC. $\square$

In the same way, obviously Lemma 6.6.1 means the following transition of the implication digraph:

$$\overline{x} \longrightarrow y \quad \rightsquigarrow \quad \overline{x} \longrightarrow v \longrightarrow y$$

$$\overline{y} \longrightarrow x \quad \rightsquigarrow \quad \overline{y} \longrightarrow \overline{v} \longrightarrow x$$

and thus one step of 1-singular extension is captured by two applications of arc-splitting.

Altogether we have shown:

Theorem 6.6.4 *The implication digraph of $F \in 2\text{-}\mathcal{MU}_{\delta=k}$ with $k \geq 2$ is a $2k$-WDC.*

Proof: By Lemma 6.6.1 we obtain F from some $F' \in 2\text{-}\mathcal{MU}^{+}_{\delta=k}$ by performing reverse 1-singular DP-reduction, which corresponds to splitting some arcs in $\mathrm{idg}(F')$. Now by Lemma 6.6.3 $\mathrm{idg}(F')$ and so $\mathrm{idg}(F)$ are $2k$-WDCs. $\square$

Lemma 6.6.5 *The implication digraph of $F \in 2\text{-}\mathcal{MU}_{\delta=k}$ for $k \geq 2$ has precisely $2k+2$ cycle digraphs, and precisely two contradictory cycles.*

Proof: The statement follows from Theorem 6.6.4 and the definition of WDCs (see the properties of WDSs presented after Definition 6.5.1). $\square$

Example 6.6.6 *In this example we show that for $F \in 2\text{-}\mathcal{MU}_{\delta=k}$, the contradictory cycles in the implication digraph are not always Hamiltonian, and only special 2-MUs have such cycles. The problem is that a contradictory cycle has to use one of the two arcs associated with a clause, but not necessarily both. For example, for 2-MUs of deficiency 1 only $\mathrm{idg}(\mathrm{U}^2_n)$ has a Hamiltonian cycle (Section 6.3), while the other cases has no such cycle. Another counter example is our running example F (Section 6.1), where each of vertices 5,-5 is in only one contradictory cycle (and so $\mathrm{idg}(F)$ has no Hamiltonian cycle).*

More generally for $F \in 2\text{-}\mathcal{MU}_{\delta=k}$ with $k \geq 2$, in the process of obtaining $\mathrm{idg}(F)$ from $\mathrm{idg}(\mathcal{B}_k)$, splitting vertices maintains the Hamiltonian cycles (just enlarges them), and so an implication digraph with no linear vertices has precisely two Hamiltonian cycles which are the two big contradictory cycles of the digraph. But splitting an arc might destroy this property.

We are now ready to prove the main technical result of this chapter, showing that the implication digraph of F has a unique skew-symmetry, which yields the complementation of literals, and thus one can reconstruct F from the (unlabelled) $\mathrm{idg}(F)$. Since F does not have unit-clauses, we have to exclude skew-symmetries, which yield them (otherwise uniqueness would not hold). So we define, that a skew-symmetry σ of a digraph G is **unit-free** if $\forall\, v \in V(G) : (v, \sigma(v)) \notin E(G)$. We start with a lemma on the skew-symmetries of the cycle digraph:

Lemma 6.6.7 *Consider a cycle digraph G with $n \geq 2$ vertices. If n is odd then there is no complementation. For even n there are exactly $n/2$ complementations σ. All clause-sets given by (G, σ) are isomorphic to U_n^2 (and thus σ has a unit).*

Proof: W.l.o.g. we assume $G = 1 \to \ldots \to n \to 1$. Recall, the skew-symmetries are the digraph-isomorphisms $f : G \to G^{\mathrm{t}}$, which as permutations of $V(G)$ are involutions and do not have fixed-points. The isomorphisms from G to G^{t} are given by the n rotations, the n symmetries of G, composed with one fixed isomorphism from G to G^{t}, where one can use the rotation "anticlockwise", i.e., $1 \mapsto 1, 2 \mapsto n, \ldots, n \mapsto 2$. This yields that precisely the n reflections of the (undirected) cycle CG_n are the sought isomorphisms. They all are involutions, and exactly half of them are fixed-point free. Recalling the implication digraph of U_n^2 (given after Lemma 6.3.5), we see that these skew-symmetries all yield clause-sets isomorphic to U_n^2. $\square$

We also need a variation:

Lemma 6.6.8 *Consider a digraph G which is the union of two cycle digraphs G', G'', i.e., $V(G) = V(G') \cup V(G'')$ and $E(G) = E(G') \cup E(G'')$, such that the overlap $V(G') \cap V(G'')$ is not empty, and the induced sub-digraph on it is a path of length $|V(G) \cap V(G')| - 1$. Then every skew-symmetry of G has a unit.*

Proof: Assume a unit-free complementation σ of G, and let $F \in 2\text{–}\mathcal{CLS}$ be the corresponding clause-set. F is unsatisfiable, since G is (minimal) strongly connected. Indeed F is minimally unsatisfiable, since otherwise there would be a sub-digraph G' stable under σ with at least two arcs less, corresponding to an MU inside F, but by Lemma 6.6.7 G' can not have a contradictory cycle. The homeomorphism type of G is that of two cycles, either with a one-point connection or with a nontrivial overlap. If $\delta(F) \geq 2$, then by Lemma 6.6.3 there would be $2k$ cycles in it, which is not possible. So $\delta(F) = 1$. But also this requires at least three cycles, since F does not have a unit-clause (see the homeomorphism types for deficiency 1 as shown after Lemma 6.4.7). $\square$

We are ready to show that WDCs can yield at most one 2-MU (in the precise sense, not just up to isomorphism):

Theorem 6.6.9 *Every WDC has at most one unit-free complementation.*

Proof: Consider a WDC G and a unit-free skew-symmetry σ for G. We show that σ is unique. As above, $\sigma : G \to G^{\mathrm{t}}$ is an isomorphism, where G^{t} is also a WDC. We obtain the induced isomorphism $\sigma' : \mathrm{cmg}(G) \to \mathrm{cmg}(G^{\mathrm{t}})$ (recall the discussion before Lemma 6.5.3). Furthermore, there is the induced isomorphism $\sigma'' : S \to S'$, where S, S' are the induced sub-graphs given by the small cycles in G, G^{t} (here indeed just as the sub-graphs, not as sub-multigraphs). σ' is just σ on the vertices, transported to the small cycles as sub-digraphs of G. Now the small cycles of G^{t} are essentially the same as the small cycles of G, except of the reversed direction of the arcs. Thus w.l.o.g. we can consider σ'' as an automorphism (symmetry) of the undirected m-cycle S (where G is an m-WDC), that is, σ'' is one of the m rotations and m reflections.

If σ'' had a fixed-point (would map one small cycle of G to itself), then by Lemma 6.6.7, σ would not be unit-free. If m would be odd, then the only symmetries without fixed-points are the nontrivial rotations, but for odd m none of them is an involution. So m is even. This leaves for σ'' the m reflections and the point-symmetry, the rotation by 180 degrees. We now exclude the reflections, which proves the theorem (since from σ'' one can reconstruct σ). And this is indeed easy now: Assume σ is a reflection. As already used in Lemma 6.6.7, there are two neighbouring vertices of S which are mapped by σ'' to each other. Now by Lemma 6.6.8, σ again would not be unit-free. $\square$

We finally have shown the main result of this chapter:

Theorem 6.6.10 *Consider $F, F' \in 2\text{-}\mathcal{MU}_{\delta=k}$ with $k \geq 2$. Then the set of isomorphisms $f : F \to F'$, as maps $f : \mathrm{lit}(F) \to \mathrm{lit}(F')$, is equal to the set of isomorphisms $f : \mathrm{idg}(F) \to \mathrm{idg}(F')$ (as maps $f : V(\mathrm{idg}(F)) \to V(\mathrm{idg}(F'))$).*

Proof: In general every isomorphism from F to F' is an isomorphism from $\mathrm{idg}(F)$ to $\mathrm{idg}(F')$; so assume that f is an isomorphism from $\mathrm{idg}(F)$ to $\mathrm{idg}(F')$, and we have to show that f is an isomorphism from F to F'. This follows by observing that f transports any skew-symmetry σ for $\mathrm{idg}(F)$ to a skew-symmetry σ_f for $\mathrm{idg}(F')$, and f then becomes an isomorphism from $(\mathrm{idg}(F), \sigma)$ to $(\mathrm{idg}(F'), \sigma_f)$. By Theorem 6.6.9, σ is the natural skew-symmetry of $\mathrm{idg}(F)$ as given by the complementation of F, and σ_f is the natural skew-symmetry as given by F'. Since digraphs with given skew-symmetry are the same as 2-CNFs, the statement follows. $\square$

6.7 Applications

We obtain a number of applications:

Corollary 6.7.1 *For $F, F' \in 2\text{-}\mathcal{MU}_{\delta=k}$ holds $F \cong F'$ iff $\mathrm{idg}(F) \cong \mathrm{idg}(F')$, where the implication digraphs are $2k$-WDCs.*

Proof: The assertion follows from Theorems 6.6.4 and 6.6.10. $\square$

That the isomorphisms between 2-MUs are exactly the isomorphisms between their implication digraphs (Theorem 6.6.10), together with the process in Lemma 6.5.3 for computing the isomorphism between two WDCs, we obtain the following results.

Corollary 6.7.2 *For $F, F' \in 2\text{–}\mathcal{MU}_{\delta=k}$ the number of isomorphisms between F and F' is at most $4k$. The automorphism group of F is a subgroup of the Dihedral group with $4k$ elements, and construction of the group table can be done in time $O(k \cdot \|F\|)$, using $\|F\|$ for the length of F.*

Corollary 6.7.3 *The isomorphism problem for inputs $F, F' \in 2\text{–}\mathcal{MU}$ can be decided in time $O(\delta(F) \cdot \|F + F'\|))$, assuming $\delta(F) = \delta(F')$ (otherwise $F \not\cong F'$).*

Corollary 6.7.4 *The number of isomorphism types of $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ with (exactly) $n(F) = n \in \mathbb{N}_0$ variables is $\Theta(n^{3k-1})$ (for fixed deficiency k).*

Proof: There are $2k$ cycles in $\mathrm{idg}(F)$, with half of them duplicated by skew-symmetry, so that we have k essential cycles. These cycles are arranged in a big cycle, and so have three non-overlapping parts, say the upper, right, and lower parts, which makes $3k$ numbers adding up to n, and so the number of isomorphism types is $O(n^{3k-1})$. By Corollary 6.7.2 the equivalence classes are bounded. $\square$

For clause-sets $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ we now characterise the homeomorphism types $\mathrm{sm}(F)$, which by Lemma 6.4.7 are the implication graphs of $1\mathrm{sDP}(F) \in 2\text{–}\mathcal{MU}^{+}_{\delta=k}$ (the canonical normalform of F obtained by 1-singular DP-reduction).

Corollary 6.7.5 *The homeomorphism types of $2\text{–}\mathcal{MU}_{\delta=k}$ are in one-to-one correspondence with the equivalence classes of binary bracelets of length k.*

Proof: The homeomorphism types of $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ are the isomorphism types of the implication graphs $\mathrm{ig}(F')$ for $F' := 1\mathrm{sDP}(F) \in 2\text{–}\mathcal{MU}^{+}_{\delta=k}$ (Lemma 6.4.7), and correspond to the isomorphism types of nonlinear $2k$-WDCs with skew-symmetry (Lemma 6.6.3). Isomorphism types of nonlinear $2k$-WDCs correspond to equivalence classes of binary bracelets of length $2k$ (recall Lemma 6.5.11), and due to skew-symmetry, half of them are discarded. $\square$
For example, the implication digraph of the non-1-singular normalform for our running example $F \in 2\text{–}\mathcal{MU}_{\delta=3}$ (introduced in the Introduction) is a nonlinear 6-WDC where the derived bracelet (starting top-left, moving clockwise) is 010.

Example 6.7.6 *As shown in Example 6.5.10, for the implication digraph of the non-1-singular normalform* $1\mathrm{sDP}(F)$ *of our running example F (see Section 6.1), the derived binary bracelet is* 010010. *Therefore the homeomorphism type* *of* $1\mathrm{sDP}(F)$ *and so* F *corresponds to the equivalent class of* 010 *(note that due to skew-symmetry, we only need half of* 010010*).*

Chapter 7

Conclusion and outlook

This thesis extended the understanding of the structure of MUs via investigating their combinatorial properties. Connecting MUs to graph theory, we used the strength of graph-theoretical reasoning in combination with logical reasoning to classify various classes of MUs. The main contribution of Chapter 5 was to show the strong correspondence between the new class $\mathcal{DFM}$ (using simple syntactical criteria) and the powerful world of MSDs, via the positive implication digraph. Via saturation and marginalisation, we related this class to two basic classes of MUs (namely, $\mathcal{MU}'_{\delta=2}$ and $2\text{–}\mathcal{MU}'$) and achieved their complete classification. In Chapter 6 we showed that implication digraphs of 2-MUs are closely related to skew-symmetric WDCs. So we reduced determining isomorphisms/automorphisms of 2-MUs to a purely graph-theoretical problem between simple digraphs. As direct applications of this relation we obtained a full classification of 2-MUs together with a polytime isomorphism decision.

To conclude, a more detailed summary of contributions of Chapters 5 and 6 together with discussions on computing special MUSs, conjectures and future directions are presented in Sections 7.1 resp. 7.2. Also in Section 7.3 we discus the Finite-Patterns Conjecture, and give an overview of the main characterisations of MUs and open questions. Finally in Section 7.4 we review some connections between MUs and graph theory, and discuss further conjectures and open questions.

7.1 FMs and DFMs

In Chapter 5 we introduced the novel classes $\mathcal{FM}$ and $\mathcal{DFM}$, which offer new conceptual insights into MUs. Fundamental for $F \in \mathcal{FM}$ is the observation, that the easy syntactical criterion of having both full monotone clauses immediately yields the complete understanding of the semantics of the core (F without the full monotone clauses). Namely that the satisfying assignments of the core are precisely the negations of the full monotone clauses, and so all variables are either all true or all false, i.e., all variables are equivalent (AllEqual).

$\mathcal{DFM}$ is the class of FMs where the core is a 2-CNF. This is equivalent to the clauses of the core, which must be mixed binary clauses $\{\overline{v}, w\}$, constituting an MSD via the arcs $v \to w$. Due to the strong correspondence between DFMs and MSDs, once we connect a class of MUs (e.g. $\mathcal{MU}_{\delta=2}$ and 2–$\mathcal{MU}$) to $\mathcal{DFM}$, we can use graph-theoretical reasoning. As a first application of this approach, we provided the known characterisations of $\mathcal{MU}'_{\delta=2}$ and 2–$\mathcal{MU}'$ in an accessible manner, unified by revealing the underlying graph-theoretical reasoning. We remark that another conceptual contribution of Chapter 5 was to strengthen the Splitting Ansatz by saturation, in two forms, full saturation for MUs of deficiency 2, and local saturation, which is introduced for the first time in this work, for 2-MUs.

Now that we understand $\mathcal{DFM}$, we need to extend this knowledge, using the various relations between MUs. Below are some examples of related open questions:

1. Which MUs reduce via singular DP-reduction to elements of $\mathcal{DFM}$? Let $\mathcal{DFM}^+$ be the set of $F \in \mathcal{MU}$ with $\mathrm{sDP}(F) \cap \mathcal{DFM} \neq \emptyset$. In other words, performing singular extensions starting at $\mathcal{DFM}$ we introduce "trivial variables" and obtain finally $\mathcal{DFM}^+$. So we have $\mathcal{DFM} \cup \mathcal{MU}_{\delta \leq 2} \cup$ 2–$\mathcal{MU} \subseteq \mathcal{DFM}^+$. Is there a "nice" description of $\mathcal{DFM}^+$? Do we have confluence modulo isomorphism, that is, does for all $F \in \mathcal{DFM}^+$ and $F', F'' \in \mathrm{sDP}(F)$ hold $F' \cong F''$? If not, do we at least always have $\mathrm{sDP}(F) \subseteq \mathcal{DFM}$?

2. What are the saturations of the elements of $\mathcal{DFM}$ (these only concern the binary clauses)? It is not hard to see that the only saturated elements of $\mathcal{DFM}$ are the trivial elements and the $\mathcal{F}_n$. Saturations in a sense label the arcs of the underlying MSDs and break the flow of free movements between the cycles.

3. What are the marginalisations of the elements of $\mathcal{DFM}$ (this can only concern the two monotone clauses; in Lemma 5.2.19 we have seen an example)?

Considering the isomorphism problem, as explained in [144] the graph isomorphism for MSDs conjectured to be as hard as deciding graph isomorphism. That is, for MSDs G, G', the decision "whether $G \cong G'$" is GI-complete. Therefore the isomorphism decision for strong digraphs (SDs) is also conjectured to be GI-complete, and we immediately obtain the following conjecture (see Section 5.2):

Conjecture 7.1.1 *The isomorphism problem for classes $\mathcal{DFM}$ and $\mathcal{FM}$ is GI-complete.*

In Section 7.3 we discuss two conjectures concerning the isomorphism problem for FMs and DFMs with fixed deficiency.

Finally we discuss an application of DFMs concerning enumeration of MUSs (listing with no repetition all the MU-Sub-clause-sets in an unsatisfiable clause-set). First we consider mixed binary clause-sets F where the positive implication digraph $\mathrm{pdg}(F)$ is an SD. Adding the full positive and negative clauses to F, we obtain an unsatisfiable clause-set F'. We see that any MUS in F' has some mixed binary clauses plus both full monotone clauses (otherwise would be satisfiable). Thus MUSs of F' are DFM, and their positive implication digraphs are sub-digraphs of $\mathrm{pdg}(F)$. Since $\mathrm{pdg}(F)$ is an SD, by Lemma 5.2.5 and Theorem 5.2.9 MUSs of F' correspond exactly to the minimal strong sub-digraphs of $\mathrm{pdg}(F)$:

Lemma 7.1.2 *For a mixed binary clause-set F with $\mathrm{pdg}(F)$ being an SD, let P, N be the full positive resp. full negative clauses over $\mathrm{var}(F)$. Then MUSs of $F' := F \mathbin{\dot\cup} \{P, N\}$ are DFM and correspond precisely to the MSDs in $\mathrm{pdg}(F)$.*

Considering enumeration complexity, [76] studied the problem of enumerating all minimal strong sub-digraphs of a given SD, and showed that this problem can be solved in "incremental polynomial time" ([76, Theorem 2]). An enumeration algorithm is incremental polynomial time if the time needed to enumerate the first i outputs is polynomial in i and in the size of the input (an overview of enumeration complexity classes can be found in [32] and in Chapter 2 of [141]). Now by Lemma 7.1.2 it follows that:

Lemma 7.1.3 *For inputs F' as explained in Lemma 7.1.2, we can enumerate in incremental polynomial time all contained MUSs.*

Proof: By Lemma 7.1.2, any MUS F'' of F' is a DFM, and $\mathrm{pdg}(F'')$ is an MSD in $\mathrm{pdg}(F)$ (see Definition 5.2.3); also for any MSD G in $\mathrm{pdg}(F)$, the clause-set $\mathrm{mcs}(G)$ is a DFM and an MUS of F' (recall the two formations in Theorem 5.2.9). Since MSDs in a given SD can be enumerated in incremental polynomial time ([76]), the statement follows. □

In general for an unsatisfiable clause-set with both full monotone clauses and some binary clauses, the positive implication digraph is not SD. So in order to generalise Lemma 7.1.3 we need to restrict to the strongly connected components of the positive implication digraph. A basic observation is that the strongly connected components of a given digraph can be computed in linear time (there are several known algorithms, see for example [143]). So [76, Theorem 2] can be generalised to all digraphs, i.e., for a given digraph all minimal strong sub-digraphs can be enumerated in incremental polynomial time. Therefore we can generalise Lemma 7.1.3 to clause-sets consisting of some mixed binary clauses and the two full monotone clauses as follows (note that MUSs are all DFMs):

Theorem 7.1.4 *For inputs F which are mixed binary clauses plus the two full monotone clauses, all MUSs of F can be enumerated in incremental polynomial time.*

Proof: Let F' be the set of mixed binary clauses in F. Since MUSs of F are DFM (otherwise would they be satisfiable), they correspond precisely to the MSDs in the strongly connected components of $\mathrm{pdg}(F')$ (via the two formations in Theorem 5.2.9). The linear time algorithm for enumeration of the strongly connected components of a given digraph G in [143], and the incremental polynomial time algorithm for enumerating all MSDs in a given SD shown in [76] imply that enumeration of all MSDs in G can be done in incremental polynomial time. So all MSDs in $\mathrm{pdg}(F')$ can be listed in incremental polynomial time, from which the assertion follows. $\square$

Theorem 7.1.4 can be generalised to all unsatisfiable clause-sets F which are 2-CNFs plus two full complementary clauses, with the additional assumption that they are subsumption-free: Obtain F' from F by renaming the full (complementary) clauses to full positive and negative clauses. Then $F' \cong F$ consists of some mixed binary clauses plus the two full monotone clauses (otherwise there would be some binary monotone clause subsuming one of the full clauses). Now by Theorem 7.1.4 we can enumerate MUSs of F' and so F in incremental polynomial time.

7.2 2-MUs

In Chapter 6 we determined the structure of 2-MUs, using their implication digraphs. A non-trivial result from the literature (which we also proved in Corollary 5.4.8) is that complete singular DP-reduction of a 2-MU F with deficiency k yields a clause-set isomorphic to $\mathcal{B}_k$. Reverting this process, we showed that the implication digraph $\mathrm{idg}(F)$ for $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ is a $2k$-WDC. As multigraphs, $2k$-WDCs are essentially one big cycle of $2k$ small cycles, where their homeomorphism types are the binary bracelets.

By combining arguments from logic and graph theory, we showed that every WDC has at most one skew-symmetry. Thus 2-MUs F, F' are isomorphic iff their implication digraphs $\mathrm{idg}(F), \mathrm{idg}(F')$ are isomorphic. Furthermore we showed that the isomorphisms between 2-MUs F, F' are exactly the isomorphisms between $\mathrm{idg}(F), \mathrm{idg}(F')$. So we reduced determining isomorphisms/automorphisms of 2-MUs to a purely graph-theoretical problem between digraphs, where it is easy to see that the automorphisms of $F \in 2\text{–}\mathcal{MU}_{\delta=k}$ form a subgroup of the Dihedral group with $4k$ elements.

Another contribution was to show that the special case of 2-MUs without 1-singular variables, $2\text{–}\mathcal{MU} \cap \mathcal{MU}^+$, corresponds exactly to binary bracelets. In Section 1.4.2 we discussed that the class of $F \in 2\text{–}\mathcal{MU} \cap \mathcal{MU}^+$ is similar to $\mathcal{SMU}_{\delta=1}$ and $\mathcal{MMU}_{\delta=1}$ in a sense that they all have polytime isomorphism decision with super-polynomially many isomorphism types. Now an open question is whether the class $2\text{–}\mathcal{MU} \cap \mathcal{MU}^+$ is easier than classes $\mathcal{SMU}_{\delta=1}$ and $\mathcal{MMU}_{\delta=1}$ (which have recursive classification, see Section 4.6) as its classification is non-recursive, and a conjecture is that whether this class has actually efficient isomorphism type determination (Definition 7.3.2).

Conjecture 7.2.1 *The class 2-$\mathcal{MU} \cap \mathcal{MU}^+$ has efficient isomorphism type determination.*

Another fundamental open question is the complexity of determining all MUSs for input 2-CNFs. MUSs of 2-CNFs have been studied in [28], showing how to compute shortest MUSs in polytime. By [27] the problem of deciding whether a 2-CNF contains an MUS with deficiency 1 is NP-complete. We showed that the isomorphism problem for 2-MUs is polytime decidable (Corollary 6.7.3). This result implies that via enumerating all MUSs of F, we can group them by their isomorphism type in polytime. This yields a list of all isomorphism types of MUSs and their counts, which seems very valuable and is a complete representation of MUSs of F.

Furthermore, using the framework of DFM, we provide an approach to enumerate MUSs of an unsatisfiable 2-uniform F with precisely two monotone clauses. Any MUS of such F has the two monotone clauses (otherwise would be satisfiable). In order to use Theorem 7.1.4 we need to enlarge the monotone clauses. Let $G \subset F$ be the set of mixed binary clauses, $P \in F$ be the positive clause and $N \in F$ be the negative clause (i.e., $F = G \cup \{P, N\}$). Obtain $F' := G \cup \{P', N'\}$ where P', N' are the full positive resp. full negative clauses over $\mathrm{var}(F)$. By Theorem 7.1.4 there exists an incremental polynomial time algorithm to enumerate MUSs of F'. Now for any MUS of F', obtain a new clause-set by replacing P', N' (the full monotone clauses) with P, N (the binary monotone clauses in F), and let S be the set of these new clause-sets. It is easy to see that MUSs of F correspond precisely to the minimally unsatisfiable clause-sets in S (note that all elements of S are unsatisfiable but some might be redundant, see Section 4.2). Therefore one further step is needed to check minimal unsatisfiability of 2-CNFs in S. As discussed in Section 1.4.4 the minimal unsatisfiability problem for 2-CNFs can be decided in quadratic time, however whether the whole process is still incremental polynomial time, is an open question.

7.3 The Finite-Patterns Conjecture

As explained in the Introduction (Section 1.4.2), a major motivation of this work has been the project of classifying MUs, where the main open question is a proof of the Finite-Patterns Conjecture as considered in the outlook of [109].

Considering classification of classes $\mathcal{MU}_{\delta=k}$ with fixed deficiency $k \geq 1$, [81] showed the necessity to consider some form of reduction for these classes, as the isomorphism problem is GI-complete. A "harmless" reduction for MUs in general is singular DP-reduction, since minimal unsatisfiability and deficiency are maintained (Lemma 4.5.2). A weaker form of the Finite-Patterns Conjecture is that isomorphism for the nonsingular elements of $\mathcal{MU}_{\delta=k}$ is feasibly decidable:

Conjecture 7.3.1 *For all $k \in \mathbb{N}$, $\mathcal{MU}'_{\delta=k}$ has polytime isomorphism decision.*

Although Conjecture 7.3.1 means that we can understand the general "shape" of the class, but if it has super-polynomially many isomorphism types then we still do not have precise knowledge. There are several complexity levels for the isomorphism problem which are each interesting. However we state the simplest case of complexity level where we have already seen some examples (see Table 7.1 for an overview of these examples):

Definition 7.3.2 *A class $\mathcal{C} \subseteq \mathcal{CLS}$ has an* ***efficient isomorphism type determination (EID)*** *if there exists a surjective map $f : \mathcal{C} \to \mathbb{N}$ with the following properties:*

1. *f is computable in polynomial time;*
2. *for $F, G \in \mathcal{C}$ holds: $F \cong G \Leftrightarrow f(F) = f(G)$ (that is, f is a complete invariant for $\mathcal{C}$);*
3. *for $m \in \mathbb{N}$ we can compute in polytime some F with $f(F) = m$ (m in unary notation).*

We note that in general there are two possible notations (or encodings) for m in Definition 7.3.2, namely unary notation (length of encoding is m) and binary notation (length of encoding is $\log_2 m$). Since for practical reasons we really want to explicitly list clause-sets (for example as inputs for SAT-solvers), we need to use unary notation for m as otherwise complexity of computing some F with $f(F) = m$ would be exponential. As seen before for classes $\mathcal{MU}'_{\delta=1} = \{\{\perp\}\}$ and $\mathcal{MU}'_{\delta=2}$ we have EID, where for $F \in \mathcal{MU}'_{\delta=2}$ the parameter m is the number of variable $n(F)$ (Corollary 5.3.2). Furthermore some preliminary investigation into $\mathcal{MU}'_{\delta=3}$ in [110] shows that we might have EID for this class. So strengthening Conjecture 7.3.1, the Finite-Patterns Conjecture takes the following specific form:

Conjecture 7.3.3 *For fixed $k \in \mathbb{N}$ the classes $\mathcal{MU}'_{\delta=k}$ have EID.*

In Conjecture 7.3.3 it is desirable to have fixed-parameter tractability of classes $\mathcal{MU}'_{\delta=k}$ in k as otherwise run-time of computing of f, f^{-1} (in Definition 7.3.2) might grow exponentially in k (in which case we could only handle small k).

Question 7.3.4 *Consider Conjecture 7.3.3 for classes $\mathcal{MU}'_{\delta=k}$ with fixed $k \in \mathbb{N}$. Are the maps f and their inverses in Definition 7.3.2 fixed-parameter tractable (FPT) in k? That is, whether computation of f, f^{-1} has the complexity $\beta(k) \cdot n^C$ for input size n, some constant C and a function $\beta(k)$ (which likely is exponential in k).*

A variation on Conjecture 7.3.3 is to consider the classes $\mathcal{SMU}_{\delta=k}$, where a common weakening is to consider the classes $\mathcal{SMU}'_{\delta=k}$. Once the isomorphism types of $\mathcal{SMU}'_{\delta=k}$ have been determined, then we can speak of the "nonsingularity type" of an arbitrary $\mathcal{SMU}_{\delta=k}$, since singular DP-reduction is confluent for saturated clause-sets (Lemma 4.5.5). Then from $\mathcal{SMU}'_{\delta=k}$ we obtain $\mathcal{MU}'_{\delta=k}$ via

Table 7.1: An overview of main characterisations of MUs. See Appendix A.1 for a list of classes and their definition, and Appendix A.3 for definitions of MUs $\mathcal{F}_n, \mathcal{B}_n, \mathrm{S}_n, \mathrm{U}^2_n, \mathrm{U}^1_{n,i}, \mathrm{U}^0_{n,i}, \mathrm{U}^0_{n,x,y}$.

Classes	Complexity	Characterisations
$\mathcal{MU}'_{\delta=1}$	EID	$\{\{\perp\}\}$ (Corollary 4.6.10)
$\mathcal{SMU}_{\delta=1}$	Polytime	Full binary trees (Lemma 4.6.3)
$\mathcal{MMU}_{\delta=1}$	Polytime	Trees (Lemma 4.6.38)
$\mathcal{MU}'_{\delta=2}$	EID	$\mathcal{F}_n$, as graphs, correspond to cycles (Corollary 5.3.2)
$\mathcal{UHIT}'_{\delta=1}$	EID	$= \mathcal{MU}'_{\delta=1}$ (Corollary 4.6.10)
$\mathcal{UHIT}'_{\delta=2}$	EID	$\mathcal{F}_2, \mathcal{F}_3$ (Corollary 5.3.3)
$\mathcal{UHIT}_{\delta=1}$	Polytime	$= \mathcal{SMU}_{\delta=1}$ (Corollary 4.6.10)
2–$\mathcal{MU}'$	EID	$\mathcal{B}_n$, as graphs, correspond to dipaths (Corollary 5.4.8)
2–$\mathcal{MU}_{\delta=1}$	EID	$\mathrm{U}^2_n, \mathrm{U}^1_{n,i}, \mathrm{U}^0_{n,i}, \mathrm{U}^0_{n,x,y}$ (Theorem 6.3.10)
2–$\mathcal{MU} \cap \mathcal{MU}^+$	Polytime	Binary bracelets (Corollary 6.7.5)
2–$\mathcal{MU}_{\delta=k}$, $k \geq 2$	Polytime	$2k$-WDCs (Corollary 6.7.1)
$\mathcal{RHO} \cap \mathcal{SMU}$	EID	S_n (Lemma 4.6.22)
$\mathcal{RHO} \cap$ 2–$\mathcal{MU}$	EID	$\mathrm{U}^2_n, \mathrm{U}^1_{n,i}$ (Lemma 5.4.1 and Theorem 6.3.10)
$\mathcal{DFC}_{\delta=2}$ (so $\mathcal{DFM}_{\delta=2}$)	EID	$= \mathcal{MU}'_{\delta=2}$ (Theorem 5.3.1)
$\mathcal{FC}_{\delta=2}$ (so $\mathcal{FM}_{\delta=2}$)	EID	$= \mathcal{MU}'_{\delta=2}$ (Theorem 5.3.1)

partial marginalisation. An alternative approach would be to consider marginal instances. Then we would use partial saturation instead of partial saturation. However, it seems that nonsingularity can not be used here, since partial saturation can make a singular instance nonsingular (and thus from the marginal nonsingular instances we can not obtain all nonsingular instances).

Another major step towards Conjecture 7.3.3 is the classification of unsatisfiable nonsingular hitting clause-sets in dependency on the deficiency, i.e., determining the elements of $\mathcal{UHIT}'_{\delta=k}$ which are saturated minimally unsatisfiable (Lemma 4.1.9), and thus have deficiency at least 1. In [106, Conjecture 25] it is conjectured that there are only finitely many isomorphism types of $F \in \mathcal{UHIT}'_{\delta=k}$ with fixed $k \geq 1$. This conjecture has been shown for $k \leq 3$ in [110], leaving open the determination of the isomorphism types of $\mathcal{UHIT}_{\delta=3}$. While in [88] we find a catalogue of the elements of $\mathcal{UHIT}'_{\delta=3}$ via `all_uhit_def(3)` (`Maxima` in the `OKlibrary` [100]).

Finally another special case of the Finite-Patterns Conjecture 7.3.3 is that

FMs have EID:

Conjecture 7.3.5 *For fixed $k \geq 2$ the classes $\mathcal{FM}_{\delta=k}$ have EID.*

We expect the class $\mathcal{FM}_{\delta=k}$ at least for $\delta = 3$ to be a stepping stone towards understanding $\mathcal{MU}_{\delta=3}$ (the current main frontier).

A special case of Conjecture 7.3.5 is to have EID for DFMs:

Conjecture 7.3.6 *For fixed $k \geq 2$ the classes $\mathcal{DFM}_{\delta=k}$ have EID.*

Conjecture 7.3.6 should be provable by showing that for fixed k the number of cycles in MSDs of deficiency k is bounded.

7.4 Connections to graph theory

To use the strength of graph-theoretical reasoning, several concepts have been used/introduced in this thesis to connect clause-sets to graph theory. Table 7.2 provides an overview of these concepts and their applications to characterise various classes of MUs.

An extension of these relations is concerning the conflict patterns of clauses. Definition 4.6.31 of the conflict graph $\mathrm{cg}(F)$ for $F \in \mathcal{CLS}$ can be extended to **conflict multigraph cmg(F)**, to allow parallel edges (see [109] for an overview). Conflict multigraphs do not exist for deficiency 1 (recall Corollary 4.6.14), while for higher deficiencies k conflict graph and conflict multigraphs can be used to obtain better understanding of classes $\mathcal{MU}_{\delta=k}$ and $\mathcal{UHIT}_{\delta=k}$ (at least the nonsingular versions). For example for $F \in \mathcal{MU}_{\delta=2}$ we know $\mathrm{cg}(F)$ has at least four vertices (and is connected, as is every $\mathrm{cg}(F)$ for $F \in \mathcal{MU}$). Furthermore $\mathrm{cg}(\mathcal{F}_2)$ is isomorphic to the complete graph with four vertices. For $n \geq 3$, $\mathrm{cg}(\mathcal{F}_n)$ is isomorphic to the cycle digraph CD_n with two added universal vertices, which one might call a full wheel (where there is also an edge between the two universal vertices). Now a question is that whether some $\mathrm{cg}(F)$ can be a tree? Perhaps trees are only possible for deficiency 1 ?

Finally we remark that we did not tackle the graph isomorphism problem for graphs and digraphs, however we showed that still some classes of MUs can yield some insights into the structure of graphs/digraphs. For example using the deficiency of (di)graphs (Definition 3.1.2), Conjecture 7.3.6 is equivalent to the following conjecture:

Conjecture 7.4.1 *MSDs have efficient isomorphism type determination, that is, there is a surjective map $f : \mathrm{MSD} \to \mathbb{N}$, such that two MSDs G_1, G_2 are isomorphic iff $f(G_1) = f(G_2)$, and where computation of f as well as computing some choice for f^{-1} (here input size measured in unary) can be done in polynomial time.*

This conjecture has been proved for deficiency zero in Lemma 5.2.11 (cycle digraphs). In [59] we find some results concerning enumeration of isomorphism types with numerical results on the OEIS ([139, Sequence A130756]) and [60].

Table 7.2: An overview of some connections between clause-sets and graph theory, and their applications in this thesis. $\mathcal{DFM}$ is the set of MUs which are mixed binary clauses plus both full monotone clauses, $2\text{–}\mathcal{CLS}^*$ is the set of 2-CNFs excluding $\{\bot\}$, and $2\text{–}\mathcal{MU}$ is the set of 2-CNF MUs. Also $\mathcal{SMU}_{\delta=1}$ is the set of saturated MUs of deficiency 1.

Relation to clause-sets	Classes	Connection to graph theory for an element F of the class	Main outputs
2-CNFs as implications	$\mathcal{DFM}$	The positive implication digraph $\mathrm{pdg}(F)$ (Definition 5.2.3)	Classification of $\mathcal{MU}'_{\delta=2} = \mathcal{DFC}_{\delta=2} = \mathcal{FC}_{\delta=2}$ (Section 5.3) and $2\text{–}\mathcal{MU}'$ (Section 5.4)
	$2\text{–}\mathcal{CLS}^*$	The (full) implication digraph $\mathrm{idg}(F)$ (Definition 6.2.1)	Classification of 2-MUs with deficiency $k \geq 2$ (Section 6.7)
	$2\text{–}\mathcal{CLS}^*$	The implication graph $\mathrm{ig}(F)$ (Definition 6.2.1)	Classification of $2\text{–}\mathcal{MU}_{\delta=1}$ (Theorem 6.3.10)
	$2\text{–}\mathcal{CLS}^*$	The implication multigraph $\mathrm{img}(F)$ (Section 6.4.2)	Correspondence of 1-singular DP-reduction for 2-MUs and smoothing of their implication graphs (Section 6.4.2)
	$2\text{–}\mathcal{CLS}^*$	The implication digraph with the complementation of literals (as a skew-symmetry) $\mathrm{sidg}(F)$ (Section 6.2)	The isomorphism between 2-MUs are exactly the isomorphism between their implication digraphs (Theorem 6.6.10)
1-singular DP-reduction	$2\text{–}\mathcal{MU}$	The homeomorphism type $\mathrm{sm}(F)$ (Lemma 6.4.7)	Classification of the normalforms of 2-MUs (Corollary 6.7.5)
Splitting/ resolution	$\mathcal{SMU}_{\delta=1}$	The structure tree $T(F)$ (Definition 4.6.3)	Classification of $\mathcal{SMU}_{\delta=1} = \mathcal{UHIT}_{\delta=1}$ (Section 4.6.2)
The conflict patterns of clauses	$\mathcal{CLS}$	The conflict graph $\mathrm{cg}(F)$ (Definition 4.6.31)	Classification of $\mathcal{MMU}_{\delta=1}$ (Section 4.6.4)

In Section 6.5 we obtained a polytime isomorphism decision for WDCs, and characterised their homeomorphism types using graph-theoretical reasoning, while in Section 6.6 we showed the uniqueness of skew-symmetry for WDCs via connecting 2-MUs and WDCs.

Bibliography

[1] Hoda Abbasizanjani and Oliver Kullmann. Minimal unsatisfiability and minimal strongly connected digraphs. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *Theory and Applications of Satisfiability Testing - SAT 2018*, volume 10929 of *Lecture Notes in Computer Science*, pages 329–345. Springer, 2018. doi:10.1007/978-3-319-94144-8_20.

[2] Hoda Abbasizanjani and Oliver Kullmann. Classification of minimally unsatisfiable 2-CNFs. Technical Report arXiv:2003.03639v2 [cs.DM], arXiv, May 2020. URL: http://arxiv.org/abs/2003.03639.

[3] Manindra Agrawal and Thomas Thierauf. The boolean isomorphism problem. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 422–430. IEEE, October 1996. doi:10.1109/SFCS.1996.548501.

[4] Ron Aharoni and Nathan Linial. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *Journal of Combinatorial Theory, Series A*, 43(2):196–204, November 1986. doi:10.1016/0097-3165(86)90060-9.

[5] Edoardo Amaldi, Marc E. Pfetsch, and Leslie E. Trotter, Jr. On the maximum feasible subsystem problem, iiss and iis-hypergraphs. *Mathematical Programming*, 95:533–554, March 2003. doi:10.1007/s10107-002-0363-5.

[6] Carlos Ansótegui, María Luisa Bonet, Jordi Levy, and Felip Manyà. Measuring the hardness of SAT instances. In Dieter Fox and Carla Gomes, editors, *Proceedings of the 23th AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 222–228, 2008. URL: https://www.aaai.org/Library/AAAI/2008/aaai08-035.php.

[7] Bengt Aspvall. Recognizing disguised NR(1) instances of the satisfiability problem. *Journal of Algorithms*, 1:97–103, March 1980. doi:10.1016/0196-6774(80)90007-3.

[8] Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, March 1979. `doi:10.1016/0020-0190(79)90002-4`.

[9] Mustafa K. Atlihan and Linus Schrage. Generalized filtering algorithms for infeasibility analysis. *Computers & Operations Research*, 35(5):1446–1464, May 2008. `doi:10.1016/j.cor.2006.08.005`.

[10] László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 684–697. Association for Computing Machinery, 2016. `doi:10.1145/2897518.2897542`.

[11] Fahiem Bacchus and George Katsirelos. Using minimal correction sets to more efficiently compute minimal unsatisfiable sets. In Daniel Kroening and Corina S. Păsăreanu, editors, *Computer Aided Verification*, pages 70–86, Cham, July 2015. Springer International Publishing. `doi:10.1007/978-3-319-21668-3_5`.

[12] Fahiem Bacchus and George Katsirelos. Finding a collection of muses incrementally. In Claude-Guy Quimper, editor, *Integration of AI and OR Techniques in Constraint Programming*, pages 35–44, Cham, May 2016. Springer International Publishing. `doi:10.1007/978-3-319-33954-2_3`.

[13] Jørgen Bang-Jensen and Gregory Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer, 2009. ISBN 978-1-84800-998-1. `doi:10.1007/978-1-84800-998-1`.

[14] Jaroslav Bendík and Ivana Černá. MUST: Minimal unsatisfiable subsets enumeration tool. In Armin Biere and David Parker, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 135–152, Cham, April 2020. Springer International Publishing. `doi:10.1007/978-3-030-45190-5_8`.

[15] Jaroslav Bendík, Ivana Černá, and Nikola Beneš. Recursive online enumeration of all minimal unsatisfiable subsets. In Shuvendu K. Lahiri and Chao Wang, editors, *Automated Technology for Verification and Analysis*, pages 143–159, Cham, September 2018. Springer International Publishing. `doi:10.1007/978-3-030-01090-4_9`.

[16] Jaroslav Bendík, Elaheh Ghassabani, Michael Whalen, and Ivana Černá. Online enumeration of all minimal inductive validity cores. In Einar Broch Johnsen and Ina Schaefer, editors, *Software Engineering and Formal Methods*, pages 189–204, Cham, May 2018. Springer International Publishing. `doi:10.1007/978-3-319-92970-5_12`.

[17] Jaroslav Bendík and Kuldeep S. Meel. Approximate counting of minimal unsatisfiable subsets. In Shuvendu K. Lahiri and Chao Wang, editors,

Computer Aided Verification, pages 439–462, Cham, July 2020. Springer International Publishing. doi:10.1007/978-3-030-53288-8_21.

[18] Marc A. Berger, Alexander Felzenbaum, and Aviezri S. Fraenkel. Covers of product sets and the Korec-Znám result. *European Journal of Combinatorics*, 9(2):131–138, March 1988. doi:10.1016/S0195-6698(88)80037-4.

[19] Olaf Beyersdorff and Oliver Kullmann. Hardness measures and resolution lower bounds. Technical Report arXiv:1310.7627v2 [cs.CC], arXiv, February 2014. URL: http://arxiv.org/abs/1310.7627.

[20] Olaf Beyersdorff and Oliver Kullmann. Unified characterisations of resolution hardness measures. In Uwe Egly and Carsten Sinz, editors, *Theory and Applications of Satisfiability Testing - SAT 2014*, volume 8561 of *Lecture Notes in Computer Science*, pages 170–187. Springer, 2014. doi:10.1007/978-3-319-09284-3_13.

[21] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, Ofer Strichman, and Y. Zhu. Bounded model checking. In Marvin Zelkowitz, editor, *Highly Dependable Software*, volume 58 of *Advances in Computers*, pages 117–148. Academic Press, 2003. doi:10.1016/S0065-2458(03)58003-2.

[22] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2009. URL: https://www.iospress.nl/book/handbook-of-satisfiability/.

[23] Armin Biere, Marijn J.H. Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability, Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2021. URL: https://www.iospress.com/book/handbook-of-satisfiability-2.

[24] Béla Bollobás, Christian Borgs, Jennifer T. Chayes, Jeong Han Kim, and David B. Wilson. The scaling window of the 2-SAT transition. *Random Structures and Algorithms*, 18(3):201–256, April 2001. doi:10.1002/rsa.1006.

[25] Emanuele Brugnoli. Enumerating the Walecki-type Hamiltonian cycle systems. *Journal of Combinatorial Designs*, 25:481–493, May 2017. doi:10.1002/jcd.21558.

[26] Renato Bruni. On exact selection of minimally unsatisfiable subformulae. *Annals for Mathematics and Artificial Intelligence*, 43:35–50, December 2005. doi:10.1007/s10472-005-0418-4.

[27] Hans Kleine Büning, Piotr Wojciechowski, and K. Subramani. On the computational complexity of read once resolution decidability in 2CNF formulas. In T. Gopal, G. Jger, and S. Steila, editors, *International Conference on Theory and Applications of Models of Computation (TAMC*

2017), volume 10185 of *Lecture Notes in Computer Science*, pages 362–372. Springer, 2017. doi:10.1007/978-3-319-55911-7_26.

[28] Joshua Buresh-Oppenheim and David Mitchell. Minimum witnesses for unsatisfiable 2CNFs. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006*, volume 4121 of *Lecture Notes in Computer Science*, pages 42–47. Springer, 2006. doi:10.1007/11814948_6.

[29] Joshua Buresh-Oppenheim and David Mitchell. Minimum 2CNF resolution refutations in polynomial time. In Joao Marques-Silva and Karem A. Sakallah, editors, *Theory and Applications of Satisfiability Testing - SAT 2007*, volume 4501 of *Lecture Notes in Computer Science*, pages 300–313. Springer, 2007. doi:10.1007/978-3-540-72788-0_29.

[30] Sam Buss and Jakob Nordström. Proof complexity and SAT solving. In Biere et al. [23], chapter 7, pages 233–350. URL: https://www.iospress.com/book/handbook-of-satisfiability-2, doi:10.3233/FAIA200990.

[31] Hadrien Cambazard and Barry OSullivan. Reformulating table constraints using functional dependenciesan application to explanation generation. *Constraints*, 13:385–406, April 2008. doi:10.1007/s10601-008-9042-3.

[32] Florent Capelli and Yann Strozecki. On the complexity of enumeration, July 2017. URL: https://arxiv.org/abs/1703.01928.

[33] Chin-Liang Chang. The unit proof and the input proof in theorem proving. *Journal of the Association for Computing Machinery*, 17(4):698–707, October 1970. doi:10.1145/321607.321618.

[34] Pankaj Chauhan, Edmund Clarke, James Kukula, Samir Sapra, Helmut Veith, and Dong Wang. Automated abstraction refinement for model checking large state spaces using SAT based conflict analysis. In *International Conference on Formal Methods in Computer-Aided Design*, 2002. doi:10.1007/3-540-36126-X_3.

[35] Vasek Chvátal and B. Reed. Mick gets some (the odds are on his side). In *Proc. 33th Annual Symposium on Foundations of Computer Science (Pittsburgh, PA)*, pages 620–627. IEEE Comput. Soc. Press, 1992. doi:10.1109/SFCS.1992.267789.

[36] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158, 1971. doi:10.1145/800157.805047.

[37] Stephen A. Cook. An exponential example for analytic tableaux. Manuscript (see [145], page 432), 1973.

[38] Yves Crama and Peter L. Hammer. *Boolean Functions: Theory, Algorithms, and Applications*, volume 142 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, July 2011. ISBN 978-0-521-84751-3. URL: `http://www.cambridge.org/gb/knowledge/isbn/item6222210/?site_locale=en_GB`.

[39] Adnan Darwiche and Knot Pipatsrisawat. Complete algorithms. In Biere et al. [23], chapter 3, pages 101–132. URL: `https://www.iospress.com/book/handbook-of-satisfiability-2`, `doi:10.3233/FAIA200986`.

[40] Sudeshna Dasgupta and Vijay Chandru. Minimal unsatisfiable sets: Classification and bounds. In Michael J. Maher, editor, *Advances in Computer Science - ASIAN 2004: Higher-Level Decision Making*, volume 3321 of *Lecture Notes in Computer Science*, pages 330–342, 2004. `doi:10.1007/978-3-540-30502-6\_24`.

[41] Jason Davies. Combinatorial necklaces and bracelets, 2012. URL: `https://www.jasondavies.com/necklaces/`.

[42] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, July 1962. `doi:10.1145/368273.368557`.

[43] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, July 1960. `doi:10.1145/321033.321034`.

[44] Gennady Davydov and Inna Davydova. Solvable matrices. *Vestnik St. Petersburg University, Mathematics*, 26(1):1–6, 1993.

[45] Gennady Davydov, Inna Davydova, and Hans Kleine Büning. An efficient algorithm for the minimal unsatisfiability problem for a subclass of CNF. *Annals of Mathematics and Artificial Intelligence*, 23(3-4):229–245, November 1998. `doi:10.1023/A:1018924526592`.

[46] Wenceslas Fernandez de la Vega. Random 2-SAT: results and problems. *Theoretical Computer Science*, 265:131–146, August 2001. `doi:10.1016/S0304-3975(01)00156-6`.

[47] William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984. `doi:10.1016/0743-1066(84)90014-1`.

[48] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. `doi:10.1007/978-1-4612-0515-9`.

[49] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science*, STACS99, pages 551–560, Berlin, Heidelberg, 1999. Springer-Verlag. `doi:10.1007/3-540-49116-3\_52`.

[50] Shimon Even, Alon Itai, and Adi Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal Computing*, 5(4):691–703, 1976. `doi:10.1137/0205048`.

[51] Herbert Fleischner, Oliver Kullmann, and Stefan Szeider. Polynomial–time recognition of minimal unsatisfiable formulas with fixed clause–variable difference. *Theoretical Computer Science*, 289(1):503–516, November 2002. `doi:10.1016/S0304-3975(01)00337-1`.

[52] Herbert Fleischner and Stefan Szeider. Polynomial-time recognition of minimal unsatisfiable formulas with fixed clause-variable difference. Technical Report TR00-049, Electronic Colloquium on Computational Complexity (ECCC), July 2000. URL: `http://eccc.hpi-web.de/eccc-reports/2000/TR00-049/index.html`.

[53] T. Fliti and G. Reynaud. Sizes of minimally unsatisfiable conjunctive normal forms. Faculté des Sciences de Luminy, Dpt. Mathematique-Informatique, 13288 Marseille, France, November 1994.

[54] John Franco and Allen Van Gelder. A perspective on certain polynomial-time solvable classes of satisfiability. *Discrete Applied Mathematics*, 125(2-3):177–214, 2003. `doi:10.1016/S0166-218X(01)00358-4`.

[55] John Franco and John Martin. A history of satisfiability. In Biere et al. [22], chapter 1, pages 3–74. URL: `https://www.iospress.nl/book/handbook-of-satisfiability/`, `doi:10.3233/978-1-58603-929-5-3`.

[56] Zhaohui Fu and Sharad Malik. On solving the partial MAX-SAT problem. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, volume 4121 of *Lecture Notes in Computer Science*, pages 252–265. Springer, 2006. `doi:10.1007/11814948\_25`.

[57] Nicola Galesi and Oliver Kullmann. Polynomial time SAT decision, hypergraph transversals and the hermitian rank. In Holger H. Hoos and David G. Mitchell, editors, *Theory and Applications of Satisfiability Testing 2004*, volume 3542 of *Lecture Notes in Computer Science*, pages 89–104, Berlin, 2005. Springer. `doi:10.1007/11527695_8`.

[58] J García-López, C Marijuán, and Luis Miguel Pozo-Coronado. Structural properties of minimal strong digraphs versus trees. *Linear Algebra and its Applications*, 540:203–220, March 2018. `doi:10.1016/j.laa.2017.11.027`.

[59] Jesús García-López and Carlos Marijuán. Minimal strong digraphs. *Discrete Mathematics*, 312(4):737–744, February 2012. `doi:10.1016/j.disc.2011.11.010`.

[60] Jesús García-López and Carlos Marijuán. Minimal strong digraphs, 2012. URL: `http://jglopez.etsisi.upm.es/MSC_Digraphs_Page/`.

[61] E.N. Gilbert and John Riordan. Symmetry types of periodic sequences. *Illinois Journal of Mathematics*, 5(4):657–665, 1961. `doi:10.1215/ijm/1255631587`.

[62] Enrico Giunchiglia and Armando Tacchella, editors. *Theory and Applications of Satisfiability Testing 2003*, volume 2919 of *Lecture Notes in Computer Science*, Berlin, 2004. Springer. `doi:10.1007/b95238`.

[63] Andreas Goerdt. A threshold for unsatisfiability. *Journal of Computer and System Sciences*, 53(3):469–486, 1996. `doi:10.1006/jcss.1996.0081`.

[64] Andrew V. Goldberg and Alexander V. Karzanov. Path problems in skew-symmetric graphs. *Combinatorica*, 16:353–382, 1996. `doi:10.1007/BF01261321`.

[65] Éric Grégoire, Bertrand Mazure, and Cédric Piette. On approaches to explaining infeasibility of sets of boolean clauses. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, volume 1, pages 74–83, 2008. `doi:10.1109/ICTAI.2008.39`.

[66] Jonathan L. Gross and Jay Yellen, editors. *Handbook of Graph Theory*. Discrete Mathematics and Its Applications. CRC Press, 2003. `doi:10.1201/9780203490204`.

[67] Matthew Gwynne and Oliver Kullmann. Trading inference effort versus size in CNF knowledge compilation. Technical Report arXiv:1310.5746v2 [cs.CC], arXiv, November 2013. URL: `http://arxiv.org/abs/1310.5746`.

[68] Matthew Gwynne and Oliver Kullmann. Generalising unit-refutation completeness and SLUR via nested input resolution. *Journal of Automated Reasoning*, 52(1):31–65, January 2014. `doi:10.1007/s10817-013-9275-8`.

[69] Matthew Gwynne and Oliver Kullmann. On SAT representations of XOR constraints. In Adrian-Horia Dediu, Carlos Martín-Vide, José-Luis Sierra, and Bianca Truthe, editors, *LATA 2014: Language and Automata Theory and Applications, 8th International Conference*, volume 8370 of *Lecture Notes in Computer Science*, pages 409–420. Springer, 2014. `doi:10.1007/978-3-319-04921-2_33`.

[70] Lawrence J. Henschen and Lawrence Wos. Unit refutations and Horn sets. *Journal of the Association for Computing Machinery*, 21(4):590–605, October 1974. `doi:10.1145/321850.321857`.

[71] Robert E Horton. Erosional development of streams and their drainage basins; hydrophysical approach to quantitative morphology. *Bulletin of the Geological Society of America*, 56(3):275–370, 03 1945. `doi:10.1130/0016-7606(1945)56[275:EDOSAT]2.0.CO;2`.

[72] Kazuo Iwama. CNF satisfiability test by counting and polynomial average time. *SIAM Journal on Computing*, 18(2):385–391, April 1989. `doi:10.1137/0218026`.

[73] Ulrich Junker. Quickxplain: Preferred explanations and relaxations for over-constrained problems. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 167–172. AAAI Press, 2004. URL: `http://www.aaai.org/Library/AAAI/2004/aaai04-027.php`.

[74] Vaibhav Karve and Anil N. Hirani. The complete set of minimal simple graphs that support unsatisfiable 2-CNFs. *Discrete Applied Mathematics*, 2020. In press. `doi:10.1016/j.dam.2019.12.017`.

[75] Henry Kautz, Ashish Sabharwal, and Bart Selman. Incomplete algorithms. In Biere et al. [23], chapter 6, pages 213–232. URL: `https://www.iospress.com/book/handbook-of-satisfiability-2`, `doi:10.3233/FAIA200989`.

[76] Leonid Khachiyan, Endre Boros, Khaled Elbassioni, and Vladimir Gurvich. On enumerating minimal dicuts and strongly connected subgraphs. *Algorithmica*, 50:159–172, January 2008. `doi:10.1007/s00453-007-9074-x`.

[77] Hans Kleine Büning. An upper bound for minimal resolution refutations. In Georg Gottlob, Etienne Grandjean, and Katrin Seyr, editors, *Computer Science Logic 12th International Workshop, CSL'98*, volume 1584 of *Lecture Notes in Computer Science*, pages 171–178. Springer, 1999. `doi:10.1007/10703163_12`.

[78] Hans Kleine Büning. On subclasses of minimal unsatisfiable formulas. *Discrete Applied Mathematics*, 107(1-3):83–98, 2000. `doi:10.1016/S0166-218X(00)00245-6`.

[79] Hans Kleine Büning and Oliver Kullmann. Minimal unsatisfiability and autarkies. In Biere et al. [22], chapter 11, pages 339–401. URL: `https://www.iospress.nl/book/handbook-of-satisfiability/`, `doi:10.3233/978-1-58603-929-5-339`.

[80] Hans Kleine Büning, Piotr J. Wojciechowski, and K. Subramani. Finding read-once resolution refutations in systems of 2cnf clauses. *Theor. Comput. Sci.*, 729:42–56, 2018. `doi:10.1016/j.tcs.2018.02.002`.

[81] Hans Kleine Büning and Daoyun Xu. The complexity of homomorphisms and renamings for minimal unsatisfiable formulas. *Annals of*

Mathematics and Artificial Intelligence, 43(1-4):113–127, 2005. doi:10.1007/s10472-005-0422-8.

[82] Hans Kleine Büning and Xishun Zhao. Minimal unsatisfiability: Results and open questions. Technical Report tr-ri-02-230, Series Computer Science, University of Paderborn, University of Paderborn, Department of Mathematics and Computer Science, 2002. http://wwwcs.uni-paderborn.de/cs/ag-klbue/de/research/MinUnsat/index.html.

[83] Hans Kleine Büning and Xishun Zhao. On the structure of some classes of minimal unsatisfiable formulas. *Discrete Applied Mathematics*, 130(2):185–207, 2003. doi:10.1016/S0166-218X(02)00405-5.

[84] Hans Kleine Büning and Xishun Zhao. Extension and equivalence problems for clause minimal formulae. *Annals of Mathematics and Artificial Intelligence*, 43(1):295–306, 2005. doi:10.1007/s10472-005-0435-3.

[85] Hans Kleine Büning and Xishun Zhao. The complexity of some subclasses of minimal unsatisfiable formulas. *Journal on Satisfiability, Boolean Modeling and Computation*, 3:1–17, 2007. doi:10.3233/SAT190026.

[86] Robert A. Kowalski. Logic programming. In Jörg H. Siekmann, editor, *Computational Logic*, volume 9 of *Handbook of the History of Logic*, pages 523–569. Elsevier, 2014. URL: https://doi.org/10.1016/B978-0-444-51624-4.50012-5, doi:10.1016/B978-0-444-51624-4.50012-5.

[87] Melven R. Krom. The decision problem for a class of first-order formulas in which all disjunctions are binary. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 13(8):15–20, 1967. doi:10.1002/malq.19670130104.

[88] Oliver Kullmann. Classification of unsatisfiable hitting clause-sets in dependency on the deficiency. URL: https://github.com/OKullmann/oklibrary/blob/master/ComputerAlgebra/Satisfiability/Lisp/MinimalUnsatisfiability/data/uhit_def.mac.

[89] Oliver Kullmann. Obere und untere Schranken für die Komplexität von aussagenlogischen Resolutionsbeweisen und Klassen von SAT-Algorithmen. Master's thesis, Johann Wolfgang Goethe-Universität Frankfurt am Main, April 1992. (Upper and lower bounds for the complexity of propositional resolution proofs and classes of SAT algorithms (in German); Diplomarbeit am Fachbereich Mathematik).

[90] Oliver Kullmann. Investigating a general hierarchy of polynomially decidable classes of CNF's based on short tree-like resolution proofs. Technical Report TR99-041, Electronic Colloquium on Computational Complexity (ECCC), October 1999. URL: http://eccc.hpi-web.de/report/1999/041/.

[91] Oliver Kullmann. An application of matroid theory to the SAT problem. In *Proceedings of the 15th Annual IEEE Conference on Computational Complexity*, pages 116–124, July 2000. doi:10.1109/CCC.2000.856741.

[92] Oliver Kullmann. Lean clause-sets: Generalizations of minimally unsatisfiable clause-sets. *Discrete Applied Mathematics*, 130:209–249, 2003. doi:10.1016/S0166-218X(02)00406-7.

[93] Oliver Kullmann. On the conflict matrix of clause-sets. Technical Report CSR 7-2003, University of Wales Swansea, Computer Science Report Series, March 2003. URL: http://www.cs.swansea.ac.uk/reports/yr2003/CSR7-2003.pdf.

[94] Oliver Kullmann. The combinatorics of conflicts between clauses. In Giunchiglia and Tacchella [62], pages 426–440. doi:10.1007/978-3-540-24605-3_32.

[95] Oliver Kullmann. Conflict matrices and multi-hitting clause-sets. Extended Abstract for the Guangzhou Symposium on Satisfiability and its Applications, September 2004.

[96] Oliver Kullmann. Upper and lower bounds on the complexity of generalised resolution and generalised constraint satisfaction problems. *Annals of Mathematics and Artificial Intelligence*, 40(3-4):303–352, March 2004. doi:10.1023/B:AMAI.0000012871.08577.0b.

[97] Oliver Kullmann. Constraint satisfaction problems in clausal form: Autarkies and minimal unsatisfiability. Technical Report TR07-055, Electronic Colloquium on Computational Complexity (ECCC), June 2007. URL: http://eccc.hpi-web.de/report/2007/055/.

[98] Oliver Kullmann. Constraint satisfaction problems in clausal form: Autarkies, deficiency and minimal unsatisfiability. Technical Report TR07-055, revision 1, Electronic Colloquium on Computational Complexity (ECCC), November 2008. URL: http://eccc.hpi-web.de/report/2007/055/.

[99] Oliver Kullmann. Present and future of practical SAT solving. In Nadia Creignou, Phokion Kolaitis, and Heribert Vollmer, editors, *Complexity of Constraints: An Overview of Current Research Themes*, volume 5250 of *Lecture Notes in Computer Science*, pages 283–319. Springer, 2008. doi:10.1007/978-3-540-92800-3_11.

[100] Oliver Kullmann. The OKlibrary: Introducing a “holistic” research platform for (generalised) SAT solving. *Studies in Logic*, 2(1):20–53, 2009.

[101] Oliver Kullmann. Constraint satisfaction problems in clausal form I: Autarkies and deficiency. *Fundamenta Informaticae*, 109(1):27–81, 2011. doi:10.3233/FI-2011-428.

[102] Oliver Kullmann. Constraint satisfaction problems in clausal form II: Minimal unsatisfiability and conflict structure. *Fundamenta Informaticae*, 109(1):83–119, 2011. doi:10.3233/FI-2011-429.

[103] Oliver Kullmann. The conflict matrix of (multi-)clause-sets — a link between combinatorics and (generalised) satisfiability problems. In preparation; continuation of [93], 2014.

[104] Oliver Kullmann and Horst Luckhardt. Deciding propositional tautologies: Algorithms and their complexity. Preprint, 82 pages, January 1997. URL: http://cs.swan.ac.uk/~csoliver/papers.html#Pre199799.

[105] Oliver Kullmann and Horst Luckhardt. Algorithms for SAT/TAUT decision based on various measures. Preprint, 71 pages, February 1999. URL: http://cs.swan.ac.uk/~csoliver/papers.html#Pre199799.

[106] Oliver Kullmann and Xishun Zhao. On variables with few occurrences in conjunctive normal forms. In Laurent Simon and Karem Sakallah, editors, *Theory and Applications of Satisfiability Testing - SAT 2011*, volume 6695 of *Lecture Notes in Computer Science*, pages 33–46. Springer, 2011. doi:10.1007/978-3-642-21581-0_5.

[107] Oliver Kullmann and Xishun Zhao. On Davis-Putnam reductions for minimally unsatisfiable clause-sets. In Alessandro Cimatti and Roberto Sebastiani, editors, *Theory and Applications of Satisfiability Testing - SAT 2012*, volume 7317 of *Lecture Notes in Computer Science*, pages 270–283. Springer, 2012. doi:10.1007/978-3-642-31612-8_21.

[108] Oliver Kullmann and Xishun Zhao. On Davis-Putnam reductions for minimally unsatisfiable clause-sets. *Theoretical Computer Science*, 492:70–87, June 2013. doi:10.1016/j.tcs.2013.04.020.

[109] Oliver Kullmann and Xishun Zhao. Bounds for variables with few occurrences in conjunctive normal forms. Technical Report arXiv:1408.0629v5 [math.CO], arXiv, January 2017. URL: http://arxiv.org/abs/1408.0629.

[110] Oliver Kullmann and Xishun Zhao. Unsatisfiable hitting clause-sets with three more clauses than variables. Technical Report arXiv:1604.01288v1 [cs.DM], arXiv, April 2016. URL: http://arxiv.org/abs/1604.01288.

[111] B. V. Landau. An asymptotic expansion for the wedderburnetherington sequence. *Mathematika*, 24(2):262–265, 1977. doi:10.1112/S0025579300009177.

[112] Choongbum Lee. On the size of minimal unsatisfiable formulas. *The Electronic Journal of Combinatorics*, 16(1), 2009. URL: http://www.combinatorics.org/Volume_16/Abstracts/v16i1n3.html.

[113] Leonid Anatolèvich Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973. English translation in Problems of Information Transmission, 9(3), 1973, 265–266. URL: https://zbmath.org/?q=an:0313.02026.

[114] Harry R. Lewis. Renaming a set of clauses as a horn set. *J. ACM*, 25(1):134–135, January 1978. URL: https://doi.org/10.1145/322047.322059, doi:10.1145/322047.322059.

[115] Paolo Liberatore. Redundancy in logic I: CNF propositional formulae. *Artificial Intelligence*, 163(2):203–232, April 2005. doi:10.1016/j.artint.2004.11.002.

[116] Paolo Liberatore. Redundancy in logic II: 2CNF and Horn propositional formulae. *Artificial Intelligence*, 172(2-3):265–299, February 2008. doi:10.1016/j.artint.2007.06.003.

[117] Mark H. Liffiton and Ammar Malik. Enumerating infeasibility: Finding multiple MUSes quickly. In Carla Gomes and Meinolf Sellmann, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 7874 of *Lecture Notes in Computer Science*, pages 160–175, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. doi:10.1007/978-3-642-38171-3_11.

[118] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2):223–250, April 2016. doi:10.1007/s10601-015-9183-0.

[119] Mark H. Liffiton and Karem A. Sakallah. On finding all minimally unsatisfiable subformulas. In Fahiem Bacchus and Toby Walsh, editors, *Theory and Applications of Satisfiability Testing 2005*, volume 3569 of *Lecture Notes in Computer Science*, pages 173–186, Berlin, 2005. Springer. doi:10.1007/11499107_13.

[120] Mark H. Liffiton and Karem A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning*, 40:1–33, 2008. doi:10.1007/s10817-007-9084-z.

[121] 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, August 1982. doi:10.1016/0022-0000(82)90009-5.

[122] Vasco Manquinho, Joao Marques-Silva, and Jordi Planes. Algorithms for weighted boolean optimization. In Oliver Kullmann, editor, *Theory and Applications of Satisfiability Testing - SAT 2009*, volume 5584 of *Lecture Notes in Computer Science*, pages 495–508, Berlin, Heidelberg, 2009. Springer. doi:10.1007/978-3-642-02777-2_45.

[123] Joao Marques-Silva. Computing minimally unsatisfiable subformulas: State of the art and future directions. *Journal of Multiple-Valued Logic and Soft Computing*, 19(1-3):163–183, 2012. URL: `http://www.oldcitypublishing.com/journals/mvlsc-home/mvlsc-issue-contents/mvlsc-volume-19-number-1-3-2012/mvlsc-19-1-3-p-163-183/`.

[124] Kenneth L. McMillan. Interpolation and SAT-based model checking. In *International Conference on Computer-Aided Verification*, volume 2725 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2003. `doi:10.1007/978-3-540-45069-6\_1`.

[125] Carlos Mencía, Oliver Kullmann, Alexey Ignatiev, and Joao Marques-Silva. On computing the union of MUSes. In Mikoláš Janota and Inês Lynce, editors, *Theory and Applications of Satisfiability Testing – SAT 2019*, volume 11628 of *Lecture Notes in Computer Science*, pages 211–221, Cham, 2019. Springer International Publishing. `doi:10.1007/978-3-030-24258-9\_15`.

[126] Gi-Joon Nam, Karem A. Sakallah, and Rob A. Rutenbar. Satisfiability-based layout revisited: Detailed routing of complex FPGAs via search-based boolean SAT. In *Proceedings of the 1999 ACM/SIGDA Seventh International Symposium on Field Programmable Gate Arrays*, page 167175. Association for Computing Machinery, February 1999. `doi:10.1145/296399.296450`.

[127] Nina Narodytska, Nikolaj Björner, Maria-Cristina Marinescu, and Mooly Sagiv. Core-guided minimal correction set and core enumeration. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 1353–1361. International Joint Conferences on Artificial Intelligence Organization, 7 2018. `doi:10.24963/ijcai.2018/188`.

[128] Yoonna Oh, Maher N. Mneimneh, Zaher S. Andraus, Karem A. Sakallah, and Igor L. Markov. AMUSE: a minimally-unsatisfiable subformula extractor. In *Proceedings of the 41st annual conference on Design automation (DAC)*, pages 518–523, 2004. `doi:10.1145/996566.996710`.

[129] Richard Otter. The number of trees. *Annals of Mathematics*, 49(3):583–599, July 1948. `doi:10.2307/1969046`.

[130] Christos H. Papadimitriou and David Wolfe. The complexity of facets resolved. *Journal of Computer and System Sciences*, 37(1):2–13, August 1988. `doi:10.1016/0022-0000(88)90042-6`.

[131] Christos H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal on Computer and System Sciences*, 28(2):244–259, April 1984. `doi:10.1016/0022-0000(84)90068-0`.

[132] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965. `doi:10.1145/321250.321253`.

[133] Joe Sawada. Generating bracelets in constant amortized time. *SIAM J. Comput.*, 31(1):259–268, January 2002. `doi:10.1137/S0097539700377037`.

[134] Dominik Scheder. Unsatisfiable CNF formulas contain many conflicts. In Leizhen Cai, Siu-Wing Cheng, and Tak-Wah Lam, editors, *Algorithms and Computation — 24th International Symposium, ISAAC 2013*, volume 8283 of *Lecture Notes in Computer Science*, pages 273–283. Springer, 2013. `doi:10.1007/978-3-642-45030-3_26`.

[135] Dominik Scheder and Philipp Zumstein. How many conflicts does it need to be unsatisfiable? In Hans Kleine Büning and Xishun Zhao, editors, *Theory and Applications of Satisfiability Testing - SAT 2008*, volume 4996 of *Lecture Notes in Computer Science*, pages 246–256. Springer, 2008. `doi:10.1007/978-3-540-79719-7_23`.

[136] Paul Seymour and Carsten Thomassen. Characterization of even directed graphs. *Journal of Combinatorial Theory Series B*, 42(1):36–45, 1987. `doi:10.1016/0095-8956(87)90061-X`.

[137] Ma Shaohan and Liang Dongmin. A polynomial-time algorithm for reducing the number of variables in MAX SAT problem. *Science in China (Series E)*, 40(3):301–311, June 1997. `doi:10.1007/BF02916605`.

[138] Neil J.A. Sloane. Illustration of initial terms: unlabeled full binary rooted trees. URL: `https://oeis.org/A001190/a001190.gif`.

[139] Neil J.A. Sloane. The On-Line Encyclopedia of Integer Sequences (OEIS), 2008. URL: `http://oeis.org/`.

[140] Arthur N Strahler. Hypsometric (area-altitude) analysis of erosional topography. *GSA Bulletin*, 63(11):1117–1142, November 1952. `doi:10.1130/0016-7606(1952)63[1117:HAAOET]2.0.CO;2`.

[141] Yann Strozecki. *Enumeration complexity and matroid decomposition*. PhD thesis, Université Paris Diderot - Paris 7, 2010.

[142] Stefan Szeider. Minimal unsatisfiable formulas with bounded clause-variable difference are fixed-parameter tractable. *Journal of Computer and System Sciences*, 69(4):656–674, December 2004. `doi:10.1016/j.jcss.2004.04.009`.

[143] Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1:146–160, 1972. `doi:10.1137/0201010`.

[144] Brendan McKay (`https://mathoverflow.net/users/9025/brendan-mckay`). Answer: Graph isomorphism problem for minimally strongly connected digraphs. MathOverflow, August 2017. URL:https://mathoverflow.net/q/279299 (version: 2017-08-23); question asked by Hoda Abbasizanjani (Swansea University). URL: `https://mathoverflow.net/a/279386`.

[145] Alasdair Urquhart. The complexity of propositional proofs. *The Bulletin of Symbolic Logic*, 1(4):425–467, 1995. `doi:10.2307/421131`.

[146] Zhao Xishun and Ding Decheng. Two tractable subclasses of minimal unsatisfiable formulas. *Science in China Series A: Mathematics*, 42(7):720–731, July 1999. `doi:10.1007/BF02878991`.

[147] Lintao Zhang and Sharad Malik. Extracting small unsatisfiable cores from unsatisfiable Boolean formula. In Giunchiglia and Tacchella [62], pages 239–249. `doi:10.1007/b95238`.

Appendix A

Overview of the notations and acronyms

A.1 Classes of clause-sets

Class name	Description
$\mathcal{CLS}$	The set of all clause-sets
2–$\mathcal{CLS}$	The set of all clause-sets with clauses of size at most two
2–$\mathcal{CLS}^*$	$\{F \in 2\text{–}\mathcal{CLS} : \perp \notin F\}$
$\mathcal{SAT}$	The set of all satisfiable clause-sets
$\mathcal{USAT}$	The set of all unsatisfiable clause-sets
$\mathcal{MU}$	The set of all minimally unsatisfiable clause-sets (MUs)
$\mathcal{MU}_{\delta=k}$	The set of all MUs with fixed deficiency $k \geq 1$
$\mathcal{MU}'$	The set of all nonsingular MUs
$\mathcal{MU}^+$	The set of all non-1-singular MUs
$\mathcal{MU}'_{\delta=k}$	The set of all nonsingular MUs with fixed deficiency $k \geq 1$, i.e., $\mathcal{MU}_{\delta=k} \cap \mathcal{MU}'$
$\mathcal{SMU}$	The set of all saturated MUs
$\mathcal{SMU}_{\delta=k}$	The set of all saturated MUs with fixed deficiency $k \geq 1$, i.e., $\mathcal{SMU} \cap \mathcal{MU}_{\delta=k}$

$\mathcal{SMU}'$	The set of all nonsingular saturated MUs, i.e., $\mathcal{SMU} \cap \mathcal{MU}'$
$\mathcal{SMU}'_{\delta=k}$	The set of all nonsingular saturated MUs with fixed deficiency $k \geq 1$, i.e., $\mathcal{SMU} \cap \mathcal{MU}'_{\delta=k}$
$\mathcal{MMU}$	The set of all marginal MUs
$\mathcal{MMU}_{\delta=k}$	The set of all marginal MUs with fixed deficiency $k \geq 1$, i.e., $\mathcal{MMU} \cap \mathcal{MU}_{\delta=k}$
$2\text{–}\mathcal{MU}$	The set of all MUs with clauses of size at most two (2-MUs), i.e., $\mathcal{MU} \cap 2\text{–}\mathcal{CLS}$
$2\text{–}\mathcal{MU}_{\delta=k}$	The set of all 2-MUs with fixed deficiency $k \geq 1$, i.e., $2\text{–}\mathcal{MU} \cap \mathcal{MU}_{\delta=k}$
$2\text{–}\mathcal{MU}'$	The set of nonsingular 2-MUs, i.e., $2\text{–}\mathcal{MU} \cap \mathcal{MU}'$
$2\text{–}\mathcal{MU}^*_{\delta=1}$	$2\text{–}\mathcal{MU}_{\delta=1} \setminus \{\{\bot\}\}$
$\mathcal{FM}$	The set of all MUs with two full monotone clauses (FMs)
$\mathcal{FM}'$	The set of all nonsingular FMs, i.e, $\mathcal{FM} \cap \mathcal{MU}'$
$\mathcal{FC}$	The closure of $\mathcal{FM}$ under isomorphism
$\mathcal{FC}'$	The closure of $\mathcal{FM}'$ under isomorphism
$\mathcal{DFM}$	The set of all FMs where the core consists of binary clauses
$\mathcal{DFC}$	The closure of $\mathcal{DFM}$ under isomorphism
$\mathcal{UHIT}$	The set of all unsatisfiable hitting clause-sets
$\mathcal{UHIT}_{\delta=k}$	The set of all unsatisfiable hitting clause-sets with fixed deficiency $k \geq 1$
$\mathcal{UHIT}'_{\delta=k}$	The set of nonsingular unsatisfiable hitting clause-sets with fixed deficiency $k \geq 1$
$\mathcal{HO}$	The set of all Horn clause-sets
$\mathcal{RHO}$	The set of all renamable Horn clause-sets, i.e., closure of $\mathcal{HO}$ under isomorphism

A.2 Notations on clause-sets

$\mathcal{VA}$ is the set of all variables, while $\mathcal{LIT}$ is the set of all literals. $\top$ is the empty clause-set, and $\perp$ is the empty clause. Consider a clause-set $F \in \mathcal{CLS}$.

Notation	Description
$n(F)$	The number of variables in F
$c(F)$	The number of clauses in F
$\ell(F)$	The number of literal occurrences in F
$\delta(F)$	The deficiency of F, i.e., $\delta(F) = c(F) - n(F)$
$\mathrm{var}(F)$	The set of all variables occurring in F
$\mathrm{lit}(F)$	The set of all possible literals over the variables in F
$\mathrm{ld}_F(v)$	The literal degree of v, i.e., the number of clauses of F containing literal v
$\mathrm{vd}_F(v)$	The variable degree of v, i.e., $\mathrm{ld}_F(v) + \mathrm{ld}_F(\overline{v})$
$\mu\mathrm{vd}(F)$	The minimum of the variable degrees over all variables in F
$\mathrm{var}_{\mu\mathrm{vd}}(F)$	The set of variables of minimum variable degree in F
$\mathrm{var}_{\mathrm{s}}(F)$	The set of singular variables in F
$\mathrm{DP}_v(F)$	The DP-reduction of F on a variable v
$\mathrm{sDP}(F)$	The set of all clause-sets obtained by complete singular DP-reduction of F
$1\mathrm{sDP}(F)$	The (unique) non-1-singular MU obtained by complete 1-singular DP-reduction of F

A.3 Special clause-set examples

A_n (defined in Section 2.1) is the full clause-set consisting of the 2^n full clauses over variables $1, \ldots, n$ for $n \in \mathbb{N}_0$. For example:

- $A_0 = \{\perp\}$,
- $A_1 = \{\{1\}, \{-1\}\}$,
- $A_2 = \{\{1,2\}, \{-1,2\}, \{1,-2\}, \{-1,-2\}\}$,

- $A_3 = \{\{1,2,3\},\{1,2,-3\},\{1,-2,3\},\{1,-2,-3\},\{-1,2,3\},\{-1,2,-3\},$ $\{-1,-2,3\},\{-1,-2,-3\}\}$.

For $n \in \mathbb{N}_0$ the saturated Horn-MUs S_n (Definition 4.6.21) are defined as follows:

$\mathrm{S}_n = \{\{1\},\{-1,2\},\ldots,\{-1,\ldots,-(n-1),n\},\{-1,\ldots,-n\}\} \in \mathcal{SMU}_{\delta=1} \cap \mathcal{HO}$.

Initial cases are as follows:

- $\mathrm{S}_0 = A_0 = \{\bot\}$,
- $\mathrm{S}_1 = A_1 = \{\{1\},\{-1\}\}$,
- $\mathrm{S}_2 = \{\{1\},\{-1,2\},\{-1,-2\}\}$,
- $\mathrm{S}_3 = \{\{1\},\{-1,2\},\{-1,-2,3\},\{-1,-2,-3\}\}$.

For $n \in \mathbb{N}$, $n \geq 2$ the $\mathcal{F}_n$ clause-sets (Definition 5.2.12) are as follows:

$$\begin{aligned}\mathcal{F}_n \quad := \quad & \{\{-1,2\},\{-2,3\},\ldots,\{-(n-1),n\},\{-n,1\},\\ & \{1,\ldots,n\},\{-1,\ldots,-n\}\} \in \mathcal{MU}_{\delta=2}.\end{aligned}$$

$\mathcal{F}_n$ clause-sets are DFM and the initial cases are:

- $\mathcal{F}_2 = A_2 = \{\{-1,2\},\{-2,1\},\{1,2\},\{-1,-2\}\}$,
- $\mathcal{F}_3 = \{\{-1,2\},\{-2,3\},\{-3,1\},\{1,2,3\},\{-1,-2,-3\}\}$,
- $\mathcal{F}_4 = \{\{-1,2\},\{-2,3\},\{-3,4\},\{-4,1\},\{1,2,3,4\},\{-1,-2,-3,-4\}\}$.

For $n \geq 2$ the uniform 2-MUs $\mathcal{B}_n$ (Definition 5.2.18) are defined as

$$\begin{aligned}\mathcal{B}_n \quad := \quad & \{\{-1,2\},\{1,-2\},\ldots,\{-(n-1),n\},\{n-1,-n\},\\ & \{1,n\},\{-1,-n\}\} \in \mathcal{MU}_{\delta=n}.\end{aligned}$$

The initial cases of $\mathcal{B}_n$ are

- $\mathcal{B}_n = A_2 = \{\{-1,2\},\{1,-2\},\{1,2\},\{-1,-2\}\}$,
- $\mathcal{B}_3 = \{\{-1,2\},\{1,-2\},\{-2,3\},\{2,-3\},\{1,3\},\{-1,-3\}\}$,
- $\mathcal{B}_4 = \{\{-1,2\},\{1,-2\},\{-2,3\},\{2,-3\},\{-3,4\},\{3,-4\},\{1,4\},$ $\{-1,-4\}\}$.

For $n \in \mathbb{N}_0$, DB_n clause-sets (Definition 5.2.16) are

$$\begin{aligned}\mathrm{DB}_n \quad := \quad & \{\{-1,2\},\{1,-2\},\ldots,\{-(n-1),n\},\{n-1,-n\},\\ & \{1,\ldots,n\},\{-1,\ldots,-n\}\} \in \mathcal{DFM},\end{aligned}$$

with the initial cases as follows:

- $\mathrm{DB}_0 = A_0 = \{\bot\}$,
- $\mathrm{DB}_1 = A_1 = \{\{1\}, \{-1\}\}$,
- $\mathrm{DB}_2 = A_2 = \{\{-1,2\}, \{1,-2\}, \{1,2\}, \{-1,-2\}\}$,
- $\mathrm{DB}_3 = \{\{-1,2\}, \{1,-2\}, \{-2,3\}, \{2,-3\}, \{1,2,3\}, \{-1,-2,-3\}\}$.

The following examples are defined in Section 6.3. For $n \in \mathbb{N}$ the 2-MUs U^2_n are as

$$\mathrm{U}^2_n := \{\{1\}, \{-1,2\}, \dots, \{-(n-1), n\}, \{-n\}\} \in 2\text{-}\mathcal{MU}_{\delta=1}.$$

And the initial cases are:

- $\mathrm{U}^2_1 = \{\{1\}, \{-1\}\}$,
- $\mathrm{U}^2_2 = \{\{1\}, \{-1,2\}, \{-2\}\}$,
- $\mathrm{U}^2_3 = \{\{1\}, \{-1,2\}, \{-2,3\}, \{-3\}\}$,
- $\mathrm{U}^2_4 = \{\{1\}, \{-1,2\}, \{-2,3\}, \{-3,4\}, \{-4\}\}$.

For $n \geq 2$, $1 \leq i \leq n-1$ the 2-MUs $\mathrm{U}^1_{n,i}$ are as

$$\mathrm{U}^1_{n,i} := \big\{\, \{1\}, \{-1,2\}, \dots, \{-(n-1), n\}, \{-n,-i\} \,\big\} \in 2\text{-}\mathcal{MU}_{\delta=1}.$$

Some examples are

- $\mathrm{U}^1_{2,1} = \{\{1\}, \{-1,2\}, \{-2,-1\}\}$,
- $\mathrm{U}^1_{3,2} = \{\{1\}, \{-1,2\}, \{-2,3\}, \{-3,-2\}\}$,
- $\mathrm{U}^1_{4,1} = \{\{1\}, \{-1,2\}, \{-2,3\}, \{-3,4\}, \{-4,-1\}\}$.
- $\mathrm{U}^1_{4,3} = \{\{1\}, \{-1,2\}, \{-2,3\}, \{-3,4\}, \{-4,-3\}\}$.

For $n \geq 3$, $2 \leq i \leq \frac{n+1}{2}$ the uniform 2-MUs $\mathrm{U}^0_{n,i}$ are defined as

$$\mathrm{U}^0_{n,i} := \big\{\, \{1,i\}, \{-1,2\}, \dots, \{-(n-1), n\}, \{-n,-i\} \,\big\} \in 2\text{-}\mathcal{MU}_{\delta=1}.$$

For example

- $\mathrm{U}^0_{3,2} = \{\{1,2\}, \{-1,2\}, \{-2,3\}, \{-3,-2\}\}$,
- $\mathrm{U}^0_{4,2} = \{\{1,2\}, \{-1,2\}, \{-2,3\}, \{-3,4\}, \{-4,-2\}\}$.
- $\mathrm{U}^0_{5,3} = \{\{1,3\}, \{-1,2\}, \{-2,3\}, \{-3,4\}, \{-4,5\}, \{-5,-3\}\}$.

Finally for $n \geq 4$, $2 \leq x < y \leq n-1$, $x + y \leq n+1$, the uniform 2-MUs $\mathrm{U}^0_{n,x,y}$ are defined as

$$\mathrm{U}^0_{n,x,y} := \{\{1,x\}, \{-1,2\}, \dots, \{-(n-1), n\}, \{-n,-y\}\} \in 2\text{-}\mathcal{MU}_{\delta=1}.$$

Some examples of $\mathrm{U}^0_{n,x,y}$ are as follows:

- $U^0_{4,2,3} = \{\{1,2\},\{-1,2\},\{-2,3\},\{-3,4\},\{-4,-3\}\}$,
- $U^0_{5,2,3} = \{\{1,2\},\{-1,2\},\{-2,3\},\{-3,4\},\{-4,5\},\{-5,3\}\}$,
- $U^0_{5,2,4} = \{\{1,2\},\{-1,2\},\{-2,3\},\{-3,4\},\{-4,5\},\{-5,-4\}\}$,
- $U^0_{6,2,3} = \{\{1,2\},\{-1,2\},\{-2,3\},\{-3,4\},\{-4,5\},\{-5,6\},\{-6,-3\}\}$.

A.4 Acronyms and abbreviations

Acronym	Description
SAT	The propositional satisfiability problem
CNF	Conjunctive Normal Form
DNF	Disjunctive Normal Form
MUS	Minimally Unsatisfiable Sub-clause-sets
MU	Minimally Unsatisfiable clause-set
2-MU	Minimally unsatisfiable 2-CNF
RHO-MU	Minimally unsatisfiable renamable Horn clause-set
FM	MU with Full Monotone clauses
DFM	FM whose core contains only binary clauses
EID	Efficient Isomorphism type Determination
FPT	Fixed-Parameter Tractable
SD	Strong Digraph
MSD	Minimal Strong Digraph
WDC	Weak-Double-Cycle