PRIFYSGOL CYMRU ABERTAWE
UNIVERSITY OF WALES SWANSEA

UNIVERSITY OF WALES SWANSEA

REPORT SERIES

Conjunctive normal forms with non-boolean variables, autarkies and hypergraph colouring

by

Oliver Kullmann

Report # CSR 5-2005

Conjunctive normal forms with non-boolean variables, autarkies and hypergraph colouring

Oliver Kullmann*

Computer Science Department
University of Wales Swansea
Swansea, SA2 8PP, UK

e-mail: O.Kullmann@Swansea.ac.uk
http://cs-svr1.swan.ac.uk/~csoliver/

March 30, 2005

Abstract

We investigate in depth the properties of a natural generalisation, called "sets of no-goods" in the AI literature, of boolean formulas in conjunctive normal forms such that *non-boolean variables* can be used. After building up a solid foundation, we generalise the notion of *deficiency*, and we obtain *polynomial time satisfiability decision* for generalised clause-sets with fixed maximal deficiency (generalising the boolean case, where the deficiency is the difference between the number of clauses and the number of variables). The main tool here is *autarky theory* (where autarkies are generalisations of satisfying assignments), and we concentrate especially on *matching autarkies* (based on matching theory) and special cases of *linear autarkies* (based on linear programming and linear algebra). As an application we study the problem of *hypergraph colouring*, considering early results of P.D. Seymour on the number of edges in *minimally non-2-colourable hypergraphs*, which are reinterpreted and generalised in the light of autarky theory.

1 Introduction

Satisfiability problems with constraint variables having more than two values occur naturally at many places, for example in colouring problems. Translations into boolean satisfiability problems are interesting and useful (see [14, 33] for various techniques), however often they cause performance problems, and they hide to a certain degree the structure of the original problem, which causes these translations typically to be not very well suited for theoretical studies on the structure of the original problem. In this article we study non-boolean satisfiability problems closest to boolean conjunctive normal form, namely satisfiability of what is called *generalised clause-sets* (or sets of "no-goods").

Two aspects of clauses make processing of boolean clause-sets especially efficient: Only when assigning a value to all the variables in a clause can we falsify the clause,

*Supported by EPSRC Grant GR/S58393/01

and for each variable the value here is uniquely determined; on the other hand, by giving just one variable the right value we are always able to satisfy a clause. Thus a literal in a (generalised) clause should have exactly one possibility to become false, while otherwise it should evaluate to true. We arrive naturally at the concept for generalised literals (the earliest systematic use seems to be in [2]): A variable v has a domain D_v of values, and a literal is a pair (v, ε) of the variable and a value $\varepsilon \in D_v$ such that the literal becomes true under an assignment φ iff φ sets v to a value different than ε (i.e., $\varphi(v) \in D_v \setminus \{\varepsilon\}$). In case of $D_v = \{0, 1\}$ variable v becomes an ordinary boolean variable (the literal $(v, 0)$ representing the positive literal). In this article we investigate the basic combinatorial properties of generalised clause-sets, concentrating on the *theory of autarkies* and on structural properties of *minimally unsatisfiable generalised clause-sets*.

One driving force for the study of generalised clause-sets comes from the applications of learning falsifying partial assignments in SAT algorithms — the notion of a generalised clause as a "no-good" embodies this point of view (a clause encodes the partial assignment leading to a failure, and a partial assignment falsifies this clause iff it extends the failing assignment). In [29] a theoretical study of such uses of generalised clauses was initiated, however in this article our main application area is the *hypergraph colouring problem*; as a first evidence for the potential power of autarky theory as a unifying framework in combinatorics we will see, that early results of Seymour on critically colourable hypergraphs can be extended naturally by interpreting hypergraph colouring as a satisfiability problem on generalised clauses — here the formulation as a (generalised) satisfiability problem does not mask structure (the transformation is canonical, and does not use auxiliary variables), but actually helps to *reveal* structure.

1.1 Generalising the notion of deficiency

Using $c(F)$ for the number of clauses in a boolean clause-set, and $n(F)$ for the number of variables, in [13] the *deficiency* $\boldsymbol{\delta(F)} := c(F) - n(F)$ has been introduced and made fruitful for the study of minimally unsatisfiable boolean clause-sets as well as for the introduction of a new polynomial time decidable class of "matched" satisfiable clause-sets. Let $\mathcal{MUSAT}$ denote the class of minimally unsatisfiable clause-sets (unsatisfiable clause-sets, where each strict sub-clause-set is satisfiable). For $F \in \mathcal{MUSAT}$ the property $\forall\, F' \subset F : \delta(F') < \delta(F)$ has been shown; using $\delta^*(F) := \max_{F' \subseteq F} \delta(F')$ we get $\delta^*(F) = \delta(F)$ as well as "Tarsi's lemma" $\delta(F) \geq 1$ (since for the empty clause-set $\top \subset F$ we have $\delta(\top) = 0$). Furthermore the class $\mathcal{MSAT}$ of "matching satisfiable" clause-sets F defined by the condition $\delta^*(F) = 0$ has been introduced. All matching satisfiable clause-sets are in fact satisfiable, since by Hall's theorem the bipartite graph $B(F)$ contains a matching covering all variables, where the vertices of $B(F)$ are the clauses of F on the one side and the variables of F on the other side, while an edge joins a variable and a clause if that variable appears in the clause (positively or negatively). Or, using Tarsi's lemma, one argues that if $F \in \mathcal{MSAT}$ would be unsatisfiable, then F would contain some minimally unsatisfiable $F' \subseteq F$, for which $\delta(F') \geq 1$ would hold, contradicting $\delta^*(F) = 0$.

The study of the levels $\mathcal{MUSAT}(k)$ of minimally unsatisfiable boolean clause-sets F with $\delta(F) \leq k$ has attracted some attention. In [1] (where also Tarsi's lemma has been proven) the class $\mathcal{SMUSAT}$ of "strongly minimally unsatisfiable clause-sets" has been introduced, which are minimally unsatisfiable clause-sets such that

adding any literal to any clause renders them satisfiable, and a nice characterisation of $\mathcal{SMUSAT}(1) = \{F \in \mathcal{SMUSAT} : \delta(F) = 1\}$ has been given (yielding polynomial time decision of $\mathcal{SMUSAT}(1)$). Then in [7] a (poly-time) characterisation of $\mathcal{MUSAT}(1)$ has been obtained, followed by a characterisation of $\mathcal{MUSAT}(2)$ in [4], while in [39] some subclasses of $\mathcal{MUSAT}(3)$ and $\mathcal{MUSAT}(4)$ have been shown to be poly-time decidable. For arbitrary (constant) $k \in \mathbb{N}$ it has been shown in [3] that for $F \in \mathcal{MUSAT}(k)$ there is a tree resolution refutation using at most $2^{k-1} \cdot n(F)^2$ steps, and thus the classes $\mathcal{MUSAT}(k)$ are at least in NP. In [3] it has been conjectured that in fact all classes $\mathcal{MUSAT}(k)$ are in P.

This conjecture has been proven true in [22] (using tools from matroid theory), where more generally the classes $\mathcal{SAT}(k)$, consisting of all satisfiable clause-sets F with $\delta^*(F) \leq k$, have to be shown poly-time decidable, from which immediately poly-time decision of the classes $\mathcal{MUSAT}(k)$ and $\mathcal{SMUSAT}(k)$ follows. Actually the classes $\mathcal{USAT}(k)$ of *unsatisfiable* clause-sets F with $\delta^*(F) \leq k$ have been shown poly-time decidable by improving the "splitting theorem" from [7], yielding tree resolution refutations for F using at most $2^{k-1} \cdot n(F)$ steps and of a simple recursive structure, so that these refutations can be found in polynomial time by means of enumeration of the circuits of the transversal matroid $T(F)$ associated to the bipartite graph $B(F)$ (the independent subsets of $T(F)$ are the matching satisfiable sub-clause-sets of F). Independently also in [11] poly-time decision of the classes $\mathcal{MUSAT}(k)$ has been derived by extending techniques from bipartite matching theory to *directed* bipartite graphs. Improving the proofs from [11], the present author joint the team in [10]. Actually refining the techniques from [22], in [36] fixed-parameter tractability of $\mathcal{SAT}(k)$ is shown (all this for the boolean case).

After setting syntax and semantics for generalised clause-sets, the first main task tackled in the present paper is to transfer these results regarding the deficiency to generalised clause-sets. After suitably generalising the notion of deficiency, in Corollary 4.8 the "satisfiability-based" approach from [10] yields polynomial time satisfiability decision for generalised clause-sets with bounded maximal deficiency. Generalising fixed-parameter tractability turns out not to be straight-forward, and it might be the case, that non-boolean clause-sets do not allow fixed-parameter tractability w.r.t. the deficiency. The general framework for our considerations is autarky theory as started in [23], with emphasise on *matching autarkies* as introduced in [25].

A key point for structural investigations in (generalised) clause-sets is to understand the effects of applying partial assignments (see for example [5], where splitting of minimally unsatisfiable boolean clause-sets is studied in some depth), and in this paper we consider the basic questions regarding irredundant and minimally unsatisfiable generalised clause-sets (which leads in a natural way to the study of hitting clause-sets and generalisations). Finally, we leave generalisations behind, and with the third main subject of this paper, the applications of autarky theory to hypergraph colouring, we enter new ground; I believe that such combinations of (generalised) satisfiability theory and combinatorial theory have a future, waiting for us to be explored.

1.2 Hypergraph colouring

Given a hypergraph G and set C of "colours", a C-colouring of G is a map $f : V(G) \to C$ such that no hyperedge $E \in E(G)$ is "monochromatic" (that is, there

must be vertices $v, w \in E$ with $f(v) \neq f(w)$). Translating this colouring problem into a generalised satisfiability problem $F_C(G)$ is straightforward: For each hyperedge $E \in E(G)$ and each colour $\varepsilon \in C$ form the clause $\{(v, \varepsilon) : v \in E\}$, and $F_C(G)$ is the set of all these clauses; obviously the C-colourings of G correspond 1-1 to the (total) satisfying assignments for $F_C(G)$. Interesting examples of hypergraph colouring problems are given by the diagonal van der Waerden problems and the diagonal Ramsey problems. The van der Waerden numbers have been considered in [8, 32], and actually it seems that SAT solvers are performing quite well on them, and that possibly SAT solvers could help to compute new van der Waerden numbers[1)], so here is the problem: Consider natural numbers $k, m, n \in \mathbb{N}$, and let the hypergraph $\mathrm{WH}(m, n)$ have vertex set $\{1, \ldots, n\}$, while the hyperedges of $\mathrm{WH}(m, n)$ are the subset $E \subseteq \{1, \ldots, n\}$ of size m which form an arithmetic progression (that is, for every E there exist $a, d \in \{1, \ldots, n\}$ with $E = \{a+i \cdot d : i \in \{0, \ldots, m-1\}\}$); now the van der Waerden number $\mathrm{N_W}(k, m)$ is the minimal n such that $\mathrm{WH}(m, n)$ is not k-colourable. The corresponding generalised clause-sets are $\mathrm{F_W}(k, m, n) := F_{\{1,\ldots,k\}}(\mathrm{WH}(m, n))$, and if $\mathrm{F_W}(k, m, n)$ is satisfiable, then $\mathrm{N_W}(k, m) > n$, while if $\mathrm{F_W}(k, m, n)$ is unsatisfiable, then $\mathrm{N_W}(k, m) \leq n$; for $k = 2$ we obtain boolean clause-sets (I would like to point out how natural the translation is — no auxiliary variables are involved). Directly expressing the problem as a generalised clause-set, in this way also the non-diagonal versions of van der Waerden- and Ramsey problems can be immediately translated into generalised clause-sets.

In this article we consider problems from hypergraph theory related to the notion of deficiency. In [35] it was proven, that a hypergraph which is minimally non-2-colourable and has no isolated vertices, has at least as many edges as vertices. By recognising that this property only needs the non-existence of certain autarkies (as in the case of Tarsi's lemma), we can immediately generalise this result to the statement, that if a hypergraph without isolated vertices is minimally non-k-colourable for some $k \geq 2$, then it has at least as many edges as vertices. I hope that this (first) example can demonstrate the potential of considering hypergraph colouring problems as special generalised satisfiability problems[2)], while on the other hand many treasures of (hyper)graph theory are still waiting to be discovered for the SAT community (see the last section for an intriguing example from [35]).

1.3 Overview and main results

In **Section 2** we lay the foundations for our study of generalised clause-sets: Partial assignments for non-boolean variables, and fundamental notions and notations for graphs and hypergraphs. Then in **Section 3** generalised clause-sets are introduced and the operations associated with them. Autarkies and autarky systems for generalised clause-sets are reviewed in Subsection 3.4 (a useful result here is Lemma 3.1, showing how to actually find a non-trivial autarky when just given an oracle deciding whether a non-trivial autarky exists or not), while resolution for gener-

[1)] The problem sizes of formulas related to unknown Ramsey numbers on the other hand are likely too big to be manageable by any (current) SAT solver.

[2)] in graph theory and combinatorics there seems to be a tendency to ignore this potential; for example in [31], where (boolean) resolution is transferred to the 2-colouring problem for hypergraphs, the point is stressed that satisfiability is a special case of the 2-colouring problem for hypergraphs by reducing the SAT problem for boolean conjunctive normal form in a relatively simple way to the hypergraph 2-colouring problem — however that the inverse transformation is even simpler (not using any sort of "gadget") is not mentioned in this paper(!)

alised clause-sets is the subject of Subsection 3.5 (in Theorem 3.2 it is proven, that clauses can be used in some resolution refutation iff they cannot be satisfied by some autarky; computation of the lean kernel via "intelligent backtracking solvers" follows). The most basic polynomial time reductions for generalised clause-sets are presented in Subsection 3.6, and finally in Subsection 3.7 the conflict graph and related notions are introduced.

Section 4 on matching autarkies for generalised clause-sets is the largest section of the paper, and some of the main results are contained in here. First in Subsection 4.1 the notion of matching satisfiable clause-sets (first studied in [13]) is generalised in a natural way to generalised clause-sets, based on the generalised notion of deficiency. Theorem 4.6 in Subsection 4.2 as the first main result, guaranteeing the existence of satisfying assignments "close enough" to matching satisfying assignments, is established for generalised clause-sets, so that in Corollary 4.7 satisfiability decision for generalised clause-sets with bounded maximal deficiency can be derived, generalising and strengthening the approach from [10]. Then in Subsection 4.3 matching autarkies for generalised clause-sets are introduced, and the main properties are proven. A typical result here is the generalisation of "Tarsi's Lemma" in Corollary 4.20 (every generalised minimally unsatisfiable clause-set has deficiency at least one).

In **Section 5** we turn to the study of generalised clause-sets which are minimally unsatisfiable. Considering the larger class of irredundant generalised clause-sets (no clause is implied by the others), we study the question when irredundancy is preserved by applying partial assignments. The class of irredundant clause-sets which stay irredundant for all partial assignments is characterised in Corollary 5.5 as the class of hitting clause-sets, while in Lemma 5.7 we consider the bigger class of multihitting (generalised) clause-sets and show, that they have a unique minimally unsatisfiable core (if they are unsatisfiable). In Subsection 5.3 we then discuss the process of "saturation" as introduced [12]; for generalised clause-sets we have to face a considerably more complicated situation here than in the boolean case.

The main result of **Section 6** on linear autarkies for generalised clause-sets is Lemma 6.1, generalising one of the main results from [1], namely that for a minimally non-2-colourable hypergraph there exists a matching in the associated bipartite graph which covers all vertex nodes. Finally in **Section 7** we consider applications of our results to the theory of hypergraph colouring; as a first appetiser in Corollary 7.4 we generalise a well-known early result of P.D. Seymour (similar to Tarsi's Lemma, and also based on autarky theory now). We conclude by interpreting a result of Seymour on the characterisation of certain critical 3-colourable hypergraphs as a classification of a certain class of "diagonal" minimally unsatisfiable clause-sets with (relative) minimal deficiency. And finally a refinement of the chromatic number, namely the "autarky number", is outlined.

2 Preliminaries

2.1 Partial assignments

Fundamental for our considerations is the **monoid** $(\mathcal{PASS}, \circ, \emptyset)$ **of partial assignments** as introduced in Subsection 2.1 of [29], where the reader can find more information. Here we just recall the basic definitions.

The universe of variables is denoted by $\mathcal{VA}$; for each variable $v \in \mathcal{VA}$ by D_v we denote the **domain** of v, a finite non-empty set (otherwise arbitrary). We consider the domain of a variable as always known from the context. To avoid running out of variables, we make the assumption, that for all variables $v \in \mathcal{VA}$ and for all non-empty subsets $S \subseteq D_v$ there are infinitely many variables $v' \in \mathcal{VA}$ with $D_{v'} = S$. A variable $v \in \mathcal{VA}$ is called **boolean** if $D_v = \{0, 1\}$.

A **partial assignment** is a map φ with finite domain $\mathbf{var}(\boldsymbol{\varphi}) := \mathrm{dom}(\varphi) \subseteq \mathcal{VA}$, such that for all $v \in \mathrm{var}(\varphi)$ we have $\varphi(v) \in D_v$. The domain size of a partial assignment φ is denoted by $\boldsymbol{n}(\boldsymbol{\varphi}) := |\mathrm{var}(\varphi)| \in \mathbb{N}_0$. A special partial assignment is the empty partial assignment $\emptyset$. The set of all partial assignments is denoted by $\mathcal{PASS}$. We use the notation $\langle v_1 \to \varepsilon_1, \ldots, v_m \to \varepsilon_m \rangle$ to denote the partial assignment φ with $n(\varphi) = m$ and $\varphi(v_i) = \varepsilon_i$. For two partial assignments $\varphi, \psi \in \mathcal{PASS}$ their **composition** $\boldsymbol{\varphi \circ \psi}$ is defined as the partial assignment $\varphi \circ \psi$ with domain $\mathrm{var}(\varphi \circ \psi) = \mathrm{var}(\varphi) \cup \mathrm{var}(\psi)$ such that first ψ is evaluated and then φ, i.e., $(\varphi \circ \psi)(v) = \psi(v)$ if $v \in \mathrm{var}(\psi)$ while otherwise $(\varphi \circ \psi)(v) = \varphi(v)$. It is $(\mathcal{PASS}, \circ, \emptyset)$ a monoid. An alternative representation of this structure is obtained as follows: Make each D_v a semigroup $(D_v, \cdot)$ by defining $\varepsilon_1 \cdot \varepsilon_2 := \varepsilon_2$ for $\varepsilon_1, \varepsilon_2 \in D_v$. Adjoin an identity element "$*$" to each D_v, obtaining monoids D_v^*. Now $\mathcal{PASS}$ is isomorphic to the direct sum $\sum_{v \in \mathcal{VA}} D_v^*$ of the monoids (the sub-monoid of the direct product $\prod_{v \in \mathcal{VA}} D_v^*$ given by those elements where only finitely many components are different from $*$), where $\varphi \in \mathcal{PASS}$ corresponds to the map $\varphi^* \in \prod_{v \in \mathcal{VA}} D_v^*$ with $\varphi(v) = \varphi^*(v)$ for $v \in \mathrm{var}(\varphi)$ and $\varphi^*(v) = *$ for $v \in \mathcal{VA} \setminus \mathrm{var}(\varphi)$. This representation of partial assignments as total maps with distinguished "undefined" value $*$ actually has certain advantages over the above representation, since working with total maps is often easier than working with partial maps, and we get a somewhat richer algebraic structure; however in this article we stick to the first representation of partial assignments.

2.2 Graphs and hypergraphs

A *graph* G is a pair $G = (V, E)$ with vertex set $V(G) = V$ and edge set $E(G) = E \subseteq \binom{V}{2}$, where for a set M and $k \in \mathbb{N}_0$ by $\binom{M}{k}$ we denote the set of all subsets $T \subseteq M$ with $|T| = k$. So graphs here have no parallel edges and no loops. *Multigraphs* allow for parallel edges, while *general graphs* give names to edges and allow also for loops. A graph G' is a *subgraph* of a graph G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$; G' is called a *partial subgraph* of G if G' is a subgraph of G and $V(G') = V(G)$. A graph G is *complete* if all distinct vertices $v, w \in V(G)$ are adjacent. G is *bipartite*, if the chromatic number of G is at most 2, while G is *complete bipartite* if G is bipartite and addition of any edge to G either destroys the graph property or the bipartiteness property. More generally, G is called *complete k-partite* for $k \in \mathbb{N}_0$ if the chromatic number of G is at most k, and addition of any edge to G either destroys the graph property or increases the chromatic number. It is G complete k-partite iff G is the union of at most k independent sets, such that each pair of vertices from different independent sets is adjacent (equivalently, iff the complement of G is the disjoint union of at most k cliques).

A function $f : S \to \mathbb{R}$, where S is some set system stable under union and intersection, is called *submodular* resp. *supermodular* if for all $A, B \in S$ we have $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ resp. $f(A \cup B) + f(A \cap B) \geq f(A) + f(B)$, while f is called **modular** if f is submodular and supermodular. A prototypical example for a modular function is $f : \mathbb{P}_{\mathrm{f}}(X) \to \mathbb{N}_0$ given by $A \in \mathbb{P}_{\mathrm{f}}(X) \mapsto f(A) := |A|$,

where for a set X by $\mathbb{P}_\mathrm{f}(X)$ we denote the set of finite subsets of X.

For a graph G and a vertex set $A \subseteq V(G)$ the *neighbourset* $\Gamma_G(A)$ is defined as the set of vertices adjacent to at least one element of A. For a finite graph G it is $A \subseteq V(G) \mapsto |\Gamma(A)|$ a prototypical example for a submodular function, while the *deficiency* $\delta(A) := |A| - |\Gamma(A)| \in \mathbb{Z}$ is a supermodular function (as the difference of a modular function and a submodular function).

A *hypergraph* G is a pair $G = (V, E)$, where the set $V(G) = V$ is the set of "vertices" of G, while $E(G) = E \subseteq \mathbb{P}_\mathrm{f}(V)$ is the set of "hyperedges" of G (often also just called "edges"). Every graph is a hypergraph. An *isolated vertex* v of a hypergraph G is an element of $V(G) \setminus \bigcup E(G)$. Any set S of finite sets can be considered as a hypergraph G_S with vertex set $V(G_S) = \bigcup S$ and hyperedge set $E(G_S) = S$ (the hypergraphs arising in this way are precisely the hypergraphs without isolated vertices). A hypergraph G is *finite* if $V(G)$ is finite (and then also $E(G)$ must be finite).

A *C-colouring* of a hypergraph G for some "colour-set" C is a map $f : V(G) \to C$ such that for all hyperedges $e \in E(G)$ there exist vertices $v, w \in e$ with $f(v) \neq f(w)$; a *k-colouring* for $k \in \mathbb{N}_0$ is a $\{1, \ldots, k\}$-colouring. A hypergraph G containing a hyperedge $e \in E(G)$ with $|e| \leq 1$ has no C-colouring for any C, and thus is called *uncolourable*, while otherwise the identity yields a $V(G)$-colouring for every hypergraph G, and G is called *colourable*.[3)] The *chromatic number* $\chi(G)$ of a colourable hypergraph G is the minimal cardinality of a set C such that a C-colouring of G exists. We have $0 \leq \chi(G) \leq |V(G)|$, where $\chi(G) = 0$ iff $V(G) = \emptyset$ and $\chi(G) = 1$ iff $V(G) \neq \emptyset$ but $E(G) = \emptyset$. A (colourable) hypergraph G is called *k-chromatic critical* for $k \in \mathbb{N}_0$ if $\chi(G) = k$, and for every edge $e \in E(G)$ we have $\chi((V(G), E(G) \setminus \{e\})) < k$ (if $\chi(G) \leq 1$, then G is critically $\chi(G)$-colourable, since there are no edges). A colourable G is *minimally non-k-colourable* for $k \in \mathbb{N}_0$ (i.e., G is not k-colourable, but after removing any edge G becomes k-colourable) iff G is $(k+1)$-chromatic critical (note that the hypergraph $(\emptyset, \emptyset)$ with empty vertex set is not minimally non-k-colourable for any $k \in \mathbb{N}_0$, while $(\emptyset, \emptyset)$ is the only hypergraph which is critical 0-colourable); if G is uncolourable, then G is minimally non-k-colourable iff $|E(G)| = 1$, where in case of $V(G) = \emptyset$ (and thus $E(G) = \{\emptyset\}$) it is G minimally non-k-colourable for all $k \in \mathbb{N}_0$, while otherwise G is minimally non-k-colourable for all $k \in \mathbb{N}$.

3 Generalised (multi-)clause-sets

In this section we review the notion of generalised multi-clause-sets and the basic facts about them regarding autarkies and resolution.

In Subsection 3.1 we introduce the notion of "generalised multi-clause-sets" and "generalised clause-sets", while in Subsection 3.2 (partial) assignments and their operation on (multi-)clause-sets is discussed. This introduction into "syntax and

[3)] Typically, in hypergraph theory colourings ignore hyperedges of size at most one (see for example Chapter 7 in [18]), which unnecessarily breaks the natural connection to the theory of generalised clause-sets. It seems better to me to depart from this tradition. In this way we are also consistent with the notion of colourings for graphs: If a (general) graph has a loop, then it has no colouring at all, and in the same vain, if a hypergraph contains the empty edge or an edge of size one, then it has no colouring at all. When considering the canonical translation of hypergraph colouring problems into satisfiability problems for generalised clause-sets, then hypergraphs with hyperedges of size one or zero translate into clause-sets which are ("automatically") unsatisfiable.

semantics of generalised clause-sets" is completed in Subsection 3.3 with the discussion of various operations on (multi-)clause-sets F regarding their variable structure (that is, disregarding the different "polarities", i.e., disregarding the literal structure).

Of central importance to our work is Subsection 3.4, where the notion of *autarkies* (special partial assignments, which satisfy parts of the formula, and leave the rest untouched) and *autarky systems* (allowing to tailor the notion of autarkies for special purposes) for multi-clause-sets are introduced. In Subsection 3.5 then resolution for generalised clause-sets is discussed, while in Section 3.6 we give the most basic reductions for generalised clause-sets. Finally in Subsection 3.7 some very basic notions regarding the conflict structure of generalised clause-sets are introduced.

For more background information, see [29, 24] for a general, axiomatic framework for "generalised satisfiability problems", while in Subsection 2.3 of [29] generalised clause-sets are discussed, and in Section 2 of [27] boolean multi-clause-sets are considered (see also [26] for more information). In this paper, when we speak of "clause-sets" then we always mean "generalised clause-sets", while clause-sets in the "traditional" sense are always qualified as "boolean clause-sets"; however in lemmas, corollaries and theorems we always speak of "generalised clause-sets" to ease independent access.

3.1 Syntax: The notion of "multi-clause-sets"

A **literal** is a pair (v, ε) of a variable $v \in \mathcal{VA}$ and a value $\varepsilon \in D_v$; we write $\mathbf{var}(v, \varepsilon) := v$ and $\mathbf{val}(v, \varepsilon) := \varepsilon$. The set of all literals is denoted by $\mathcal{LIT}$. For a partial assignment $\varphi \in \mathcal{PASS}$ and a literal (v, ε) with $v \in \mathrm{var}(\varphi)$ we set $\varphi((v, \varepsilon)) = 1$ if $\varphi(v) \neq \varepsilon$, while we set $\varphi((v, \varepsilon)) = 0$ if $\varphi(v) = \varepsilon$; thus a literal (v, ε) has the meaning "v shall *not* get value ε".

A **clause** C is a finite set of literals not containing "clashing literals", that is for literals $x, y \in C$ with $x \neq y$ we have $\mathrm{var}(x) \neq \mathrm{var}(y)$. The set of all clauses is denoted by $\mathcal{CL}$. For a clause C we set $\mathrm{var}(C) := \{\mathrm{var}(x) : x \in C\}$. Using the representation of maps as ordered pairs of arguments and values, actually $\mathcal{CL} = \mathcal{PASS}$, and I consider this 1-1 correspondence between clauses and partial assignments as fundamental; explicitely said, a clause corresponds to the partial assignment which sets exactly the literals in the clause to false.[4)] The empty clause is denoted by $\bot \in \mathcal{CL}$.

A **multi-set of clauses** is a map $F : \mathcal{CL} \to \mathbb{N}_0$ (assigning to each clause its number of occurrences), while a **set of clauses** is a subset of $\mathcal{CL}$. Sets of clauses F can be implicitly converted to multi-sets of clauses by setting $F(C) := 1$ for $C \in F$ and $F(C) := 0$ otherwise. We write $C \in F$ for a multi-set F of clauses iff $F(C) > 0$. For a multi-set of clauses F we set $\mathrm{var}(F) := \bigcup\{\mathrm{var}(C) : C \in F\}$. The underlying **variable hypergraph** of a (multi-)clause-set F is the hypergraph given by the set-system $\{\mathrm{var}(C) : C \in F\}$. For a multi-set of clauses F and a variable $v \in \mathcal{VA}$ we define $\mathbf{val}_v(F) := \{\varepsilon \in D_v \mid \exists\, C \in F : (v, \varepsilon) \in C\}$. We have $\mathrm{var}(F) = \{v \in \mathcal{VA} : \mathrm{val}_v(F) \neq \emptyset\}$. Finally, for a multi-set F of clauses the

[4)] If in the literature a correspondence between clauses and partial assignments for the boolean case is considered, then often it goes what I consider the wrong way, namely to a clause the partial assignment which sets all the literals in the clause to true is assigned. For boolean clause-set one can always live with that, it only makes the argumentation less straight-forward; but for generalised clause-sets it becomes obvious that this can not be generalised.

underlying set of clauses $\hat{\mathrm{t}}(F)$ is defined as $\hat{\mathrm{t}}(F) = \{C \in \mathcal{CL} : C \in F\}$, and the empty clause-set as well as the empty multi-clause-set is denoted by $\top$.

We use the following complexity measures for multi-sets F of clauses (with nonnegative integers and possibly $+\infty$ as values of the measures):

1. $\#_{(v,\varepsilon)}(F) := \sum_{C \in F, (v,\varepsilon) \in C} F(C) \in \mathbb{N}_0 \cup \{+\infty\}$ measures the number of occurrences of a literal;
2. $\#_v(F) := \sum_{\varepsilon \in D_v} \#_{(v,\varepsilon)}(F) = \sum_{C \in F, v \in \mathrm{var}(C)} F(C) \in \mathbb{N}_0 \cup \{+\infty\}$ measures the number of occurrences of a variable;
3. $s_{(v,\varepsilon)}(F) := \sum_{\varepsilon' \in D_v \setminus \{\varepsilon\}} \#_{(v,\varepsilon)}(F) = \#_v(F) - \#_{(v,\varepsilon)}$ measures the number of occurrences of literals with variable v and value different from ε (this is the number of satisfied clauses when assigning value ε to v; see below);
4. $n(F) := |\mathrm{var}(F)| \in \mathbb{N}_0 \cup \{+\infty\}$ measures the number of variables;
5. $c(F) := \sum_{C \in F} F(C) \in \mathbb{N}_0 \cup \{+\infty\}$ measures the number of clauses;
6. $\ell(F) := \sum_{C \in F} F(C) \cdot |C| = \sum_{v \in \mathrm{var}(F)} \#_v(F) \in \mathbb{N}_0 \cup \{+\infty\}$ measures the number of literal occurrences;
7. $\mathbf{mds}(F) := \max_{v \in \mathrm{var}(F)} |D_v| \in \mathbb{N}_0 \cup \{+\infty\}$ measures the maximal domain size.

And for multi-sets F_1, F_2 of clauses we use the following operations and relations:

1. the multi-set $F_1 + F_2$ of clauses is defined by $(F_1 + F_2)(C) := F_1(C) + F_2(C)$ for clauses C;
2. the multi-set $F_1 \cup F_2$ resp. $F_1 \cap F_2$ of clauses is given by $(F_1 \cup F_2)(C) := \max(F_1(C), F_2(C))$ resp. $(F_1 \cap F_2)(C) := \min(F_1(C), F_2(C))$ for clauses C; if F_1, F_2 are sets of clauses, then these operations coincide with the ordinary set operations;
3. if F_2 is a set of clauses, then the multi-set $F_1 \setminus F_2$ of clauses is defined by $(F_1 \setminus F_2)(C) := 0$ for $C \in F_2$, while otherwise $(F_1 \setminus F_2)(C) := F_1(C)$; if also F_1 is a set of clauses, then $F_1 \setminus F_2$ is the ordinary set operation;
4. the relation $F_1 \leq F_2$ holds if for all clauses C we have $F_1(C) \leq F_2(C)$; we use $F' \lneq F$ for $F' \leq F \wedge F' \neq F$; if F_1, F_2 are sets of clauses, then $F_1 \leq F_2 \Leftrightarrow F_1 \subseteq F_2$;
5. F_1 is called a **sub-multi-clause-set** of F_2 if $F_1 \leq F_2$ holds, while F_1 is called an **induced sub-multi-clause-set** of F_2 if $F_1 \leq F_2$ and $\forall\, C \in F_1 : F_1(C) = F_2(C)$ holds; every sub-clause-set of a clause-set is induced;
6. if F_2 is a sub-multi-clause-set of F_1, then the multi-set $F_1 - F_2$ of clauses is defined via $(F_1 - F_2)(C) := F_1(C) - F_2(C)$ for clauses C.

A **multi-clause-set** is a multi-set F of clauses with finite $\mathrm{var}(F)$, and a **clause-set** is a set F of clauses with finite $\mathrm{var}(F)$ (thus a multi-set of clauses is a multi-clause-set iff the underlying set of clauses is a clause-set). The set of all multi-clause-sets is denoted by $\mathcal{MCLS}$, the set of all clause-sets by $\mathcal{CLS}$. If $\mathcal{C}$ is a set of multi-clause-sets and $f : \mathcal{C} \to \mathbb{R}$, then by $\mathcal{C}_{f \leq b}$ for some $b \in \mathbb{R}$ we denote the set of all $F \in \mathcal{C}$ with $f(F) \leq b$; analogously we define $\mathcal{C}_{f=b}$, $\mathcal{C}_{f \geq b}$ etc.

Some comments on the various finiteness conditions employed in this paper: Since the domains D_v of variables $v \in \mathcal{VA}$ are finite, clause-sets are the finite sets of

clauses, and we have the (well-known) "compactness property", that is, if a set F of clauses is unsatisfiable, then there exists a sub-clause-set $F' \subseteq F$ such that already F' is unsatisfiable. Allowing infinite sets of clauses is a sometimes useful extension, which does not really change the underlying logic due to compactness, however allowing infinite *clauses* would go up a big step in logical complexity, employing now infinitary logics (which do not have compactness). Thus the finiteness of clauses seems essential to me, and accordingly also partial assignments must be finite. Considering only hypergraphs with finite hyperedges corresponds to this restriction, and witnessing the higher complexity of infinitary logics, hypergraphs with infinite edges show a much more complicated structure.

3.2 Semantics: The operation of partial assignments on multi-clause-sets

Now we define the operation $* : \mathcal{PASS} \times \mathcal{MCLS} \to \mathcal{MCLS}$ of $\mathcal{PASS}$ on multi-clause-sets, and the operation $* : \mathcal{PASS} \times \mathcal{CLS} \to \mathcal{CLS}$ on clause-sets, which in both cases have the meaning of substituting values into variables and carrying out the resulting simplifications (viewing a clause as a disjunction of its literals, and a (multi-)clause-set as a conjunction of its clauses), with the only difference that in the case of clause-sets contractions in the result are carried out (distinct clauses can become equal after a substitution). The case of clause-sets is reduced to the case of multi-clause-sets, using the explicit transformation $\check{\mathrm{t}} : \mathcal{CLS} \to \mathcal{MCLS}$ of clause-sets into multi-clause-sets. For $F \in \mathcal{MCLS}$ and $\varphi \in \mathcal{PASS}$ we define $\boldsymbol{\varphi * F} \in \mathcal{MCLS}$ by

$$(\varphi * F)(C) = \sum_{\substack{C' \in \mathcal{CL} \\ \varphi * \{C'\} = \{C\}}} F(C'),$$

for $C \in \mathcal{CL}$, where for a clause C we set $\varphi * \{C\} := \top \in \mathcal{CLS}$ if there exists a literal $x \in C$ with $\varphi(x) = 1$, while otherwise we set $\varphi * \{C\} := \{C \setminus C_\varphi\} \in \mathcal{CLS}$ (i.e., we remove the falsified literals from C). And for $F \in \mathcal{CLS}$ we define $\varphi * F \in \mathcal{CLS}$ as

$$\varphi * F := \hat{\mathrm{t}}(\varphi * \check{\mathrm{t}}(F)).$$

We have here $\varphi * F = \bigcup_{C \in F} \varphi * \{C\}$. The effect on the basic measures of applying a partial assignment $\langle v \to \varepsilon \rangle$ to $F \in \mathcal{MCLS}$ with $v \in \mathrm{var}(F)$ is given by

$$\begin{aligned} n(\langle v \to \varepsilon \rangle * F) &\leq n(F) - 1 \\ c(\langle v \to \varepsilon \rangle * F) &= c(F) - s_{(v,\varepsilon)}(F). \end{aligned}$$

A multi-clause-set $F \in \mathcal{MCLS}$ is **satisfiable** if there exists a partial assignment $\varphi \in \mathcal{PASS}$ with $\varphi * F = \top$, while otherwise F is **unsatisfiable**; the set of all satisfiable multi-clause-sets is denoted by $\boldsymbol{\mathcal{SAT}}$, the set of all unsatisfiable multi-clause-sets by $\boldsymbol{\mathcal{USAT}}$. A generalised multi-clause-set $F \in \mathcal{MCLS}$ is called **minimally unsatisfiable** if F is unsatisfiable, but every $F' \subsetneq F$ is satisfiable; obviously if F is minimally unsatisfiable, then F actually is a clause-set. The set of all minimally unsatisfiable generalised clause-sets is denoted by $\boldsymbol{\mathcal{MUSAT}}$.

It is useful to have some notations for the set of satisfying assignments ("models") as well as for the set of falsifying assignments. For $V \in \mathbb{P}_\mathrm{f}(\mathcal{VA})$ let $\boldsymbol{\mathcal{PASS}(V)}$ be the set of $\varphi \in \mathcal{PASS}$ with $\mathrm{var}(\varphi) = V$. Note that we have $|\mathcal{PASS}(V)| = \prod_{v \in V} |D_v|$. Now for a multi-clause-set $F \in \mathcal{MCLS}$ and for a set V of variables with $\mathrm{var}(F) \subseteq V \in \mathbb{P}_\mathrm{f}(\mathcal{VA})$ let $\boldsymbol{\mathfrak{S}_V(F)}$ be the set of $\varphi \in \mathcal{PASS}(V)$ with $\varphi * F = \top$,

while $\boldsymbol{\mathfrak{F}_V(F)}$ is the set of $\varphi \in \mathcal{PASS}(V)$ with $\bot \in \varphi * F$. Thus F is satisfiable iff $\mathfrak{S}_V(F) \neq \emptyset$, and for any clause C with $\mathrm{var}(C) \subseteq V$ we have

$$|\mathfrak{F}_V(\{C\})| = |\mathcal{PASS}(V \setminus \mathrm{var}(C))| = \prod_{v \in V \setminus \mathrm{var}(C)} |D_v|.$$

Obviously $\mathfrak{S}_V(F) \cap \mathfrak{F}_V(F) = \emptyset$ and $\mathfrak{S}_V(F) \cup \mathfrak{F}_V(F) = \mathcal{PASS}(V)$. By definition we have

$$\mathfrak{F}_V(F) = \bigcup_{C \in F} \mathfrak{F}_V(\{C\}).$$

For multi-clause-sets F_1, F_2 we write $\boldsymbol{F_1 \models F_2}$ ("F_1 implies F_2") if for all $\varphi \in \mathcal{PASS}$ with $\varphi * F_1 = \top$ we have $\varphi * F_2 = \top$ as well, and for clauses C we write $F \models C$ instead of $F \models \{C\}$. It is $F \in \mathcal{MCLS}$ unsatisfiable iff $F \models \bot$. Note that $F_1 \models F_2$ holds iff for $V := \mathrm{var}(F_1) \cup \mathrm{var}(F_2)$ we have $\mathfrak{F}_V(F_2) \subseteq \mathfrak{F}_V(F_1)$. We call F_1, F_2 **equivalent** if $F_1 \models F_2$ and $F_2 \models F_1$.

The basic laws for the operation of partial assignments on multi-clause-sets are as follows, using $F, F_1, F_2 \in \mathcal{MCLS}$ and $\varphi, \psi \in \mathcal{PASS}$:

$$\emptyset * F = F$$
$$\varphi * \top = \top$$
$$(\varphi \circ \psi) * F = \varphi * (\psi * F)$$
$$\varphi * (F_1 + F_2) = \varphi * F_1 + \varphi * F_2.$$

If $F_1, F_2 \in \mathcal{CLS}$, then we have

$$\varphi * (F_1 \cup F_2) = \varphi * F_1 \cup \varphi * F_2$$

(but this does not hold for multi-clause-sets in general). Furthermore for a multi-clause-set F and a clause-set F' we have $\varphi * (F \setminus F') \geq (\varphi * F) \setminus (\varphi * F')$.

3.3 Three operations of sets of variables on multi-clause-sets

Finally we consider various operations with sets of variables. The operation $* : \mathbb{P}_\mathrm{f}(\mathcal{VA}) \times \mathcal{MCLS} \to \mathcal{MCLS}$ is defined for $V \in \mathbb{P}_\mathrm{f}(\mathcal{VA})$ and $F \in \mathcal{MCLS}$ via

$$(V * F)(C) := \sum_{\substack{C' \in \mathcal{CL} \\ V * C' = C}} F(C'),$$

where for a clause C we set $V * C := \{x \in C : \mathrm{var}(x) \notin V\} \in \mathcal{CL}$. That is, $\boldsymbol{V * F}$ is obtained from F by crossing out all literal occurrences x with $\mathrm{var}(x) \in V$. Two basic properties are

$$\begin{aligned} \mathrm{var}(V * F) &= \mathrm{var}(F) \setminus V \\ c(V * F) &= c(F). \end{aligned}$$

The operation $* : \mathbb{P}_\mathrm{f}(\mathcal{VA}) \times \mathcal{CLS} \to \mathcal{CLS}$ is defined for $F \in \mathcal{CLS}$ by

$$V * F := \hat{\mathrm{t}}(V * \check{\mathrm{t}}(F)).$$

We have here $V * F = \{V * C : C \in F\}$. The basic laws for $F, F_1, F_2 \in \mathcal{MCLS}$ and $V, V' \in \mathbb{P}_\mathrm{f}(\mathcal{VA})$ are

$$\emptyset * F = F$$
$$V * \top = \top$$
$$(V \cup V') * F = V * (V' * F)$$
$$V * (F_1 + F_2) = V * F_1 + V * F_2.$$

If $F_1, F_2 \in \mathcal{CLS}$, then we have

$$V * (F_1 \cup F_2) = V * F_1 \cup V * F_2$$

(again this does not hold for multi-clause-sets in general).

We conclude with different forms of selecting parts of a multi-clause-set. By $\boldsymbol{F_V}$ we denote the induced sub-multi-clause-set of F with $C \in F_V \Leftrightarrow \mathrm{var}(C) \cap V \neq \emptyset$; in other words, $F_V = F \setminus \{C \in F : \mathrm{var}(C) \cap V = \emptyset\}$. Basic properties are:

1. $F_\emptyset = \top$ and $F_{\mathrm{var}(F)} = F \setminus \{\bot\}$.
2. If $V_1 \subseteq V_2$, then F_{V_1} is an induced sub-multi-clause-set of F_{V_2}.
3. $F_{V_1 \cup V_2} = F_{V_1} \cup F_{V_2}$.
4. For $v \in \mathcal{VA}$ we have $c(F_{\{v\}}) = \#_v(F)$.

Finally

$$\boldsymbol{F[V]} := (\mathrm{var}(F) \setminus V) * F_V = ((\mathrm{var}(F) \setminus V) * F) \setminus \{\bot\} \in \mathcal{MCLS}.$$

Basis properties are

1. $F[\emptyset] = \top$ and $F[\mathrm{var}(F)] = F \setminus \{\bot\}$.
2. $c(F[V]) = c(F_V)$, $\mathrm{var}(F[V]) \subseteq \mathrm{var}(F_V)$.
3. $\mathrm{var}(F[V]) = V$ for $V \subseteq \mathrm{var}(F)$.

To summarise: We obtain $V * F$ from F by keeping all clauses but removing those literals x from them with $\mathrm{var}(x) \in V$, while we obtain F_V from F by removing those clauses C from F with $\mathrm{var}(C) \cap V = \emptyset$ (while keeping all clauses intact); finally $F[V]$ is obtained from F by first constructing F_V, and then crossing out all literal occurrences for literals x where there exists a clause $C \in F$ with $\mathrm{var}(C) \cap V = \emptyset$ and $\mathrm{var}(x) \in \mathrm{var}(C)$.

3.4 Autarky systems for generalised multi-clause-sets

Now we review the general properties of autarkies and autarky systems for generalised multi-clause-sets. See Section 3 in [24] for a general theory of autarkies and autarky systems, while in Section 4 of [24] autarky systems for generalised clause-sets have been discussed (easily generalised to autarky systems for generalised multi-clause-sets). General properties of autarkies for boolean clause-sets are thoroughly investigated in [23], Section 3, while autarky systems for boolean clause-sets have been introduced in [25] (see Sections 4 and 8 for the general theory).

A partial assignment $\varphi \in \mathcal{PASS}$ is an **autarky** for $F \in \mathcal{MCLS}$ if one (and thus all) of the following four equivalent conditions is fulfilled:

1. for all clauses $C \in F$ we have $\mathrm{var}(\varphi) \cap \mathrm{var}(C) \neq \emptyset \Rightarrow \varphi * \{C\} = \top$;

2. $\forall\, F' \leq F : \varphi * F' \leq F'$;
3. φ is a satisfying assignment for $F_{\mathrm{var}(\varphi)}$;
4. φ is a satisfying assignment for $F[\mathrm{var}(\varphi)]$.

Obviously, φ is an autarky for F iff φ is an autarky for $F \setminus \{\bot\}$. The set of all autarkies for F is denoted by $\mathbf{Auk}(\boldsymbol{F})$; it is $\mathrm{Auk}(F)$ a sub-monoid of $\mathcal{PASS}$, containing all satisfying assignments for F in case F is satisfiable, and $\mathrm{Auk}(F) = \mathrm{Auk}(\hat{\mathrm{t}}(F))$. If $F' \leq F$, then $\mathrm{Auk}(F) \subseteq \mathrm{Auk}(F')$, and for $V \in \mathbb{P}_{\mathrm{f}}(\mathcal{VA})$ we have $\{\varphi \in \mathrm{Auk}(V * F) : \mathrm{var}(\varphi) \cap V = \emptyset\} = \{\varphi \in \mathrm{Auk}(F) : \mathrm{var}(\varphi) \cap V = \emptyset\}$. Furthermore we have $\mathrm{Auk}(F_1 + F_2) = \mathrm{Auk}(F_1) \cap \mathrm{Auk}(F_2)$. If $\varphi \in \mathrm{Auk}(F)$ and $\psi \in \mathrm{Auk}(\varphi * F)$, then $\psi \circ \varphi \in \mathrm{Auk}(F)$. An autarky $\varphi \in \mathrm{Auk}(F)$ is called **non-trivial** if $\mathrm{var}(\varphi) \cap \mathrm{var}(F) \neq \emptyset$ holds. F is called **lean**, if F has no non-trivial autarky; the set of all lean multi-clause-sets is denoted by $\boldsymbol{\mathcal{LEAN}}$. A sum of lean multi-clause-sets again is lean. If F is lean, so is $V * F$ for $V \subseteq \mathcal{VA}$.

An **autarky reduction** is a reduction $F \to \varphi * F$ for some non-trivial autarky φ for F (note that $\varphi * F$ is satisfiability equivalent to F). Autarky reduction is terminating and confluent (generalising Lemma 4.1 in [25], a special case of Lemma 3.7 in [24]), and thus the result of iterated autarky reductions until no further reductions are possible is uniquely determined; we denote it by $\mathbf{N_{Auk}}(\boldsymbol{F}) \leq F$. It is $\mathbf{N_a} := \mathrm{N}_{\mathrm{Auk}}$ a "kernel operator", that is, $\mathrm{N_a}(F) \leq F$, $\mathrm{N_a}(\mathrm{N_a}(F)) = \mathrm{N_a}(F)$, and $F_1 \leq F_2 \Rightarrow \mathrm{N_a}(F_1) \leq \mathrm{N_a}(F_2)$; furthermore $\mathrm{N_a}(F)$ is satisfiability equivalent to F, and $\mathrm{N_a}(F) = \top$ iff $F \in \mathcal{SAT}$. We have $\mathrm{N_a}(F) \in \mathcal{LEAN}$, and $\mathrm{N_a}(F)$ is called the **lean kernel** of F; F is lean iff $\mathrm{N_a}(F) = F$. There exists an autarky $\varphi \in \mathrm{Auk}(F)$ with $\mathrm{N_a}(F) = \varphi * F$ (while for all $\varphi \in \mathrm{Auk}(F)$ we have $\mathrm{N_a}(F) \leq \varphi * F$). It is $\mathrm{N_a}(F)$ the largest lean sub-multi-clause-set of F.

An **autark sub-multi-clause-set** F' of F is an induced sub-multi-clause-set of F, such that there exists an autarky $\varphi \in \mathrm{Auk}(F)$ so that for $C \in F$ we have $C \in F'$ iff $\varphi * \{C\} = \top$ (note that in this case we have $F' = F_{\mathrm{var}(\varphi)}$). The set of autark sub-multi-clause-sets of F is closed under union, and contains the smallest element $\top$ and the largest element $F \setminus \mathrm{N_a}(F)$. It is $F' \leq F$ an autark sub-multi-clause-set of F iff there is $V \subseteq \mathrm{var}(F)$ with $F_V = F'$ and $F[V] \in \mathcal{SAT}$.

The relation between the lean kernel of F and the largest autark sub-multi-clause-set of F can be summarised as follows: For $F \in \mathcal{MCLS}$ there exist induced sub-multi-clause-sets $F_1, F_2 \leq F$ with $F_1 + F_2 = F$, such that F_1 is lean, while $\mathrm{var}(F_1) * F_2$ is satisfiable; in this decomposition F_1, F_2 are uniquely determined, namely $F_1 = \mathrm{N_a}(F)$ is the largest lean sub-multi-clause-set (the lean kernel), while F_2 is the largest autark sub-multi-clause-set.

After having reviewed the general facts for autarkies for generalised multi-clause-sets, we now consider "autarky systems". The motivation for doing so is, that instead of (computationally infeasible) general autarkies we want to consider restricted autarkies, and under mild assumptions on these restricted autarkies all the above facts carry over (in generalised form). The monoid $(\mathcal{PASS}, \circ, \emptyset)$ together with the partial order $(\mathcal{MCLS}, \leq, \top)$ with least element and together with the operation $*$ of $\mathcal{PASS}$ on $\mathcal{MCLS}$ fulfils all the axioms required in Section 3 of [24], and thus all the general results there on autarky systems hold here.

An **autarky system** for generalised multi-clause-sets is a map $\mathcal{A}$, which assigns to every $F \in \mathcal{MCLS}$ a sub-monoid $\mathcal{A}(F)$ of $\mathrm{Auk}(F)$, such that for $F_1 \leq F_2$ we have $\mathcal{A}(F_2) \subseteq \mathcal{A}(F_1)$. The elements of $\mathcal{A}(F)$ are called $\boldsymbol{\mathcal{A}}$**-autarkies** for F. Further possible restrictions on $\mathcal{A}$ are expressed by the following notions:

1. $\mathcal{A}$ is **iterative**, if for $\varphi \in \mathcal{A}(F)$ and $\psi \in \mathcal{A}(\varphi * F)$ we always have $\psi \circ \varphi \in \mathcal{A}(F)$.
2. $\mathcal{A}$ is called **standardised**, if for a partial assignment $\varphi \in \mathcal{PASS}$ we have $\varphi \in \mathcal{A}(F)$ iff $\varphi \mid \mathrm{var}(F) \in \mathcal{A}(F)$ (where $\varphi \mid \mathrm{var}(F)$ is the restriction of the map φ to the domain $\mathrm{var}(\varphi) \cap \mathrm{var}(F)$). (Remark: Thus for a standardised autarky system $\mathcal{A}$ all partial assignments φ with $\mathrm{var}(\varphi) \cap \mathrm{var}(F) = \emptyset$ are (trivial) $\mathcal{A}$-autarkies for F. In [24] only the direction "$\varphi \in \mathcal{A}(F) \Rightarrow \varphi \mid \mathrm{var}(F) \in \mathcal{A}(F)$ is required, but now it seems more systematic to me to require also the other direction.)
3. $\mathcal{A}$ is **$\bot$-invariant**, if always $\mathcal{A}(F) = \mathcal{A}(F + \{\bot\})$ holds (in [24, 25] this was called "normal").
4. $\mathcal{A}$ is **stable under variable elimination**, if for $V \in \mathbb{P}_\mathrm{f}(\mathcal{VA})$ we always have $\{\varphi \in \mathcal{A}(V * F) : \mathrm{var}(\varphi) \cap V = \emptyset\} = \{\varphi \in \mathcal{A}(F) : \mathrm{var}(\varphi) \cap V = \emptyset\}$.

An autarky system $\mathcal{A}$ is called **normal**, if it is iterative, standardised, $\bot$-invariant and stable under variable elimination. In [24, 25] "normal autarky systems" have been called "strong autarky systems", but meanwhile the above four properties seem not so strong anymore to me, but quite "normal". Examples for normal autarky systems are the smallest standardised autarky system $F \in \mathcal{MCLS} \mapsto \{\varphi \in \mathcal{PASS} : \mathrm{var}(\varphi) \cap \mathrm{var}(F) = \emptyset\}$ and the largest autarky system $F \in \mathcal{MCLS} \mapsto \mathrm{Auk}(F)$. In this paper our main interest is in normal autarky systems, and thus we don't investigate further the relations between the above notions and the other properties of autarky systems, but we will state general results only either for all autarky systems or for all normal autarky systems.

Consider an autarky system $\mathcal{A}$. An **$\mathcal{A}$-reduction** is a reduction $F \mapsto \varphi * F$ for some non-trivial $\varphi \in \mathcal{A}(F)$. Since multi-clause-sets have finite variable sets, $\mathcal{A}$-reduction is terminating, and thus by Lemma 3.7 in [24] $\mathcal{A}$-reduction is confluent, and the result of applying $\mathcal{A}$-reductions as long as possible is uniquely determined, yielding a normal form $\mathbf{N_\mathcal{A}(}F\mathbf{)} \leq F$. As before, the operator $\mathrm{N}_\mathcal{A}$ is a kernel operator, that is, $\mathrm{N}_\mathcal{A}(F) \leq F$, $\mathrm{N}_\mathcal{A}(\mathrm{N}_\mathcal{A}(F)) = \mathrm{N}_\mathcal{A}(F)$ and $F_1 \leq F_2 \Rightarrow \mathrm{N}_\mathcal{A}(F_1) \leq \mathrm{N}_\mathcal{A}(F_2)$. Multi-clause-sets F with $\mathrm{N}_\mathcal{A}(F) = \top$ are called **$\mathcal{A}$-satisfiable**, while in case of $\mathrm{N}_\mathcal{A}(F) = F$ we call F **$\mathcal{A}$-lean**; the set of all $\mathcal{A}$-satisfiable multi-clause-sets is denoted by $\mathcal{SAT}_\mathcal{A}$, the set of all $\mathcal{A}$-lean multi-clause-sets by $\mathcal{LEAN}_\mathcal{A}$. It is F $\mathcal{A}$-lean iff $\mathcal{A}(F)$ contains no non-trivial autarky. The learn kernel $\mathrm{N}_\mathcal{A}(F)$ is the largest $\mathcal{A}$-lean sub-multi-clause-set of F. A sum of $\mathcal{A}$-lean multi-clause-sets again is $\mathcal{A}$-lean.

For the remainder of this subsection now assume that the autarky system $\mathcal{A}$ is normal. Then F is $\mathcal{A}$-satisfiable iff there exists $\varphi \in \mathcal{A}(F)$ with $\varphi * F = \top$. More generally, there always exists $\varphi \in \mathcal{A}(F)$ with $\varphi * F = \mathrm{N}_\mathcal{A}(F)$. If F is $\mathcal{A}$-lean, then so is $V * F$ for $V \in \mathbb{P}_\mathrm{f}(\mathcal{VA})$. The $\mathcal{A}$-autark sub-multi-clause-sets of F, i.e., those multi-clause-sets F' where there is $\varphi \in \mathcal{A}(F)$ with $F' = F_{\mathrm{var}(\varphi)}$, are exactly those F_V for some $V \subseteq \mathrm{var}(F)$ where $F[V]$ is $\mathcal{A}$-satisfiable. On the other hand, if F is $\mathcal{A}$-lean, then so is $F[V]$ for all $V \in \mathbb{P}_\mathrm{f}(\mathcal{VA})$. The set of $\mathcal{A}$-autark sub-multi-clause-sets of F is closed under union, and contains the smallest element $\top$ and the largest element $F \setminus \mathrm{N}_\mathcal{A}(F)$. As before, the relation between the $\mathcal{A}$-lean kernel of F and the largest $\mathcal{A}$-autark sub-multi-clause-set of F can be summarised as follows: For $F \in \mathcal{MCLS}$ there exist induced sub-multi-clause-sets $F_1, F_2 \leq F$ with $F_1 + F_2 = F$, such that F_1 is $\mathcal{A}$-lean, while $\mathrm{var}(F_1) * F_2$ is $\mathcal{A}$-satisfiable; in this decomposition F_1, F_2 are uniquely determined, namely $F_1 = \mathrm{N}_\mathcal{A}(F)$ is the largest $\mathcal{A}$-lean sub-multi-clause-set (the $\mathcal{A}$-lean kernel), while F_2 is the largest $\mathcal{A}$-autark sub-multi-clause-set.

We finish our review on autarkies and autarky systems by generalising Lemma 8.6 in [25]. The proof can be literally transferred to our generalised context, and thus is not reproduced here.

Lemma 3.1 *Let $\mathcal{A}$ be a normal autarky system. Given decision of membership in $\mathcal{LEAN}_\mathcal{A}$ as an oracle, the normal form $F \mapsto \mathrm{N}_\mathcal{A}(F)$ for $F \in \mathcal{MCLS}$ can be computed in polynomial time as follows*

1. *If $F \in \mathcal{LEAN}_\mathcal{A}$ then output F.*
2. *Let $\mathrm{var}(F) = \{v_1, \ldots, v_{n(F)}\}$.*
3. *Since $\emptyset * F = F \notin \mathcal{LEAN}_\mathcal{A}$ and $\mathrm{var}(F) * F = c(F) \cdot \check{\mathrm{t}}(\{\bot\}) \in \mathcal{LEAN}_\mathcal{A}$ holds, there is an index $1 \leq i \leq n(F)$ with*

$$\{v_1, \ldots, v_{i-1}\} * F \notin \mathcal{LEAN}_\mathcal{A} \quad \text{and} \quad \{v_1, \ldots, v_i\} * F \in \mathcal{LEAN}_\mathcal{A}.$$

Replace F by the induced sub-multi-clause-set of F given by the clauses of F not containing variable v_i, and go to Step 1.

3.5 Resolution

For autarky systems the number of occurrences of a clause in a multi-clause-set might make a difference (as it is the case for matching autarkies introduced in the subsequent section), however for all known resolution systems we do not need this distinction, and thus only (generalised) clause-sets are considered for resolution (that is, if multi-clause-sets $F \in \mathcal{MCLS}$ are to be treated, then they are automatically "downcast" to the underlying clause-set $\hat{\mathrm{t}}(F)$).

The resolution rule for generalised clause-sets is well-known. The most thorough study for my knowledge in given in [29], where actually resolution is considered for general "fipa-systems" (systems with finite instantiation by partial assignments) by reducing resolution for such axiomatic systems to resolution for generalised clause-sets, which act as "no-goods", i.e., out of the general system we get the clauses C belonging to the resolution refutation as clauses C_φ associated with such partial assignments, which led to a contradiction. In this subsection the most basic notions are reviewed, and the interesting connection to autarkies is given.

Consider a variable $v \in \mathcal{VA}$. "Parent clauses" $C_1, \ldots, C_{|D_v|}$ are called **resolvable with resolution variable** v, if $\mathrm{val}_v(\{C_1, \ldots, C_{|D_v|}\}) = D_v$ and the **resolvent** $R := \bigcup_{i=1}^{|D_v|} \{v\} * C_i$ actually is a clause (contains no clashing literals), that is, whenever there are literals $x \in C_i$, $y \in C_j$ for some $i, j \in \{1, \ldots, |D_v|\}$ with $x \neq y$ and $\mathrm{var}(x) = \mathrm{var}(y)$, then $\mathrm{var}(x) = v$ must be the case. Resolution is a complete and sound refutation system; see for example Corollary 5.9 in [29], where, translating branching trees into resolution trees, the existence of a resolution tree with at most $\mathrm{mds}(F)^{n(F)}$ many leaves for unsatisfiable generalised clause-sets F is shown. Also stated in [29] is the (well-known) "strong completeness" of resolution, that is, for a multi-clause-set $F \in \mathcal{MCLS}$ and a clause $C \in \mathcal{CL}$ we have $F \models \{C\}$ iff there exists a resolution tree with axioms from F deriving a clause $C' \subseteq C$.

In Theorem 3.16 in [23] it was shown for boolean clause-sets, that the lean kernel of a clause-set F consists exactly of all clauses $C \in F$ which can be used in some resolution refutation of F.[5)] This theorem can be immediately generalised to

[5)] Where the resolution refutation may not contain "dead ends", which can be most easily enforced by considering only resolution *trees*.

generalised clause-sets, using exactly the proof from [23] (together with the proof transformation tools provided in [29]). In [24], Theorem 4.1 this generalisation is stated, but without a proof, which we now outline as follows. Consider the set $U(F)$ of clauses $C \in F$ for which there exists a tree resolution refutation of F using C as an axiom. The direction, that a clause $C \in F \setminus \mathrm{N_a}(F)$ can not be used in tree resolution refutation of F (i.e., $U(F) \subseteq \mathrm{N_a}(F)$), is easily proved by induction (an autarky of F satisfying C satisfies also all clauses derived from C in the tree). For the reverse direction the main technical lemma is, that for each variable $v \in \mathrm{var}(U(F))$ and each $\varepsilon \in D_v$ the unit clause $\{(v, \varepsilon)\}$ can be derived from $U(F)$ by resolution (this is a little proof-theoretic exercise; see Lemma 3.14 in [23] for the boolean case). Now it follows, that $F \setminus U(F)$ is an autark sub-clause-set of F, since if the clause-set $\mathrm{var}(U(F)) * (F \setminus U(F))$ would be unsatisfiable, then there would be a tree resolution refutation T of $\mathrm{var}(U(F)) * (F \setminus U(F))$, where the axioms of T could be derived from the clauses in $F \setminus U(F)$ and the clauses in $U(F)$ by the above technical lemma, and thus we could construct a tree resolution refutation involving some clause of $F \setminus U(F)$ contradicting the definition of $U(F)$ (compare with Lemma 3.15 in [23] for the boolean case). That $F \setminus U(F)$ is an autark sub-clause-set of F means $\mathrm{N_a}(F) \subseteq U(F)$, and altogether we have shown

Theorem 3.2 *For any generalised clause-set $F \in \mathcal{CLS}$ the lean kernel $\mathrm{N_a}(F)$ equals the set $U(F)$ of clauses of F usable in some (tree) resolution refutation of F. Especially it is F lean if and only if $F = U(F)$, that is, if every clause of F can be used in some (tree) resolution refutation of F.*

As shown in Section 6 of [24], Theorem 3.2 yields an algorithm for computing $\mathrm{N_a}(F)$ by using "intelligent backtracking solvers", which on unsatisfiable instances can return the set of variables used in some resolution refutation of the input. Crossing out these variables from the input, removing the empty clause obtained, and repeating this process, we finally obtain a satisfiable clause-set F^*, and now any satisfying assignment φ for F^* with $\mathrm{var}(\varphi) \subseteq \mathrm{var}(F^*)$ is an autarky for F with $\varphi * F = F \setminus \mathrm{N_a}(F)$.

We conclude this subsection by defining the **Davis-Putnam operator** DP for generalised clause-sets. Consider a clause-set $F \in \mathcal{CLS}$ and a variable $v \in \mathrm{var}(F)$. Let F_v be the set of all resolvents of parent clauses in F with resolution variable v. Now we set $\mathbf{DP_v(F)} := \{C \in F : v \notin \mathrm{var}(C)\} \cup F_v$. From the completeness results for (generalised) resolution in [29] it follows immediately, that $\mathrm{DP}_v(F)$ is satisfiability equivalent to F, and that F is unsatisfiable if and only if by repeated applications of the Davis-Putnam operator we finally obtain the clause-set $\{\bot\}$ (while for satisfiable F finally we will obtain the clause-set $\top$). We can also generalise Lemma 7.6 in [30] about the commutativity of the Davis-Putnam operator, that is, if G_1 is the result of applying first DP_{v_1}, then DP_{v_2}, ..., and finally applying DP_{v_m}, while G_2 is the result of applying first $\mathrm{DP}_{v_{\pi(1)}}$, then $\mathrm{DP}_{v_{\pi(2)}}$, ..., and finally applying $\mathrm{DP}_{v_{\pi(m)}}$, for some permutation $\pi \in S_m$, then after elimination of subsumed clauses in G_1 and G_2 (see the following subsection) G_1 becomes equal to G_2. It follows that for some set of variables V the operator DP_V, computed by running through the variables of V in some order, is uniquely determined up to subsumption reduction in the result. We always have $\mathrm{DP}_V(F) = \mathrm{DP}_V(F_V) \cup (F \setminus F_V)$. If for some $V \subseteq \mathrm{var}(F)$ we have $\mathrm{DP}_V(F_V) = \top$, then F and $F \setminus F_V$ are satisfiability equivalent, generalising the elimination of autark sub-clause-sets: If $\varphi \in \mathrm{Auk}(F)$, then $\mathrm{DP}_{\mathrm{var}(\varphi)}(F_{\mathrm{var}(\varphi)}) = \top$, while the reverse direction need not hold, as the example $F = \{\{v, a\}, \{\overline{v}, \overline{a}\}\} \cup F'$, $v \notin \mathrm{var}(F')$, with $V = \{v\}$ shows

(for boolean variables). We see, that the Davis-Putnam operator, whose application for generalised clause-sets is basically the same as existential quantification, yields more powerful reductions, but this at the cost of potential exponential space usage.

3.6 Reductions

In this subsection we review the most basic polynomial time reduction concepts. For a thorough discussion in the boolean case, see [30]. We consider only clause-sets $F \in \mathcal{CLS}$, but all results are easily generalised to multi-clause-sets.

The most basic reduction (by which we mean a satisfiability-equivalent transformation, simplifying the clause-set in some sense) is **subsumption elimination**, the elimination of subsumed clauses, i.e., the transition $F \to F \setminus \{C\}$ for $C \in F$ in case there exists $C' \in F$ with $C' \subset C$. Iterated elimination of subsumed clauses is confluent, and thus the result of applying subsumption elimination as long as possible is uniquely determined (namely it is the set of all minimal clauses of F); if F has no subsumed clauses, then we call F **subsumption-free**.

The next reduction can be called the **trivial-domain reduction**: If there exists $v \in \mathrm{var}(F)$ with $|D_v| = 1$, then for $D_v = \{\varepsilon\}$ reduce $F \mapsto \langle v \to \varepsilon \rangle * F$.

Elimination of "pure literals" is now better called **elimination of pure variables**: If there is $v \in \mathrm{var}(F)$ with $|\mathrm{val}_v(F)| < |D_v|$, then for some $\varepsilon \in D_v \setminus \mathrm{val}_v(F)$ reduce $F \mapsto \langle v \to \varepsilon \rangle * F$. This is the basic form of a **pure autarky** as mentioned in Subsection 4.4 of [24].

Unit-clause-elimination for generalised clause-sets is less powerful than in the boolean case: If F contains a unit-clause $\{(v, \varepsilon)\} \in F$, then in case of $D_v = \{\varepsilon\}$ by trivial-domain reduction we conclude that F is unsatisfiable, but otherwise we can only conclude that value ε is to be excluded from the domain of v, while in general we cannot eliminate the variable v. In our context the simplest way to handle this situation seems to me to eliminate all clauses containing the literal (v, ε) from F, and to replace variable v in the remaining occurrences by a new variable v' with $D_{v'} = D_v \setminus \{\varepsilon\} \neq \emptyset$. The effect of unit clause elimination in the boolean case is obtained when combining this generalised form of unit clause elimination with trivial domain reduction. Ordinary repeated unit clause elimination (aka unit clause propagation) is confluent when combined with subsumption elimination in case the empty clause is created; now the generalised form of unit clause elimination combined with trivial-domain reduction is confluent modulo renaming (again using subsumption elimination if the empty clause is created by trivial-domain reduction).

Finally we consider the most harmless cases for **DP-reductions**. In general, application of DP_v to F eliminates $\#_v(F) = \sum_{\varepsilon \in D_v} \#_{(v,\varepsilon)}(F)$ clauses and creates up to $\prod_{\varepsilon \in D_v} \#_{(v,\varepsilon)}(F)$ new clauses (less iff some of the parent clause combinations are not eligible for resolution due to additional clashes). Thus we have

$$c(\mathrm{DP}_v(F)) \leq c(F) - \sum_{\varepsilon \in D_v} \#_{(v,\varepsilon)}(F) + \prod_{\varepsilon \in D_v} \#_{(v,\varepsilon)}(F). \tag{1}$$

If in (1) we have a strict inequality or v is a pure variable for F, then we call v a **degenerated DP-variable w.r.t.** F, while otherwise v is called a **non-degenerated DP-variable w.r.t.** F. Note that a missing new clause due to additional clashes is not the only cause of a strict inequality, but it is also possible that a resolvent is already contained in the rest of F (and thus contraction occurs). If variable $v \in \mathrm{var}(F)$ has a trivial domain (i.e., $|D_v| = 1$), then v is a non-degenerated

DP-variable with $c(\mathrm{DP}_v(F)) = c(F)$, and $\mathrm{DP}_v(F)$ is the result of applying trivial domain reduction to F, while if v is pure w.r.t. F, then F is a degenerated DP-variable with $c(\mathrm{DP}_v(F)) = c(F) - \#_v(F)$, and $\mathrm{DP}_v(F)$ is the result of applying elimination of pure variables to F. Besides these cases, in this article we consider only one very restricted form of DP-resolution, characterised by the condition that at most one of the factors in the product from (1) might be greater than one:

We call a variable v a **singular DP-variable** w.r.t. F if there exists $\varepsilon \in D_v$ such that for all $\varepsilon' \in D_v \setminus \{\varepsilon\}$ we have $\#_{(v,\varepsilon')}(F) \leq 1$. In such a case of a singular DP-variable, application of DP_v eliminates $|D_v| - 1 + \#_{(v,\varepsilon)}(F)$ clauses and creates up to $\#_{(v,\varepsilon)}(F)$ new clauses, so that the number of clauses goes down at least by one if $|D_v| \neq 1$. If a singular DP-variable v is non-degenerated then we have $c(\mathrm{DP}_v(F)) = c(F) - |D_v| + 1$. If v is a degenerated singular DP-variable, then at least one of the clauses containing v can be eliminated satisfiability-equivalently, and we call such a clause elimination a **singular DP-degeneration reduction**. Since a singular DP-degeneration reduction cannot be applied to a minimally unsatisfiable clause-set, a singular variable w.r.t. a minimally unsatisfiable clause-set must be non-degenerated.

In the boolean case, such applications of DP-reduction are used at many places in the literature (in [22], Appendix B, the application of DP_v for non-degenerated singular DP-variables v is called "$(1,\infty)$-reduction" (the boolean case)). We conclude by another reduction arising from the DP-operator. The notion of **blocked clauses** for boolean clause-sets (see [20, 21]) can be generalised by calling a clause C **blocked w.r.t.** F if there exists a variable $v \in \mathrm{var}(C)$ with $\mathrm{DP}_v(F \cup \{C\}) = \mathrm{DP}_v(F \setminus \{C\})$. If $C \in F$ is blocked w.r.t. F, then F is satisfiability equivalent to $F \setminus \{C\}$, and such a reduction is called **elimination of blocked clauses**. If v is a pure variable for F, then all clauses of F containing variable v are blocked w.r.t. F. And if v is a degenerated singular DP-variable, then F has at least one blocked clause containing v, and so singular DP-degeneration reduction is also covered by elimination of blocked clauses.

3.7 Conflict structure

A study of the "combinatorics of conflicts" for boolean clause-sets has been initiated with [26, 27] and continued with [15, 28]. We generalise here only a very few simple notions used later in this article.

The **conflict multigraph cmg(F)** of a multi-clause-set $F \in \mathcal{MCLS}$ has as vertices the pairs (C, i) for $C \in F$ and $i \in \{1, \ldots, F(C)\}$, and as many edges joining two vertices $(C, i), (D, j)$ as there are **clashing literals** between C and D, that is, the number of parallel edges joining (C, i) and (D, j) is the number of literals $x \in C$ for which there exists a literal $y \in D$ with $\mathrm{var}(x) = \mathrm{var}(y)$ and $x \neq y$. The **conflict graph cg(F)** is the graph underlying $\mathrm{cmg}(F)$. A clause-set F is called a **hitting clause-set** if the conflict graph of F is a complete graph (note that if a multi-clause-set F has a complete conflict graph, then F actually "is" a clause-set). More generally a clause-set F is called **multihitting** if the conflict graph of F is complete k-partite for some $k \in \mathbb{N}_0$ (it is F hitting iff $k = c(F)$). For a given multihitting clause-set F there is a unique partition $\mathbb{F}$ of F (that is, $\mathbb{F}$ is a set of sub-clause-sets of F which are non-empty and pairwise disjoint, such that their union is F), so that for any clauses $C_1, C_2 \in F$ with $C_i \in F_i \in \mathbb{F}$ for $i \in \{1, 2\}$ the clauses C_1 and C_2 clash if and only if $F_1 \neq F_2$; we call $\mathbb{F}$ the **multipartition** of F (if F is bihitting, then $\mathbb{F}$ is also called the **bipartition** of F).

4 Matching autarkies

In this section we introduce the autarky system for generalised clause-sets given by "matching autarkies", and we show various polynomial time procedures. In case of a pure variable $v \in \mathrm{var}(F)$ for some $F \in \mathcal{MCLS}$ (that is, not all values $\varepsilon \in D_v$ are used in F) we assume that D_v contains exactly one value not used in F (i.e., $|D_v| = |\mathrm{val}_v(F)| + 1$); in this way we are not troubled anymore by the unknown domain size D_v, but we can measure the size of F just by $\ell(F)$, while this modification has no influence on any of the notions and procedure in this article.

4.1 Matching satisfiable generalised clause-sets

We whish to generalise the notion of "matching satisfiable clause-sets", introduced in [25] for boolean clause-sets. Consider a multi-clause-set F together with a decomposition $F = F_1 + \cdots + F_m$ for $m \in \mathbb{N}_0$ and $F_i \in \mathcal{MCLS}$, fulfilling the following conditions:

(i) for $i \in \{1, \ldots, m\}$ there are variables $v_i \in \mathrm{var}(F_i)$ such that for all $C \in F_i$ we have $v_i \in \mathrm{var}(C)$;

(ii) the variables $v_1, \ldots, v_m$ are pairwise different;

(iii) for all $i \in \{1, \ldots, m\}$ we have $|D_{v_i}| > |\mathrm{val}_v(F_i)|$.

Given such a decomposition, we see that F is satisfiable, since for each i there exists $\varepsilon_i \in D_{v_i} \setminus \mathrm{val}_v(F_i)$, and the assignment $\langle v_i \to \varepsilon_i : i \in \{1, \ldots, m\}\rangle$ is a satisfying assignment for F (note that none of the variables v_i needs to be a pure variable in F). If we consider on the other side a partial assignment φ satisfying F with $\mathrm{var}(\varphi) = \{v_1, \ldots, v_m\}$, and set F_i for $i \in \{1, \ldots, m\}$ as the induced sub-multi-clause-set of F given by the clauses $C \in F$ with $v_i \in \mathrm{var}(C)$ and $(v_i, \varphi(v_i)) \notin C$, then we obviously fulfil the above conditions, and we see that conditions (i) - (iii) need to be restricted so that we can obtain a class of satisfiable clause-sets which is decidable in polynomial time. Now we have $c(F_i) \geq |\mathrm{val}_v(F_i)|$, and thus condition

(iii)' for all $i \in \{1, \ldots, m\}$ we have $|D_{v_i}| > c(F_i)$

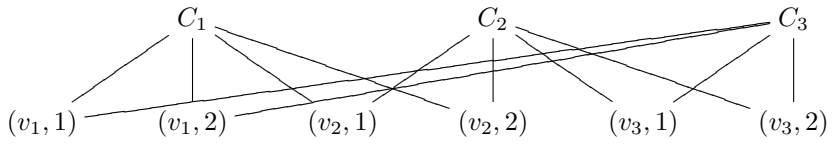
strengthens condition (iii). We call multi-clause-sets $F \in \mathcal{MCLS}$ having a decomposition $F = F_1 + \cdots + F_m$ fulfilling conditions (i), (ii) and (iii)' **matching satisfiable**, and the set of all matching satisfiable (generalised) multi-clause-sets is denoted by $\boldsymbol{\mathcal{MSAT}}$.

To understand the connection to matching problems, we introduce the bipartite graph $B(F)$ for generalised multi-clause-sets $F \in \mathcal{MCLS}$:

- Let
$$\begin{aligned} \underline{F} &:= \{\, (C, i) : C \in F,\ i \in \{1, \ldots, F(C)\} \,\} \\ \underline{V} &:= \{\, (v, j) : v \in \mathrm{var}(F),\ j \in \{1, \ldots, |D_v| - 1\} \,\}. \end{aligned}$$
- the vertex set of $B(F)$ is defined as $V(B(F)) := \underline{F} \uplus \underline{V}$; (the elements of $\underline{F}$ are called the *clause-nodes*, while the elements of $\underline{V}$ are called the *variable-nodes*)
- the edge set $E(B(F))$ is the set of all (undirected) edges $\{(C, i), (v, j)\}$ over $V(B(F))$ such that $v \in \mathrm{var}(C)$.

In other words, the graph $B(F)$ has as vertices $F(C)$-many copies of clauses $C \in F$ together with $(|D_v| - 1)$-many copies of variables $v \in \mathrm{var}(F)$, while edges connect copies of variables v with copies of clauses C such that $v \in \mathrm{var}(C)$. The canonical bipartition of $B(F)$ is $(\underline{F}, \underline{V})$. Consider for example the clause-set $F = \{C_1, C_2, C_3\}$ with $C_1 = \{(v_1, a), (v_2, a)\}$, $C_2 = \{(v_2, b), (v_3, b)\}$, $C_3 = \{(v_3, c), (v_1, c)\}$, where $D_{v_i} = \{a, b, c\}$. Now $B(F)$ is (suppressing the indices for the clause-copies, since here we just have a clause-set):

$B(F) =$



For a set V of variables we obtain $B(V * F)$ from $B(F)$ by deleting the variable-nodes (v, j) of $B(F)$ with $v \in V$, while $B(F[V])$ is the induced subgraph of $B(F)$ given by the variable-nodes (v, j) of $B(F)$ with $v \in V$ together with their neighbours (those clause-nodes (C, i) with $\mathrm{var}(C) \cap V \neq \emptyset$).

Using the **weighted number of variables $\mathrm{wn}(F)$** $:= \sum_{v \in \mathrm{var}(F)} (|D_v| - 1) \in \mathbb{N}_0$, the number of vertices of $B(F)$ is $|V(B(F))| = c(F) + \mathrm{wn}(F)$, while the number of edges is $|E(B(F))| = \sum_{v \in \mathrm{var}(F)} \#_v(F) \cdot (|D_v| - 1)$. We have $\mathrm{wn}(F) = (\sum_{v \in \mathrm{var}(F)} |D_v|) - n(F)$. If F is boolean, then $\mathrm{wn}(F) = n(F)$.

By definition a multi-clause-set F is matching satisfiable iff there exists a matching in $B(F)$ covering all vertices of $\underline{F}$. Let the **deficiency** of a (generalised) multi-clause-set F be defined as $\boldsymbol{\delta(F)} := c(F) - \mathrm{wn}(F) \in \mathbb{Z}$, while the **maximal deficiency** is defined as $\boldsymbol{\delta^*(F)} := \max_{F' \leq F} \delta(F') \in \mathbb{N}_0$ (we have $\delta^*(F) \geq 0$ due to $\delta(\top) = 0$; by definition we have $\delta^*(F) \leq c(F)$). Considering $F' \leq F$ as a subset of $\underline{F}$, the deficiency $\delta(F')$ of $F' \leq F$ is just the deficiency of this subset in $B(F)$ (as we have defined it for arbitrary graphs). By well-known results from graph theory, the maximal number of nodes of $\underline{F}$ coverable by some matching thus is $c(F) - \delta^*(F)$ (see for example Theorem 22.2 in [34], where the notion of "transversals" or "systems of distinct representatives" is used (not to be mixed up with "transversals" in hypergraphs): the set system is $\underline{F}$, but each literal (v, ε) in a clause C is replaced by the $|D_v| - 1$ copies of variable v; the transversals of this set system correspond to the matchings in $B(F)$). Summarising we have (generalising Lemma 7.2 in [25]):

Lemma 4.1 *Consider a generalised multi-clause-set $F \in \mathcal{MCLS}$.*

1. *The maximal size of a matching satisfiable sub-multi-clause-set $F' \leq F$ is $c(F') = c(F) - \delta^*(F)$.*
2. *F is matching satisfiable if and only if $\delta^*(F) = 0$.*

As an application we can generalise the well-known fact, apparently first mentioned in the literature in [37], that if a boolean clause-set F has minimal clause-length k and maximal variable occurrence k for some $k \geq 1$, then F must be satisfiable (see [19] for recent further developments):

Corollary 4.2 *Consider a generalised clause-set $F \in \mathcal{CLS}$ with $\mathrm{var}(F) \neq \emptyset$. Then*

$$\frac{\max_{v \in \mathrm{var}(F)} \#_v(F)}{\min_{C \in F} |C|} \leq \min_{v \in \mathrm{var}(F)} |D_v| - 1 \Longrightarrow F \in \mathcal{MSAT}.$$

Proof: Assume the condition holds, and consider $F' \subseteq F$. We have to show $\delta(F') \leq 0$. Let $d := \min_{v \in \mathrm{var}(F)} |D_v|$. Then $\delta(F') \leq c(F') - (d-1)n(F')$, and a sufficient condition for $\delta(F') \leq 0$ is $\frac{c(F')}{n(F')} \leq d-1$. Let $a := \max_{v \in \mathrm{var}(F)} \#_v(F)$ and $b := \min_{C \in F} |C|$. We know $c(F') \cdot b \leq \ell(F') \leq n(F') \cdot a$, and thus $\frac{c(F')}{n(F')} \leq \frac{a}{b}$. ∎

Since matchings of maximal size can be computed in polynomial time (see Chapter 16 in [34]), we get the following poly-time results:

Lemma 4.3 *For every generalised clause-set $F \in \mathcal{MCLS}$, in polynomial time in $\ell(F)$ we can compute $F' \leq F$ with $F' \in \mathcal{MSAT}$ such that $c(F')$ is maximal. Since $F' = F$ iff F is matching satisfiable, it follows that whether F is matching satisfiable or not is decidable in polynomial time. And due to $c(F') = c(F) - \delta^*(F)$ the maximal deficiency $\delta^*(F)$ is computable in polynomial time.*

4.2 Satisfying assignments versus matching satisfying assignments

Consider a (generalised) multi-clause-set $F \in \mathcal{MCLS}$ and a partial assignment $\varphi \in \mathcal{PASS}$. The partial subgraph $B_\varphi(F)$ of $B(F)$ is obtained from $B(F)$ by eliminating all edges $\{(C,i),(v,j)\}$ such that for the literal $(v,\varepsilon) \in C$ we have $\varphi((v,\varepsilon)) = 0$ (i.e., $\varphi(v) = \varepsilon$). Now φ is called a **matching-satisfying assignment** for F if $B_\varphi(F)$ contains a matching covering all clause-vertices (thus matching satisfying assignments are satisfying assignments). Obviously F is matching satisfiable iff there exists a matching satisfying assignment for F. The following two lemmas give simple basic properties regarding this notion.

Lemma 4.4 *Consider a generalised multi-clause-set $F \in \mathcal{MCLS}$ and a partial assignment $\varphi \in \mathcal{PASS}$.*

1. *If φ is satisfying for F, then there exists $\varphi' \subseteq \varphi$ with $n(\varphi') \leq c(F)$ such that also φ' is satisfying for F.*
2. *If φ is matching satisfying for F, then there exists $\varphi' \subseteq \varphi$ with $n(\varphi') = c(F)$ such that also φ' is matching-satisfying for F.*
3. *If φ is satisfying for F, and there is no $\varphi' \subseteq \varphi$ with $n(\varphi') < c(F)$ such that φ' is satisfying for F, then φ is matching-satisfying for F.*

Proof: Assertions 1 and 2 follow by definition, while assertion 3 follows by Hall's criterion. ∎

Lemma 4.5 *Consider a generalised multi-clause-set $F \in \mathcal{MCLS}$ and partial assignments $\varphi, \psi \in \mathcal{PASS}$. If $\varphi \circ \psi$ is matching-satisfying for F, then φ is matching-satisfying for $\psi * F$.*

The main result of this subsection is the following theorem.

Theorem 4.6 *Consider a satisfiable generalised multi-clause-set $F \in \mathcal{MCLS}$. There exists a satisfying assignment φ for F and a sub-multi-clause-set $F' \leq F$ with $c(F') = c(F) - \delta^*(F)$, such that φ is matching-satisfying for F'.*

We obtain the following generalisation of Theorem 7.16 in [25]:

Corollary 4.7 *For every satisfiable generalised multi-clause-set $F \in \mathcal{MCLS}$ there exists a partial assignment $\varphi \in \mathcal{PASS}$ with $n(\varphi) \leq \delta^*(F)$ such that $\varphi * F$ is matching satisfiable.*

Proof: By Theorem 4.6 there exists a satisfying assignment φ_0 for F and $F' \leq F$ with $c(F') = c(F) - \delta^*(F)$, such that φ_0 is matching-satisfying for F'. Let $F'' := F - F'$. We have $c(F'') = \delta^*(F)$ and φ_0 is satisfying for F'', so by Lemma 4.4, Part 1 there exists $\varphi \subseteq \varphi_0$ with $n(\varphi) \leq \delta^*(F)$ such that φ is satisfying for F''. Now $\varphi_0 = \varphi_0 \circ \varphi$ is matching-satisfying for F', and thus by Lemma 4.5 it is φ_0 matching-satisfying for $\varphi * F'$, where $\varphi * F = \varphi * (F' + F'') = \varphi * F' + \varphi * F'' = \varphi * F'$, and thus φ_0 is matching-satisfying for $\varphi * F$. ∎

Corollary 4.8 *The satisfiability problem for generalised multi-clause-sets F with $\delta^*(F) \leq k$ for constant $k \in \mathbb{N}_0$ is decidable in polynomial time (and if F is satisfiable, then a satisfying assignment can be computed).*

In [36] it was shown, that in the boolean case the satisfiability problem for bounded maximal deficiency actually is fixed-parameter tractable. Whether this can be extended also to non-boolean clause-sets is not known to me, and seems to be an interesting problem.

The remainder of this subsection is devoted to the proof of Theorem 4.6 (generalising and simplifying the results on "admissible matchings" in [10]). We remind at the notion of a matching M in a graph G, which is a set $M \subseteq E(G)$ of edges such that two distinct elements of M are non-adjacent; a *maximal matching* is one which can not be extended, while a *maximum matching* is a matching of maximal size. The vertices covered by M are the vertices incident to one of the edges in M. We begin with two auxiliary lemmas, using the following notions: Consider partial assignments φ, φ' with $\mathrm{var}(\varphi), \mathrm{var}(\varphi') \supseteq \mathrm{var}(F)$; we call φ' a *good neighbour* of φ w.r.t. F, if there is exactly one variable $v \in \mathrm{var}(F)$ with $\varphi(v) \neq \varphi'(v)$, and every clause of F falsified by φ' is also falsified by φ.

Lemma 4.9 *Consider a generalised multi-clause-set $F \in \mathcal{MCLS}$, a partial assignment $\varphi \in \mathcal{PASS}$, a matching M in $B_\varphi(F)$ and an edge $\{v, C\} \in E(B_\varphi(F))$ such that neither v nor C is covered by M. We assume furthermore, that if there exists a matching $M^* \supset M$ in $B_\varphi(F)$, then M^* does not cover v. Then there exists a good neighbour φ' of φ w.r.t. F, such that $M' := M \cup \{\{v, C\}\}$ is a matching in $B_{\varphi'}(F)$.*

Proof: Let v_0 be the underlying variable of variable-node v, and C_0 the underlying clause of clause-node C. Let E be the set of values $\varepsilon \in D_{v_0}$, such that there is an edge $\{(v_0, i), (A, j)\} \in M$ with $(v_0, \varepsilon) \in A$. Since v is not covered by M, we have $|E| \leq |D_{v_0}| - 2$, and thus there are $\varepsilon_1, \varepsilon_2 \in D_{v_0} \setminus E$, $\varepsilon_1 \neq \varepsilon_2$. W.l.o.g. we can assume $\varphi(v_0) = \varepsilon_1$ and $(v_0, \varepsilon_1) \in C_0$ (otherwise M' would be a matching in $B_\varphi(F)$ covering v). Set $\varphi' := \varphi \circ \langle v_0 \to \varepsilon_2 \rangle$. By definition it is M' a matching in $B_{\varphi'}(F)$. Now consider a clause $A \in F$ falsified by φ', and assume that A is not falsified by φ. Thus $(v_0, \varepsilon_2) \in A$, and the literal (v_0, ε_2) is the only literal in A satisfied by φ. So no clause-node covered by M has clause A associated with it. It follows that $M^* := M \cup \{\{v, (A, 1)\}\}$ is a matching in $B_\varphi(F)$ extending M and covering v, contradicting the assumption. ∎

Lemma 4.10 *Consider a generalised multi-clause-set $F \in \mathcal{MCLS}$, a partial assignment $\varphi \in \mathcal{PASS}$, a maximal matching M' in $B_\varphi(F)$ and an edge $\{v, C\} \in$*

$E(B_\varphi(F))$ such that v is not covered by M', while C is covered by M' and thus there is a (unique) edge $\{C, w\} \in M'$. Then there exists a partial assignment φ', such that either $\varphi' = \varphi$ or φ' is a good neighbour φ' of φ w.r.t. F, and such that $M'' := (M' \cup \{\{v, C\}\}) \setminus \{\{C, w\}\}$ is a matching in $B_{\varphi'}(F)$ with $|M''| = |M'|$, variable-node w is not covered by M'', and the clause-nodes covered by M'' are exactly the clause-nodes covered by M'.

Proof: Let v_0 be the underlying variable of variable-node v, C_0 the underlying clause of clause-node C, and let $(v_0, \varepsilon) \in C_0$. If $\varphi((v_0, \varepsilon)) = 1$, then let $\varphi' := \varphi$ and we are done. Otherwise let $M := M' \setminus \{\{C, w\}\}$ and apply Lemma 4.9. ∎

We further remind at the notion of an *M-augmenting path* for a matching M in a graph G (see Section 16.1 in [34]), which is a path of odd length with endpoints not covered by M and whose edges are alternatively out of and in M.

Lemma 4.11 *Consider a generalised multi-clause-set $F \in \mathcal{MCLS}$, a partial assignment $\varphi \in \mathcal{PASS}$, a matching M in $B_\varphi(F)$ and an M-augmenting path in $B(F)$ (note that here we can use all edges). Then we can construct (in polynomial time) partial assignments $\varphi_0, \ldots, \varphi_m$, $m \in \mathbb{N}_0$, and a matching M^+ in $B_{\varphi_m}(F)$ with $|M^+| = |M| + 1$, such that $\varphi_0 = \varphi$ and φ_i is a good neighbour of φ_{i-1} for $i \in \{1, \ldots, m\}$.*

Proof: In the following construction we will construct $\varphi_0, \ldots, \varphi_{m'}$ for some $m' \in \mathbb{N}_0$, fulfilling the above conditions but allowing $\varphi_i = \varphi_{i+1}$ for some i; $m \le m'$ and the final list of partial assignments is obtained by removing identical neighbours from the list $\varphi_0, \ldots, \varphi_{m'}$. W.l.o.g. we can assume that the augmenting path P is of the form $P = (v_1, C_1, v_2, C_2, \ldots, v_k, C_k)$, $k \in \mathbb{N}$, where the v_i are (distinct) variable-nodes, and the C_i are (distinct) clause-nodes. We construct now partial assignments φ_i and matchings M_i in $B_{\varphi_i}(F)$ for $i = 0, \ldots, m'$, where $0 \le m' < k$ is determined by the following construction:

1. Set $\varphi_0 := \varphi$ and $M_0 := M$.
2. Set $i := 1$.
3. While $i < k$ do
   (a) if matching M_{i-1} is not maximal in $B_{\varphi_{i-1}}(F)$, then extend M_{i-1} by one edge to obtain M^+, and with $m' := i - 1$ we are done with the whole construction;
   (b) otherwise apply Lemma 4.10 with $M' := M_{i-1}$, $\varphi := \varphi_{i-1}$ and $v := v_i$, $C := C_i$, and let $M_i := M''$ and $\varphi_i := \varphi'$;
   (c) set $i := i + 1$.
4. After completing the while-loop (we now have $i = k$), apply Lemma 4.9 with $M := M_{i-1}$, $\varphi := \varphi_{i-1}$ and $v := v_i$, $C := C_i$, and we set $m' := i$, $\varphi_i := \varphi'$ and $M^+ := M'$. ∎

Using the fact, that if a matching in a graph is not maximum, then it has an augmenting path (see Theorem 16.1 in [34]), we prove Theorem 4.6 as follows: Start with any satisfying assignment φ_0 for F and the empty matching M_0. Apply Lemma 4.11 repeatedly until a maximum matching M^+ in $B(F)$ is obtained, and set $\varphi := \varphi_m$.

4.3 Matching autarkies for generalised clause-sets

A partial assignment φ is called a **matching autarky** for $F \in \mathcal{MCLS}$ if φ is matching-satisfying for $F_{\mathrm{var}(\varphi)}$. The set of all matching autarkies for F is denoted by $\mathbf{MAuk}(\boldsymbol{F})$. Generalising Lemma 7.1 and the remarks in Section 8 of [25] we get

Lemma 4.12 *It is $F \in \mathcal{MCLS} \mapsto \mathrm{MAuk}(F) \subseteq \mathrm{Auk}(F)$ a normal autarky system.*

We denote by $\mathbf{N_{ma}} := \mathrm{N}_{\mathrm{MAuk}}$ the normal form for multi-clause-sets obtained by eliminating all matching autarkies. According to our general results and definitions on autarky systems, the set of MAuk-satisfiable multi-clause-sets is just $\mathcal{MSAT}$, the set of matching satisfiable multi-clause-sets. The set of MAuk-lean clause-sets is denoted by $\boldsymbol{\mathcal{MLEAN}}$, its elements are called **matching lean** multi-clause-sets. We now seek to characterise $\mathcal{MLEAN}$, and to compute $\mathrm{N}_{\mathrm{ma}}(F)$ in polynomial time.

A sub-multi-clause-set $F' \leq F$ of a multi-clause-set $F \in \mathcal{MCLS}$ is called **tight** if $\delta(F') = \delta^*(F)$ holds. If F' is tight for F, then F' is an induced sub-multi-clause-set of F. By super modularity of the deficiency (for graphs) we immediately get

Lemma 4.13 *Union and intersection of tight sub-multi-clause-sets of a multi-clause-set are again tight.*

Generalising Lemma 7.3 in [25], we obtain the fundamental relation between tight sub-multi-clause-sets and matching autarkies:

Lemma 4.14 *Consider a generalised multi-clause-set $F \in \mathcal{MCLS}$.*

1. *For every autarky φ for F we have $\delta(\varphi * F) = \delta(F) - \delta(F[\mathrm{var}(\varphi)])$.*
2. *For every matching autarky φ for F we have $\delta(\varphi * F) \geq \delta(F)$.*
3. *Consider an induced sub-multi-clause-set F' of F.*
   (a) *$\delta^*(\mathrm{var}(F') * (F - F')) \leq \delta^*(F) - \delta(F')$.*
   (b) *If F' is tight, then there is a matching autarky φ for F with $\varphi * F = F'$.*

Proof: For Part 1 note that by definition we have

$$c(F) = c(\varphi * F) + c(F[\mathrm{var}(\varphi)]),\ n(F) = n(\varphi * F) + n(F[\mathrm{var}(\varphi)])$$

due to $F = \varphi * F + F_{\mathrm{var}(\varphi)}$. Part 2 follows from Part 1. For Part 3a consider $G \leq \mathrm{var}(F') * (F - F')$. There exists $G_0 \leq F - F'$ with $\mathrm{var}(F') * G_0 = G$. Now

$$\delta^*(F) \geq \delta(F' + G_0) = c(F' + G_0) - \mathrm{wn}(F' + G_0) = \\ c(F') + c(G) - \mathrm{wn}(F') - \mathrm{wn}(G) = \delta(F') + \delta(G),$$

and thus $\delta(G) \leq \delta^*(F) - \delta(F')$. Now Part 3b follows immediately from Part 3a due to $\delta^*(\mathrm{var}(F') * (F - F')) \leq \delta^*(F) - \delta(F') = 0$, i.e., $F - F'$ is a matching autark sub-multi-clause-set of F. ∎

Generalising Theorem 7.5 in [25], we now can characterise matching lean multi-clause-sets:

Lemma 4.15 *Consider a generalised multi-clause-set $F \in \mathcal{MCLS}$. The following conditions are equivalent:*

1. *F is matching lean;*
2. *$\forall\, C \in F : \delta^*(F - \{C\}) < \delta^*(F)$;*
3. *$\forall\, F' \lneq F : \delta(F') < \delta(F)$;*
4. *F is a tight sub-multi-clause-set of F, and there are no other tight sub-multi-clause-sets of F.*

Proof: From Part 1 follows Part 4 by Lemma 4.14, Part 3b. Obviously, Part 4 implies Part 3, and Part 3 implies Part 2. Finally, Part 1 follows from Part 2 by Lemma 4.14, Part 2. ∎

By Lemma 4.15, Part 2 we get

Corollary 4.16 *It is decidable in polynomial time, whether a generalised multi-clause-set $F \in \mathcal{MCLS}$ is matching lean.*

Thus by Lemma 3.1:

Corollary 4.17 *The matching lean kernel $\mathrm{N}_{\mathrm{ma}}(F)$ for generalised multi-clause-sets $F \in \mathcal{MCLS}$ is computable in polynomial time.*

By Lemma 4.15, Part 4 together with Lemma 4.14, Part 3b we get

Corollary 4.18 *For every generalised multi-clause-set $F \in \mathcal{MCLS}$ the lean kernel $\mathrm{N}_{\mathrm{ma}}(F)$ is the intersection of all tight sub-multi-clause-sets of F (thus $\mathrm{N}_{\mathrm{ma}}(F)$ is itself tight, i.e., $\delta(\mathrm{N}_{\mathrm{ma}}(F)) = \delta^*(\mathrm{N}_{\mathrm{ma}}(F)) = \delta^*(F)$).*

Using $\delta(\top) = 0$, from Lemma 4.15, Part 3 we get the following generalisation of "Tarsi's Lemma" (see [1]):

Corollary 4.19 *If the generalised multi-clause-set $F \neq \top$ is matching lean, then $\delta(F) \geq 1$.*

Obviously $\mathcal{MUSAT} \subset \mathcal{LEAN}$, and thus:

Corollary 4.20 *If a generalised clause-set $F \in \mathcal{CLS}$ is minimally unsatisfiable, then we have $\delta(F) \geq 1$.*

In [6], Theorem 4.5, arbitrary constraints over boolean variables are considered, and a lower bound on the number of clauses in terms of the number of variables for minimally unsatisfiable constraint satisfaction problems is derived, which necessarily is much weaker than Corollary 4.20.

Removing any clause from a matching lean multi-clause-set F with $\delta(F) = 1$ yields a matching satisfiable multi-clause-set, and thus

Corollary 4.21 $\mathcal{MUSAT}_{\delta=1} = \mathcal{MLEAN}_{\delta=1} \cap \mathcal{USAT}$.

5 Minimally unsatisfiable generalised clause-sets

On of the main motivation for the notion of "lean clause-sets" is, that in this way we get a "smooth" and flexible generalisation of the "rigid" notion of minimally unsatisfiable clause-sets. In this section we will consider some of the basic facts on minimally unsatisfiable clause-sets in our generalised setting. We start in Subsection 5.1 with a discussion of the notion of "irredundant clause-sets" (a notion applicable also to satisfiable clause-sets). In Subsection 5.2 we consider the in some sense most extreme case of irredundant clause-sets, namely "hitting clause-sets", and the natural generalisation to "multihitting clause-sets", while in Subsection 5.3 "saturated minimally unsatisfiable clause-sets" are discussed; here we see a concrete example, where generalised clause-sets behave essentially more complicated than boolean clause-sets.

5.1 Irredundant clause-sets

A clause $C \in F$ is called **redundant** for clause-set $F \in \mathcal{CLS}$ if $F \setminus \{C\} \models C$ holds, while otherwise C is called **irredundant** for F. It is C redundant for F if and only if the set $\mathfrak{F}_{\mathrm{var}(F)}(C)$ of falsifying assignments for C is covered by $(\mathfrak{F}_{\mathrm{var}(F)}(C'))_{C' \in F \setminus \{C\}}$, the set of falsifying assignments for the remaining clauses. We are interested here in the question, given a partial assignment φ and a clause $C \in F$ with $\varphi * \{C\} \neq \top$, under what circumstances is the clause $\varphi * C = C \setminus C_\varphi$ redundant for $\varphi * F$? We will see, that this question is closely related to the question, how "much irredundant" C is for F, that is, how much of $\mathfrak{F}_{\mathrm{var}(F)}(C)$ is covered by $(\mathfrak{F}_{\mathrm{var}(F)}(C'))_{C' \in F \setminus \{C\}}$, which can be recast as the question, whether for some $C' \supseteq C$ we have $F \setminus \{C\} \models C'$.

Assume that $\varphi * C$ is redundant for $\varphi * F$, that is, $(\varphi * F) \setminus (\varphi * \{C\}) \models \varphi * C$ holds. Due to $(\varphi * F) \setminus (\varphi * \{C\}) \subseteq \varphi * (F \setminus \{C\})$ it follows $\varphi * (F \setminus \{C\}) \models \varphi * C$, which is equivalent to $F \setminus \{C\} \models C \cup C_\varphi$. Let us call C **φ-redundant** for F if $F \setminus \{C\} \models C \cup C_\varphi$ holds, and otherwise **φ-irredundant**. In other words, C is φ-redundant for F iff the part of $\mathfrak{F}_{\mathrm{var}(F)}(C)$ which consists of assignments compatible with φ is covered by $(\mathfrak{F}_{\mathrm{var}(F)}(C'))_{C' \in F \setminus \{C\}}$.

If C is φ-irredundant for F, then $\varphi * C$ is irredundant for $\varphi * F$, but the reverse direction is not true in general due to the fact, that there might be other clauses $C' \in F$ with $\varphi * C' = \varphi * C$. To repair this, let us call clause C **contraction-φ-redundant** for F if

$$F \setminus \{C' \in F : \varphi * \{C'\} = \varphi * \{C\}\} \models C \cup C_\varphi,$$

while otherwise we call C **contraction-φ-irredundant** for F. We summarise (and extend) the foregoing discussion in Lemma 5.1, whose proof should be obvious by now.

Lemma 5.1 *Consider a generalised clause-set $F \in \mathcal{CLS}$, a clause $C \in F$ and a partial assignment $\varphi \in \mathcal{PASS}$ such that $\varphi * \{C\} \neq \top$.*

1. *$\varphi * C$ is (ir)redundant for $\varphi * F$ if and only if C is contraction-φ-(ir)redundant for F.*
2. *(a) If C is φ-irredundant for F, then C is contraction-φ-irredundant for F.*

(b) *If there is no clause* $C' \in F \setminus \{C\}$ *with* $\varphi * \{C'\} = \varphi * \{C\}$ *(that is,* C *is "contraction-free" in* F *w.r.t.* φ*), then also the reverse direction holds, that is, if* C *is contraction-*φ*-irredundant for* F *then* C *is* φ*-irredundant for* F*. It is* C *contraction-free in* F *w.r.t.* φ *in the following cases:*

(i) $n(\varphi) = 0$ *(i.e.,* φ *is the empty partial assignment);*

(ii) $n(\varphi) = 1$ *and* F *is subsumption-free;*

(iii) C *clashes with every* $C' \in F \setminus \{C\}$*.*

Corollary 5.2 *Consider a generalised clause-set* $F \in \mathcal{CLS}$ *which is subsumption-free, a clause* $C \in F$ *and a variable* $v \in \mathcal{VA}$ *together with* $\varepsilon \in D_v$ *such that for* $\varepsilon' \in D_v \setminus \{\varepsilon\}$ *we have* $(v, \varepsilon') \notin C$*. Then* $\langle v \to \varepsilon\rangle * C = C \setminus \{(v, \varepsilon)\}$ *is irredundant for* $\langle v \to \varepsilon\rangle * F$ *if and only if* C *is* $\langle v \to \varepsilon\rangle$*-irredundant for* F*, that is, iff* $F \setminus \{C\} \not\models C \cup \{(v, \varepsilon)\}$*.*

A (generalised) clause-set $F \in \mathcal{CLS}$ is called **irredundant** if all $C \in F$ are irredundant for F, otherwise F is called **redundant**. A clause-set F is minimally unsatisfiable if and only if F is unsatisfiable and irredundant. Obviously irredundant clause-sets are subsumption-free, and from Corollary 5.2 we get immediately:

Corollary 5.3 *Consider an irredundant generalised clause-set* $F \in \mathcal{CLS}$*, a clause* $C \in F$ *and a variable* $v \in \mathcal{VA}$ *together with* $\varepsilon \in D_v$*.*

1. *If there exists* $\varepsilon' \in D_v \setminus \{\varepsilon\}$ *with* $(v, \varepsilon') \in C$*, then clause* C *vanishes when applying* $\langle v \to \varepsilon\rangle$ *to* F *(and in that sense it becomes redundant in* $\langle v \to \varepsilon\rangle$*). So assume* $\mathrm{val}_v(\{C\}) \subseteq \{\varepsilon\}$ *in the sequel.*
2. *If* $(v, \varepsilon) \in C$*, then* $\langle v \to \varepsilon\rangle * C = C \setminus \{(v, \varepsilon)\}$ *is irredundant for* $\langle v \to \varepsilon\rangle * F$*.*
3. *If* $(v, \varepsilon) \notin C$*, then* C *is irredundant for* $\langle v \to \varepsilon\rangle * F$ *if and only if* C *is* $\langle v \to \varepsilon\rangle$*-irredundant for* F*, i.e., iff* $F \setminus \{C\} \not\models C \cup \{(v, \varepsilon)\}$*.*

5.2 Hitting and multihitting clause-sets

The next lemma answers the question which clauses remain irredundant for a clause-set F under *all* applications of partial assignments; this strongest form of irredundancy of C for F turns out to be equivalent to the condition, that the set of falsifying assignments for C is not covered at all by $(\mathfrak{F}_{\mathrm{var}(F)}(C'))_{C' \in F \setminus \{C\}}$. A simple but important observation useful here is, that for two clauses C, C' and $\mathrm{var}(C) \cup \mathrm{var}(C') \subseteq V$ we have $\mathfrak{F}_V(C) \cap \mathfrak{F}_V(C') = \emptyset$ iff C and C' clash.

Lemma 5.4 *Consider a generalised clause-set* $F \in \mathcal{CLS}$ *and a clause* $C \in \mathcal{CL}$*. Then the following assertions are equivalent:*

(i) C *is* φ*-irredundant for all* $\varphi \in \mathcal{PASS}$*.*

(ii) C *is contraction-*φ*-irredundant for all* $\varphi \in \mathcal{PASS}$*.*

(iii) $\mathfrak{F}_{\mathrm{var}(F)}(C) \cap \bigcup_{C' \in F \setminus \{C\}} \mathfrak{F}_{\mathrm{var}(F)}(C') = \emptyset$*.*

(iv) C *clashes with every* $C' \in F \setminus \{C\}$*, i.e., clause* C *is connected in the conflict-graph* $\mathrm{cg}(F)$ *to every other vertex.*

Proof: By the above remark we see that (iii) and (iv) are equivalent. By definition (iii) is equivalent to (i), while by Lemma 5.1, part 2 it is (i) equivalent to (ii). ∎

Corollary 5.5 *A generalised clause-set $F \in \mathcal{CLS}$ is a hitting clause-set if and only if for all $\varphi \in \mathcal{PASS}$ it is $\varphi * F$ irredundant.*

Generalising Theorem 32 in [27]:

Corollary 5.6 *A generalised clause-set $F \in \mathcal{CLS}$ is an unsatisfiable hitting clause-set if and only if for every partial assignment $\varphi \in \mathcal{PASS}$ it is $\varphi * F$ minimally unsatisfiable.*

Hitting clause-sets are irredundant; the more general class of *multihitting clause-sets* (clause-sets with multipartite conflict graph) contains redundant clause-sets, but all redundancies can be removed efficiently (and canonically), as the following lemma shows. We use the notion of an **irredundant core** of a clause-set $F \in \mathcal{CLS}$ which is an irredundant $F' \subseteq F$ such that F' is equivalent to F. An irredundant core of an unsatisfiable clause-set is called a **minimally unsatisfiable core**.

Lemma 5.7 *Consider a generalised clause-set $F \in \mathcal{CLS}$ which is multihitting. Let $\mathbb{F}$ be the multipartition of F, and $V := \mathrm{var}(F)$.*

1. *For $F_1, F_2 \in \mathbb{F}$, $F_1 \neq F_2$ we have $\mathfrak{F}_V(F_1) \cap \mathfrak{F}_V(F_2) = \emptyset$.*
2. *If for $F' \subseteq F$ and $C \in F \setminus F'$ we have $F' \models C$, then there must be some $C' \in F'$ with $C' \subset C$.*
3. *F has exactly one irredundant core, which is obtained from F by subsumption-elimination. Thus if F is unsatisfiable, then F has exactly one minimally unsatisfiable core, which is obtained from F by subsumption-elimination.*
4. *A hitting clause-set F is unsatisfiable iff $\sum_{C \in F} |\mathfrak{F}_V(\{C\})| = |\mathcal{PASS}(V)|$.*

Proof: Part 1 follows by definition. In Part 2 it is $\mathfrak{F}_V(\{C\})$ covered by $\mathfrak{F}_V(F')$, and thus by Part 1 in fact $\mathfrak{F}_V(\{C\})$ is covered by $\mathfrak{F}_V(F' \cap F_C)$, where $F_C \in \mathbb{F}$ with $C \in F_C$; i.e., $F_C \cap F' \models \{C\}$. By the strong completeness of resolution and the fact that within F_C no clashes exist, it follows that there must be $C' \in F' \cap F_C$ with $C' \subset C$. Part 3 follows immediately from Part 2. Finally Part 4 follows immediately from Part 1. ∎

Using $|\mathfrak{F}_{\mathrm{var}(F)}(C)| = \prod_{v \in \mathrm{var}(F) \setminus \mathrm{var}(C)} |D_v|$ for $C \in F$ it follows that satisfiability for generalised hitting clause-sets is decidable in polynomial time (generalising the well-known special case for boolean clause-sets). For boolean *bihitting* clause-sets (where the conflict graph is bipartite) in [15] it was shown, that satisfiability decision can be done in quasi-polynomial time (where "quasi-polynomial" means a polynomial upper bound with the exponent of logarithmic order in the size of the input); this can immediately be generalised:

Lemma 5.8 *Satisfiability for generalised clause-sets which are bihitting is decidable in quasi-polynomial time.*

Proof: Variables with a domain size greater than two appearing in a bihitting clause-set must be pure variables, since if a generalised clause-set contains a variable

of domain size k, then the conflict graph contains the complete graph K_k (which is not bipartite). ∎

It seems to be a very interesting question, to what degree (generalised) multi-hitting clause-sets have efficient satisfiability decision (see [28] for more information in the boolean case).

5.3 Saturated minimally unsatisfiable clause-sets

A clause-set $F \in \mathcal{CLS}$ is called **saturated minimally unsatisfiable**, if F is unsatisfiable, but for any clause $C \in F$ replacing C in F by $C \cup \{x\}$ for any literal x with $\mathrm{var}(x) \notin \mathrm{var}(C)$ and $|D_{\mathrm{var}(x)}| \geq 2$ yields a satisfiable clause-set. Saturated minimally unsatisfiable clause-sets are minimally unsatisfiable (consider x such that $\mathrm{var}(x) \notin \mathrm{var}(F)$), and actually a clause-set F is saturated minimally unsatisfiable iff it is minimally unsatisfiable and addition of a literal x with $\mathrm{var}(x) \in \mathrm{var}(F)$ to any clause C with $\mathrm{var}(x) \notin \mathrm{var}(C)$ yields a satisfiable clause-sets. The set of all saturated minimally unsatisfiable clause-sets is called $\boldsymbol{\mathcal{SMUSAT}}$. By Lemma 5.7, part 4 we see that unsatisfiable hitting clause-sets are in $\mathcal{SMUSAT}$.

Lemma 5.9 *Every minimally unsatisfiable clause-set $F \in \mathcal{MUSAT}$ can be **saturated**, that is there exists $F^* \in \mathcal{SMUSAT}$ with $\mathrm{var}(F^*) = \mathrm{var}(F)$ and a bijection $\pi : F \to F^*$ such that for all $C \in F$ we have $C \subseteq \pi(C)$.*

Proof: The observation needed here is, that if for a minimally unsatisfiable clause-set F we replace some clause $C \in F$ by a clause $C' \supset C$, obtaining $F' := (F \setminus \{C\}) \cup \{C'\}$, then F' is minimally unsatisfiable if F' is unsatisfiable (the only possibly redundant clause in F' is C', and if C' is redundant in F', then F' is satisfiable, since $F' \setminus \{C'\} = F \setminus \{C\} \in \mathcal{SAT}$). So we can add literals x with $\mathrm{var}(x) \in \mathrm{var}(F)$ to clauses such that we maintain (minimally) unsatisfiability, and finally we will end up with a saturated F^*. ∎

For boolean clause-sets the characterisation of $\mathcal{SMUSAT}$ from Lemma C.1 in [22] is fundamental: A minimally unsatisfiable boolean clause-set F is saturated if and only if for every variable $v \in \mathrm{var}(F)$ and each $\varepsilon \in D_v = \{0, 1\}$ it is $\langle v \to \varepsilon \rangle * F$ minimally unsatisfiable. Together with saturation this characterisation provides a powerful method for proving properties of minimally unsatisfiable clause-sets via induction on the number of variables. For generalised clause-sets saturatedness is weaker, and the above condition is only sufficient for being minimally unsatisfiable, but is no longer necessary. The following lemma develops these fundamental facts, using the following notion: We say that *addition of literal x renders clause-set F satisfiable* iff for all clauses $C \in F$ with $\mathrm{var}(x) \notin \mathrm{var}(C)$ the clause-set $(F \setminus \{C\}) \cup \{C \cup \{x\}\}$ is satisfiable (thus a clause-set F is saturated minimally unsatisfiable iff F is unsatisfiable and addition of any literal renders F satisfiable).

Lemma 5.10 *Consider a generalised clause-set $F \in \mathcal{MUSAT}$ and a literal $(v, \varepsilon) \in \mathcal{LIT}$.*

1. *If $\langle v \to \varepsilon \rangle * F \in \mathcal{MUSAT}$, then for all $\varepsilon' \in D_v \setminus \{\varepsilon\}$ addition of literal (v, ε') renders F satisfiable.*
2. *If v is boolean, and for $\varepsilon' \in D_v \setminus \{\varepsilon\}$ addition of literal (v, ε') renders F satisfiable, then $\langle v \to \varepsilon \rangle * F \in \mathcal{MUSAT}$.*

Proof: For Part 1 assume that there is $C \in F$, $v \notin \mathrm{var}(C)$ and $\varepsilon' \in D_v \setminus \{\varepsilon\}$ such that $F' := (F \setminus \{C\}) \cup \{C \cup \{(v, \varepsilon')\}\}$ is unsatisfiable. Then $\langle v \to \varepsilon\rangle * F' \in \mathcal{USAT}$ with $\langle v \to \varepsilon\rangle * F' = (\langle v \to \varepsilon\rangle * F) \setminus \{C\}$, and thus C would be redundant in $\langle v \to \varepsilon\rangle * F$.

For Part 1 assume that $\langle v \to \varepsilon\rangle * F$ is not minimally unsatisfiable; by Corollary 5.3 thus there is a clause $C \in F$, $v \notin \mathrm{var}(C)$ such that $F \setminus \{C\} \models C \cup \{(v, \varepsilon)\}$. It follows that for $F' := (F \setminus \{C\}) \cup \{C \cup \{(v, \varepsilon')\}\}$ we have $F' \models C$ (using one resolution step), and thus F' would be unsatisfiable. ∎

Corollary 5.11 *If for the generalised clause-set $F \in \mathcal{CLS}$ for every partial assignment $\varphi \in \mathcal{PASS}$ with $n(\varphi) \leq 1$ we have $\varphi * F \in \mathcal{MUSAT}$, then $F \in \mathcal{SMUSAT}$. If F is boolean, then also the reverse direction holds, that is, $F \in \mathcal{SMUSAT}$ if and only if for every partial assignment $\varphi \in \mathcal{PASS}$ with $n(\varphi) \leq 1$ we have $\varphi * F \in \mathcal{MUSAT}$.*

An example to show that the condition for $F \in \mathcal{SMUSAT}$ in Lemma 5.11 is not necessary for generalised clause-sets is as follows: Consider variables a, b with $D_a = D_b = \{0, 1, 2\}$, and let

$$F := \big\{ \{(a,0),(b,0)\}, \{(a,1),(b,0)\}, \{(a,0),(b,1)\}, \{(a,1),(b,1)\}, \{(a,2)\}, \{(b,2)\} \big\}.$$

It is $F \in \mathcal{SMUSAT}$ (as can be verified directly), while $\langle a \to 2\rangle * F = \{\bot, \{(b,2)\}\} \notin \mathcal{MUSAT}$ (as well as $\langle b \to 2\rangle * F = \{\bot, \{(a,2)\}\} \notin \mathcal{MUSAT}$).

An important application of the process of saturation for boolean clause-sets is given by Lemma C.2 in [22], proving that for every $F \in \mathcal{MUSAT}$, $F \neq \{\bot\}$ there is a variable $v \in \mathrm{var}(F)$ such that for $\varepsilon \in D_v = \{0, 1\}$ we have $\#_{(v,\varepsilon)}(F) \leq \delta(F)$. The proof is based on the characterisation of saturated minimally unsatisfiable boolean clause-sets in Corollary 5.11 and uses $\delta(F) \geq 1$ for $F' \in \mathcal{MUSAT}$, where F' is obtained from F by applying suitable partial assignments φ with $n(\varphi) = 1$. It is not clear to me how to obtain a generalisation for generalised clause-sets (the problem is that saturation is not that powerful anymore).

6 Linear autarkies

The notion of a "linear autarky", introduced for boolean clause-sets in [23] and further investigated in [38, 25], has the following natural generalisation to generalised clause-sets: A partial assignment $\varphi \in \mathcal{PASS}$ is called a **linear autarky** for $F \in \mathcal{CLS}$ if there is a weight function $w : \mathrm{var}(F) \to \mathbb{Q}_{>0}$, assigning to each variable in F a positive rational number, such that for all clauses $C \in F$ we have

$$\sum_{x \in C, \varphi(x) = 1} w(\mathrm{var}(x)) \geq \sum_{x \in C, \varphi(x) = 0} w(\mathrm{var}(x)), \tag{2}$$

that is, if for every clause the sum of the weights of the underlying variables of literals satisfied by φ is not smaller than that sum over the literals falsified by φ.[6)] Obviously, every linear autarky for F is an autarky for F. Choosing a constant weight function we see, that every satisfiable generalised clause-set with clauses of size at most two is satisfiable by a linear autarky, and thus, since for example graph

[6)] Literals with underlying variables not in the domain of φ are not taken into account.

colouring can be expressed with such clause-sets (see the following section for a general framework for expressing colouring problems via generalised clause-sets), we see that deciding the existence of a linear autarky for generalised clause-sets is NP-hard. In this article we do not further investigate the notion of linear autarkies for generalised clause-sets (see [24] for some hints on further developments), but we take a closer look at "balanced linear autarkies" for boolean clause-sets.

As introduced in [25] (Section 6), a balanced linear autarky for a generalised clause-set F is a linear autarky for F where in (2) always equally holds. As remarked at the end of Section 6 in [25], a boolean clause-set F has no non-trivial balanced linear autarky (i.e., F is **lean w.r.t. balanced linear autarkies**) iff the columns of the clause-variable matrix $M(F)$ of F are linearly independent (where the clause-variable matrix $M(F)$ of a multi-clause-set F is a $c(F)\times n(F)$ matrix over $\{-1,0,+1\}$, containing the natural encoding of the clauses in the rows); it follows that if F is balanced-linearly lean, then $\delta(F)\geq 0$. We will later see, that this is the heart of the statement proven in [35], that a minimally non-2-colourable hypergraph without isolated vertices has as least as many edges as vertices. In Theorem 2 of [1] this statement was strengthened to the statement, that in a minimally non-2-colourable hypergraph G without isolated vertices there exists a matching in the bipartite graph corresponding to G (consisting of the vertices and the hyperedges of G as new vertices, exactly in the same way as for $B(F)$) covering all vertices. We will see that the following lemma generalises this strengthening.

Lemma 6.1 *Consider a boolean clause-set $F\in\mathcal{CLS}$ which is lean w.r.t. balanced linear autarkies. Then $\delta^*(F)=\delta(F)$ holds, i.e., there is a matching in $B(F)$ covering all variable nodes. (Since $\delta^*(F)\geq 0$, this statement includes $\delta(F)\geq 0$.)*

Proof: Assume $\delta^*(F)>\delta(F)$, and consider a tight F' for F. By Lemma 4.14, Part 3b there is a matching autarky φ for F with $\varphi * F = F'$. Let $V := \mathrm{var}(\varphi)$. Since $\delta(F')>\delta(F)$, by Lemma 4.14, Part 1 we get $\delta(F[V])<0$. In general a multi-clause-set $G\in\mathcal{MCLS}$ has a non-trivial balanced linear autarky if $M(G)\cdot\vec{x}=0$ has a non-trivial solution, and this is guaranteed if this system of linear equations is under-constrained, that is, if $\delta(G)<0$ holds. Thus $F[V]$ has a non-trivial balanced linear autarky, which yields a balanced linear autarky for F. ∎

For balanced-linearly lean boolean clause-sets F we thus know $\forall\, F'\subseteq F : \delta(F')\leq\delta(F)$, which is weaker than $\forall\, F'\subset F:\delta(F')<\delta(F)$, shown in Lemma 4.15 to be equivalent to F being matching lean. But the notion of balanced linear autarkies is incomparable to the notion of matching autarkies, as the following examples show.

1. Consider three (different) boolean variables $x,y,z\in\mathcal{VA}$ and a clause-set F containing 6 (different) clauses C with $\mathrm{var}(C)=\{x,y,z\}$ (and nothing else). Consider (different) boolean variables $a,b\in\mathcal{VA}\setminus\{x,y,z\}$, and add the literals a,b to all 6 clauses in F. The partial assignment $\langle a\to 1, b\to 0\rangle$ is a balanced linear autarky for F (choose some constant weighting) satisfying F, while F is matching lean, since $\delta(F)=6-5=1$, while for $\top\neq F'\subset F$ we have $n(F')=5$ and $c(F')\leq 5$, and thus $\delta(F')\leq 0$.

2. The most trivial example for a matching satisfiable boolean clause-set which is balanced-linearly lean is $\{\{a\}\}$, while an example not containing unit clauses is $\{\{a,b\},\{a,\overline{b}\}\}$.

Besides Lemma 6.1, the importance of balanced linear autarkies for us lies in the following lemma (which follows directly from the definitions), which allows us to conclude from a boolean clause-set $F \cup \{\overline{C} : C \in F\}$ being linearly lean, that F is balanced linearly lean. For a boolean literal (x, ε) we use $\overline{(v, \varepsilon)} := (v, 1 - \varepsilon)$, while for a boolean clause C we use $\overline{C} := \{\overline{x} : x \in C\}$, and finally for a boolean clause-set F we set $\overline{F} := \{\overline{C} : C \in F\}$.

Lemma 6.2 *A boolean clause-set F is balanced linearly lean if and only if $F \cup \overline{F}$ is linearly lean.*

7 Applications to hypergraph colouring

Consider a hypergraph G without isolated vertices and $k \in \mathbb{N}$. We associate the generalised clause-set $\boldsymbol{F_{[k]}(G)} \in \mathcal{CLS}$ to G, where we set

$$F_{[k]}(G) := \big\{ \{(v, i) : v \in E\} : E \in E(G), i \in \{1, \ldots, k\} \big\},$$

that is, the variables of $F_{[k]}(G)$ are the vertices of G, while the clauses of $F_{[k]}(G)$ are the hyperedges of G in k versions corresponding to the k choices for the colour. We have $\mathrm{var}(F_{[k]}(G)) = V(G)$, $D_v = \{1, \ldots, k\}$ for all $v \in \mathrm{var}(F_{[k]}(G))$, $c(F_{[k]}(G)) = k \cdot |E(G)|$, and $\ell(F_{[k]}(G)) = k \cdot \sum_{E \in E(G)} |E|$. By definition it is G k-colourable if and only if $F_{[k]}(G)$ is satisfiable, and moreover the set of k-colourings of G is identical to $\mathfrak{S}_{\mathrm{var}(F_{[k]}(G))}(F_{[k]}(G))$. The variable hypergraph of $F_{[k]}(G)$ is G. Clause-sets expressing hypergraph colouring problems in this canonical way are up to giving different names to the variable values exactly "diagonal clause-sets" defined as follows.[7)]

For practical applications, it is useful to break the symmetry between the values $1, \ldots, k$ of the colouring as follows: Choose (different) variables $v_1, \ldots, v_{k-1} \in V(G)$ and use the restricted domains $D_{v_i} = \{1, \ldots, i\}$ for $i \in \{1, \ldots, k-1\}$; the clause-set obtained in this way is satisfiability-equivalent to $F_{[k]}(G)$, but the search space has been reduced. In practice, the savings obtained by this simple method are quite considerable, but for theoretical reasoning we gain only unnecessary complications, and thus in this paper we do not use this elementary symmetry breaking.

A generalised clause-set $F \in \mathcal{CLS}$ is called **diagonal**, if there is a universal domain $D(F)$ such that for all $v \in \mathrm{var}(F)$ we have $D_v = D$, for all clauses $C \in F \setminus \{\bot\}$ there exists $\varepsilon_C \in D(F)$ such that for all $x \in C$ we have $\mathrm{val}(x) = \varepsilon$, and for the sub-clause-sets $\boldsymbol{F^{[\varepsilon]}} := \{C \in F : \varepsilon_C = \varepsilon\}$ we have, that the underlying variable-hypergraph is the same for all $\varepsilon \in D(F)$. If $\mathrm{var}(F) \neq \emptyset$, then $D(F)$ is uniquely determined, while otherwise we set $D(F) := \emptyset$. We denote the set of all diagonal clause-sets by $\boldsymbol{\mathcal{DCLS}}$. If for $F \in \mathcal{DCLS}$ we have $D(F) = \emptyset$, then $F = \top$ or $F = \{\bot\}$. Note that for all $F \in \mathcal{DCLS}$ and $\varepsilon \in D(F)$ by definition we have $\bot \notin F^{[\varepsilon]}$; a clause-set $F \in \mathcal{CLS}$ is diagonal iff $F \setminus \{\bot\}$ is iff $F \cup \{\bot\}$ is. If for $F \in \mathcal{DCLS}$ we have $|D(F)| \leq 1$, then F is lean. For a hypergraph G without isolated vertices and for $k \geq 1$ it is $F_{[k]}(G)$ a diagonal clause-set with $D(F_{[k]}(G)) = \{1, \ldots, k\}$.

A diagonal clause-set F is satisfiable iff the variable-hypergraph of F is $|D(F)|$-colourable. Thus diagonal clause-sets are nothing else than the canonical translations of hypergraph colouring problems into (generalised) clause-sets. Boolean diagonal clause-sets are special cases of "PN-clause-sets" as introduced in [15] (boolean

[7)] There is the further tiny difference, that $F_{[k]}(G)$ is only reasonably definable as a clause-set in case of $k \geq 1$, while starting with clause-sets we do not have such a restriction.

clause-sets, where each clause either is positive or negative), which are the heart of the class of bipartite clause-sets (clause-sets, where the conflict graph is bipartite).

Lemma 7.1 *A diagonal clause-set $F \in \mathcal{DCLS}$ is minimally unsatisfiable if and only if the variable hypergraph of F is minimally non-$|D(F)|$-colourable. Put into the hypergraph perspective: A hypergraph G without isolated vertices is minimally non-k-colourable for $k \in \mathbb{N}$ if and only if $F_{[k]}(G)$ is minimally unsatisfiable.*

Proof: Consider a diagonal clause-set F, let $k := |D(F)| \in \mathbb{N}_0$, and let G denote the variable hypergraph of F. Obviously, if F is minimally unsatisfiable, then G is minimally non-k-colourable. Now assume that G is minimally non-k-colourable, but F is not minimally unsatisfiable. Thus there exists $C \in F$ such that $F \setminus \{C\}$ is unsatisfiable. If $C = \bot$, then $G = (\emptyset, \{\emptyset\})$, and thus $F = \{\bot\}$ would be minimally unsatisfiable; so let $\varepsilon \in D(F)$ be the common value of the literals in C. The hypergraph G minus the edge $\mathrm{var}(C)$ has a $D(F)$-colouring f. It must f colour all vertices in hyperedge $\mathrm{var}(C)$ with the same colour $\varepsilon' \in D(F)$. Consider a bijection $\pi \in S_{D(F)}$ with $\pi(\varepsilon') = \varepsilon$, and set $f' := \pi \circ f$. Now f' (which still is a $D(F)$-colouring for G) is a satisfying assignment for $F \setminus \{C\}$ contradicting the assumption. ∎

For the proof of the main lemma of this section we need the following observation on balanced linear autarkies, which follows directly from Lemma 6.2.

Lemma 7.2 *Consider a diagonal generalised clause-set $F \in \mathcal{DCLS}$ with $0, 1 \in D(F)$. Let F' result from $F^{[1]}$ by renaming all variables, so that they become boolean variables (i.e., their domain is restricted to $\{0, 1\}$). If now F' has a non-trivial balanced linear autarky, then this autarky, when undoing the renaming, yields a non-trivial linear autarky for F. Thus, if F is linearly lean, then F' is balanced-linearly lean.*

Lemma 7.3 *For every diagonal lean generalised clause-set F with $|D(F)| \geq 2$ we have $\delta(F) \geq n(F)$.*

Proof: Consider F' obtained from F as in Lemma 7.2. Since F is balanced-linearly lean, so is F', and by Lemma 6.1 we obtain $\delta(F') \geq 0$. Now $c(F) = |D(F)| \cdot c(F')$, while $\delta(F') = c(F') - n(F)$, from which $c(F) \geq |D(F)| \cdot n(F)$ follows; finally by $\delta(F) = c(F) - (|D(F)| - 1) \cdot n(F)$ the assertion follows. ∎

As an application we can generalise a result from [35] (where only the case $k = 2$ was considered; see also Theorem 5.12 in [9] and Lemma 4.7 in [17]).

Corollary 7.4 *Consider a hypergraph G without isolated vertices, which is minimally non-k-colourable for some $k \geq 2$. Then $|E(G)| \geq |V(G)|$.*

Proof: By Lemma 7.1 the clause-set $F_{[k]}(G)$ is minimally unsatisfiable, thus by Lemma 7.3 we have $\delta(F) \geq n(F)$, where $\delta(F) = k \cdot |E(G)| - (k-1) \cdot |V(G)|$ and $n(F) = |V(G)|$. ∎

Corollary 7.4 shows the power of the approach of translating hypergraph colouring problems into satisfiability problems for generalised clause-sets: If a hypergraph G is critical k-colourable, then G is not critical k-colourable for any $k' < k$, and thus there is no direct way to reduce the assertion for $k > 2$ to the known case $k = 2$. However if $F_{[k]}(G)$ is minimally unsatisfiable, then $F_{[k]}(G)$ is lean, thus also $F_{k'}(G)$ is lean for all $k' < k$. And the lower bound on the deficiency does not need a minimality condition, but only a certain leanness condition.

Let us close this article by some final remarks on the relations between generalised satisfiability problems and hypergraph colouring theory.

1. Translating the characterisation in [35] of "self-blocking square condensers" (Proposition 7) into our language, we arrive at an interesting characterisation of all *minimally unsatisfiable diagonal boolean bihitting clause-sets* F with minimal deficiency $\delta(F) = n(F)$:

   (a) If F contains a unit clause, then F must be isomorphic to $\{\{v\}, \{\overline{v}\}\}$.

   (b) If F contains a binary clause, then F is isomorphic to $F_{[2]}(C)$, where C is either the cycle $C^3 = (\{1,2,3\}, \{\{1,2\}, \{2,3\}, \{1,3\}\})$ with three edges, or is obtained by repeated "2-extensions" from C^3. A 2-extension of a simple hypergraph G arising from C^3 chooses an edge $\{x_1, x_2\} \in E(G)$ and some new vertex z, adds the edge $\{x_1, z\}$, and adds the vertex z to all edges containing x_2 except of $\{x_1, x_2\}$.[8)]

   (c) Otherwise F is isomorphic to $F_{[2]}(S)$, where S is the Fano plane (also known as the projective plane over $\mathbb{Z}_2$, or as the Steiner system $\mathfrak{S}(2,3,7)$; we have $|V(S)| = |E(S)| = 7$, every hyperedge of S has length 3, and for every pair of distinct vertices there is exactly one hyperedge containing them[9)]).

   It seems to me a rewarding enterprise to revise this proof using the framework of this article, and to investigate how far such a characterisation can be extended.

2. Consider a colourable hypergraph G. It is $F_{[1]}(G)$ lean, while $F_{[\chi(G)]}(F)$ is satisfiable, and thus not lean in case of $V(G) \neq \emptyset$. And if $F_{[k]}(G)$ is lean, then so is $F_{[k']}(G)$ for $k' \leq k$. So I propose to study the **autarky number $\chi_{\mathrm{a}}(G) \leq \chi(G)$**, the maximal k so that $F_{[k]}(G)$ is lean. I imagine, that $\chi_{\mathrm{a}}(G)$ has many interesting properties, and via the use of autarky systems the autarky number of hypergraphs can also be generalised (considering for example the *matching autarky number*). Corollary 7.4 then for example can be generalised as the assertion, that a hypergraph without isolated vertices and with $\chi_{\mathrm{a}}(G) \geq 2$ has at least as many edges as vertices; by introducing an appropriate normal autarky system capturing "boolean balanced linear autarkies", and using the autarky number associated with that autarky system we then arrive at what looks as a very natural "most general condition" for having at least as many edges as vertices.

8 Conclusion and open problems

The first purpose of this article was to set the stage for the study of generalised clause-sets as sets of "no-goods", where literals are given by one "forbidden value": We defined and summarised the basic properties of syntax, semantics, resolution calculus and autarky systems. Then we considered the generalisation of the notion of deficiency for these generalised clause-sets, and we studied the basic autarky systems related to this notion, matching autarkies and balanced linear autarkies.

[8)] Examples isomorphic to 2-extensions of C^3 are $(\{0,\ldots,n\},\ \{1,\ldots,n\}, \{0,1\}, \ldots, \{0,n\}\ )$ for $n \geq 2$, and $(\{1,\ldots,6\},\ \{1,2\}, \{1,3\}, \{1,4\}, \{1,5,6\}, \{2,3,4,5\}, \{2,3,4,6\}\ )$.

[9)] $S \cong (\{1,\ldots,7\},\ \{1,2,3\}, \{3,4,5\}, \{1,5,6\}, \{1,4,7\}, \{2,5,7\}, \{3,6,7\}, \{2,4,6\}\ )$

For autarky systems both the application of autarkies as reductions and the properties of autarky-free, i.e., lean clause-sets are of interest. Lean clause-sets are a generalisation of minimally unsatisfiable clause-sets, for which we considered the basic problem, when the property of being minimally unsatisfiable is preserved under application of partial assignments. Finally, we gave a canonical translation of hypergraph colouring problems into generalised satisfiability problems, and showed how to generalise a well-known result of Seymour.

The embedding of the hypergraph colouring problem into the richer space of generalised satisfiability problems enables operations which are external to the hypergraph environment, and can be expressed at this level only in clumsy ways. (However, it seems essential that this richer space is still "close enough" to the original domain.) One of these operations is the operation of autarkies. Autarky-freeness ("leanness") yields a natural and "smooth" extension of the notion of "minimality".

At several points we experienced that generalised clause-sets have a more complicated structure than boolean problems. In Lemma 5.10 we have seen, that the saturation process for minimally unsatisfiable generalised clause-sets is weaker than in the boolean case. Missing this essential tool, generalising the well-known structure of $\mathcal{MUSAT}_{\delta=1}$ from the boolean case (see [1, 7, 22, 27]) is an open problem, and so is the generalisation of the structure of boolean $\mathcal{MUSAT}_{\delta=2}$ as studied in [4] (at present I do not even know whether for these small deficiencies there are essentially more complicated examples, or whether the given characterisation can be carried over in the natural way (using stronger proof methods)).

A second case where generalised clause-sets resisted our attempts is given by the open problem, whether satisfiability decision for $\mathcal{MUSAT}_{\delta=k}$ is fixed-parameter-tractable in k (as shown in the boolean case in [36]). Similar to the problems with minimally unsatisfiable clause-sets, the problem is how to control the deficiency when applying partial assignments; in the boolean case it could be exploited that one variable contributes only 1 to the deficiency, but now it contributes $|D_v| - 1$, and this seems to be a non-trivial obstacle (already in the case where $|D_v| = 3$, which arises for example when translating 3-colouring problems into generalised satisfiability problems) .

For the future, I expect that the study of the characterisation of "self-blocking square condensers" from [35], that is, of minimally unsatisfiable diagonal boolean bihitting clause-sets as discussed in the first final remark in Section 7, yields new insights into the structure of generalised clause-sets as well as into the interplay between combinatorics and generalised satisfiability problems. And I hope that the autarky number of hypergraphs as discussed in the second final remark in Section 7 can become a powerful tool for investigations in hypergraph theory.

References

[1] Ron Aharoni and Nathan Linial. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *Journal of Combinatorial Theory*, A 43:196–204, 1986.

[2] Andrew B. Baker. *Intelligent Backtracking on Constraint Satisfaction Problems: Experimental and Theoretical Results*. PhD thesis, University of Oregon, March 1995. The ps-file can be down loaded using the URL `http://www.cirl.uoregon.edu/baker/thesis.ps.gz`.

[3] 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.

[4] Hans Kleine Büning. On subclasses of minimal unsatisfiable formulas. *Discrete Applied Mathematics*, 107:83–98, 2000.

[5] Hans Kleine Büning and Xishun Zhao. The complexity of some subclasses of minimal unsatisfiable formulas. Von Sam Buss zur Begutachtung erhalten, December 2000.

[6] Nadia Creignou and Hervé Daudé. Generalized satisfiability problems: minimal elements and phase transitions. *Theoretical Computer Science*, 302:417–430, 2003.

[7] 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*, 23:229–245, 1998.

[8] Michael R. Dransfield, Victor W. Marek, and Miroslaw Truszczyński. Satisfiability and computing van der Waerden numbers. In Giunchiglia and Tacchella [16], pages 1–13. ISBN 3-540-20851-8.

[9] Pierre Duchet. Hypergraphs. In Graham et al. [18], chapter 7, pages 381–432. ISBN 0-444-82346-8; QA164.H33 1995.

[10] 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.

[11] 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.

[12] 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.

[13] John Franco and Allen Van Gelder. A perspective on certain polynomial-time solvable classes of satisfiability. *Discrete Applied Mathematics*, 125:177–214, 2003.

[14] Alan M. Frisch and Timothy J. Peugniez. Solving non-boolean satisfiability problems with stochastic local search. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 282–288, 2001.

[15] Nicola Galesi and Oliver Kullmann. Polynomial time SAT decision, hypergraph transversals and the hermitian rank. In *The Seventh International Conference on Theory and Applications of Satisfiability Testing*, Lecture Notes in Computer Science. Springer, 2004. To appear.

[16] Enrico Giunchiglia and Armando Tacchella, editors. *Theory and Applications of Satisfiability Testing 2003*, volume 2919 of *Lecture Notes in Computer Science*, Berlin, 2004. Springer. ISBN 3-540-20851-8.

[17] Chris D. Godsil. Tools from linear algebra. In Ronald L. Graham, Martin Grötschel, and László Lovász, editors, *Handbook of Combinatorics, Volume 2*, chapter 31, pages 1705–1748. North-Holland, Amsterdam, 1995. ISBN 0-444-82351-4; QA164.H33 1995.

[18] Ronald L. Graham, Martin Grötschel, and László Lovász, editors. *Handbook of Combinatorics, Volume 1*. North-Holland, Amsterdam, 1995. ISBN 0-444-82346-8; QA164.H33 1995.

[19] Shlomo Hoory and Stefan Szeider. Families of unsatisfiable k-CNF formulas with few occurrences per variable. Available from arXiv ???, November 2004.

[20] Oliver Kullmann. New methods for 3-SAT decision and worst-case analysis. *Theoretical Computer Science*, 223(1-2):1–72, July 1999.

[21] Oliver Kullmann. Restricted versions of extended resolution. *The Bulletin of Symbolic Logic*, 5(1):119, 1999. Abstracts of contributed papers of the Logic Colloquium' 98, Prague, Czech Republic, August 9 - 15, 1998.

[22] Oliver Kullmann. An application of matroid theory to the SAT problem. In *Fifteenth Annual IEEE Conference on Computational Complexity*, pages 116–124. IEEE Computer Society, July 2000.

[23] Oliver Kullmann. Investigations on autark assignments. *Discrete Applied Mathematics*, 107:99–137, 2000.

[24] Oliver Kullmann. On the use of autarkies for satisfiability decision. In Henry Kautz and Bart Selman, editors, *LICS 2001 Workshop on Theory and Applications of Satisfiability Testing (SAT 2001)*, volume 9 of *Electronic Notes in Discrete Mathematics (ENDM)*. Elsevier Science, June 2001.

[25] Oliver Kullmann. Lean clause-sets: Generalizations of minimally unsatisfiable clause-sets. *Discrete Applied Mathematics*, 130:209–249, 2003.

[26] Oliver Kullmann. On the conflict matrix of clause-sets. Technical Report CSR 7-2003, University of Wales Swansea, Computer Science Report Series, 2003.

[27] Oliver Kullmann. The combinatorics of conflicts between clauses. In Giunchiglia and Tacchella [16], pages 426–440. ISBN 3-540-20851-8.

[28] Oliver Kullmann. Conflict matrices and multi-hitting clause-sets. Extended Abstract for the Guangzhou Symposium on Satisfiability and its Applications, September 2004.

[29] 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.

[30] Oliver Kullmann and Horst Luckhardt. Algorithms for SAT/TAUT decision based on various measures. Preprint, 71 pages; the ps-file can be obtained from `http://cs-svr1.swan.ac.uk/~csoliver`, December 1998.

[31] Nathan Linial and Michael Tarsi. Deciding hypergraph 2-colourability by H-resolution. *Theoretical Computer Science*, 38:343–347, 1985.

[32] Lengning Liu and Miroslaw Truszczyński. Local search with boostrapping. In Holger H. Hoos and David G. Mitchell, editors, *The Seventh International Conference on Theory and Applications of Satisfiability Testing*, pages 299–304, Vancouver, British Columbia, Canada, May 2004. Data for results on van der Waerden numbers available at `http://www.cs.uky.edu/~marek`.

[33] Steven Prestwich. Local search on SAT-encoded colouring problems. In Giunchiglia and Tacchella [16], pages 105–119. ISBN 3-540-20851-8.

[34] Alexander Schrijver. *Combinatorial Optimization*, volume A. Springer, Berlin, 2003. ISBN 3-540-44389-4; Series Algorithms and Combinatorics, no. 24.

[35] P.D. Seymour. On the two-colouring of hypergraphs. *The Quarterly Journal of Mathematics (Oxford University Press)*, 25:303–312, 1974.

[36] Stefan Szeider. Minimal unsatisfiable formulas with bounded clause-variable difference are fixed-parameter tractable. *Journal of Computer and System Sciences*, 69(4):656–674, 2004.

[37] C. A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8:85–89, 1984.

[38] Hans van Maaren. A short note on some tractable cases of the satisfiability problem. *Information and Computation*, 158(2):125–130, May 2000.

[39] Xishun Zhao and Ding Decheng. Two tractable subclasses of minimal unsatisfiable formulas. *Science in China (Series A)*, 42(7):720–731, July 1999.