Clause set structures and satisfiability

Stefan Porschen and Ewald Speckenmeyer

Institut für Informatik, Universität zu Köln, D-50969 Köln, Germany.
{porschen,esp}@informatik.uni-koeln.de

Abstract. We propose a new perspective on propositional clause sets and on that basis we investigate (new) polynomial time SAT-testable classes. Moreover, we study autarkies using a closure concept. A specific simple type of closures the free closures leads to a further formula class called hyperjoins that is studied w.r.t. SAT.

Key Words: CNF satisfiability, hypergraph, fibre-transversal, autarky, closure operator

1 Introduction

The intention of the present paper is to investigate certain structural properties of clause sets representing CNF formulas. Exploiting these properties we also search for subclasses of clause sets for which satisfiability is testable in polynomial time.

As the main topic, via introducing some concepts, we propose to regard clause sets from a slightly different perspective, namely as pairs of mutually set-complemented formulas with respect to the total clause set over a common intrinsic hypergraph called the *base hypergraph*. This yields no new structure from the logical point of view, because each clause set defined as usual and behaving not trivial regarding satisfiability can easily be seen to correspond to such a clause set pair. To the best of our knowledge this approach is new, and the hope is that this perspective may help to gain new structural insight into clause sets and even new algorithmic concepts. To supply such a hope we develop the basic theory around the concepts introduced and provide some new clause set classes that we prove to be polynomial time solvable via the methods presented. So e.g. satisfiability of CNF formulas, where every two distinct clauses share exactly one common variable or all (neglecting negations) can be decided in polynomial time, cf. Theorem 3. In order to establish our theory we have to introduce new concepts and notions, and we are not aware of a similar approach representing our view at satisfiability in a convenient framework.

There are several polynomial time SAT-testable classes known, as quadratic formulas, (q-)Horn formulas, matching formulas etc. [3, 4, 6, 7, 9, 10, 1, 15]. Moreover it is known that mixing polynomial-time classes, in general, yields classes for which SAT becomes NP-complete, cf. e.g. [8, 12].

As a second topic considering autarkies we introduce the notion of an *autarky closure (hull)*, helping to restrict the subsets of the variable set of a given formula that have to be tested for autarky. The simplest class of such hulls are so-called *free* hulls; we show that in a given formula existency of free hulls can be checked in polynomial time, and also whether such a free hull indeed admits an autark assignment for the formula. A specific class of formulas, so-called *hyperjoins*, is defined and studied in the last section. Such formulas Such formulas have a well-behaved free-hull-structure and might be taken as playground candidates for studying more complex autarky hulls than the free ones. We show that SAT for hyperjoins can be decided in polynomial time using the results provided before.

To fix notation let CNF denote the set of formulas (free of duplicate clauses) in conjunctive normal form over propositional variables $x_i \in \{0,1\}$. A variable x induces a positive literal (variable x) or a negative literal (negated variable: $\bar{x}$). The *complement* of a literal l is $\bar{l}$. Each formula $C \in$ CNF is considered as a set of its clauses $C = \{c_1, \ldots, c_{|C|}\}$ having in mind that it is a conjunction of these clauses. Each clause $c \in C$ is a disjunction of different literals, and is also represented as a set $c = \{l_1, \ldots, l_{|c|}\}$. A clause $c \in C$ is called *unit* iff $|c| = 1$. For a given formula C, clause c, by $V(C), V(c)$ we denote the set of variables occuring (negated or unnegated) in C resp. c. For a variable x, $l(x) \in \{x, \bar{x}\}$ denotes a fixed literal over x. Similarly, $L(C)$ denotes the set of all literals in C. The length of a formula C is denoted by $\|C\|$ whereas $|C|$, as usual, denotes the number of its clauses.

$\mathrm{CNF}_\varepsilon, \varepsilon \in \{+,-\}$, denotes the set of ε-*monotone* (CNF-)formulas, i.e., for $\varepsilon = +$ (resp. $-$) all clauses are positive (resp. negative) monotone. For $k \in \mathbb{N}$, let k-CNF denote the subset of formulas such that each clause has length at most k. We consider some subformulas of a formula C. For $U \subset V(C)$, we define $\hat{C}(U) := \{c \in C : V(c) \cap U \neq \varnothing\}$ and similarly, for $U' \subset L(C)$, we set $C(U') := \{c \in C : c \cap U' \neq \varnothing\}$. If $U = \{x\}$ resp. $U' = \{l\}$, we simply write $\hat{C}(x)$ resp. $C(l)$. Moreover, for $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$, and for a set M denote its power set by 2^M, and its k-sets by $\binom{M}{k}$.

The satisfiability problem (SAT) asks in its *decision* version, whether a given CNF instance C is *satisfiable*, i.e., whether C has a *model*, which is a truth assignment $t : V(C) \to \{0,1\}$ assigning at least one literal in each clause of C to 1. We will assume throughout that clauses do not contain complemented literals, as such clauses are trivially satisfiable and could be removed from a formula in linear time. For, $C \in \mathrm{SAT} := \{C \in \mathrm{CNF} : C \,\text{has a model}\}$, let $M(C)$ be the space of all (total) models of C, and UNSAT := CNF $-$ SAT. For convenience we allow the empty set to be a formula: $\varnothing \in$ CNF which is always satisfiable. In its *search* version SAT means to find a model t if the input formula is satisfiable.

It turns out to be convenient to identify truth assignments with vectors in the following simple way: Assume variables are enumerated according to $V = \{x_1, \ldots, x_n\}$, and let $x^0 := \bar{x}$, $x^1 := x$. Then we can identify a truth assignment $t : V \to \{0,1\}$ with the ordered literal set $\{x_1^{t(x_1)}, \ldots, x_n^{t(x_n)}\}$, and, for $b \subset V$, the *restriction* $t|_b$ of t to b is identified with the (ordered) literal set $\{x^{t(x)} : x \in b\}$. Let W_V denote the collection of the literal sets obtained in the described way by running through the set of all total truth assignments $V \to \{0,1\}$. Interpreting its members as clauses, we call W_V the *hypercube formula (over V)* because its clauses are in 1:1-correspondence with the vertex set of a hypercube of dimension $|V|$. E.g., for $V = \{x, y\}$ we have $W_V = \{xy, \bar{x}y, x\bar{y}, \bar{x}\bar{y}\}$ where we used the convention to write clauses as strings of the contained literals.

For a clause c we denote by c^γ the clause in which all its literals are complemented. Regarding truth assignments t as above as clauses also t^γ makes sense, which in the identification above, corresponds to the truth assignment $t^\gamma = 1 - t : V \to \{0,1\}$. For formula C let $C^\gamma := \{c^\gamma : c \in C\}$, and given a formula class $\mathcal{C} \subseteq$ CNF let $\mathcal{C}^\gamma := \{C^\gamma : C \in \mathcal{C}\}$. A formula $C \in$ CNF is called *symmetric* if $C = C^\gamma$, thus for each clause $c \in C$ holds $c^\gamma \in C$. Similarly, we call a formula $C \in$ CNF *asymmetric* if for each $c \in C$ holds $c^\gamma \notin C$. Let Sym $\subset$ CNF, resp. Asym $\subset$ CNF, denote the collection of all symmetric, resp. asymmetric, formulas. It is obvious that an arbitrary $C \in$ CNF has a unique decomposition $C = C_S \cup C_A$ where $C_S \in$ Sym is the largest symmetric subformula contained in C and $C_A = C \setminus C_S \in$ Asym is the remaining asymmetric subformula.

2 A New View on Clause Sets: Basic Concepts and Results

A *(variable-) base hypergraph* $\mathcal{H} = (V, B)$ is a hypergraph whose vertices $x \in V$ are regarded as Boolean variables such that for each $x \in V$ there is a (hyper)edge $b \in B$ with $x \in b$, so B can be considered as a positive monotone clause set. As above, for any $b \in B$, W_b is the hypercube formula over b. $K_\mathcal{H} := \bigcup_{b \in B} W_b$ denotes the set of all possible clauses over $\mathcal{H}$, and is called the *total clause set over $\mathcal{H}$*. Regarding each b as a *point* in the space B, we obtain the following mapping

$$\pi : K_\mathcal{H} \ni c \mapsto V(c) \in B$$

recalling that $V(c)$ is the set of variables in clause c. We call $\pi^{-1}(b) = W_b$ the *fibre of* $K_\mathcal{H}$ over b. Obviously the fibres are mutually disjoint and π is surjective, thus is a projection.

A *formula over $\mathcal{H}$ (or $\mathcal{H}$-formula)* is any subset $C \subset K_\mathcal{H}$ such that $C \cap W_b \neq \varnothing$ for each $b \in B$, implying that the restriction $\pi_C := \pi|_C$ yields a projection $\pi_C : C \to B$, let $C_b := \pi_C^{-1}(b) \subseteq W_b$ denote the *fibre(-subformula)* of C over b. Note that a $C \in$ CNF is a formula over the base hypergraph $\mathcal{H}(C) := (V(C), B(C))$ with $B(C) := \{V(c) : c \in C\}$). For each $C \subset K_\mathcal{H}$ such that $(*)$: $W_b - C_b \neq \varnothing$, for all $b \in B$, the *$\mathcal{H}$-based complement formula* $\bar{C}$ of C is defined by $\bar{C} := \bigcup_{b \in B}(W_b - C_b)$. By construction $\bar{C}$ has the same base hypergraph as C.[1] A *fibre-transversal (f-transversal)* of $K_\mathcal{H}$ is a subset $F \subset K_\mathcal{H}$ such that $|F \cap W_b| = 1$, for each $b \in B$. Hence F contains exactly one clause

[1] Clearly, any hypercube formula is unsatisfiable, therefore in case that C does not have property $(*)$ it is unsatisfiable trivially, which therefore can be ruled out. More precisely, it can be treated by a simple preprocessing checking in linear time whether there is $b \in B$ such that $W_b = C_b$.

of each fibre $\pi^{-1}(b)$ of $K_{\mathcal{H}}$, let that clause be referenced by $F(b)$.[2] Let $\mathcal{F}(K_{\mathcal{H}})$ denote the set of all f-transversals of $K_{\mathcal{H}}$. Similarly, for a specified formula C over $\mathcal{H}$, $\mathcal{F}(C)$ (resp. $\mathcal{F}(\bar{C})$) denotes the set of f-transversals of C (resp. $\bar{C}$).

Definition 1 *Let $\mathcal{H} = (V, B)$ and $K_{\mathcal{H}}$ as defined above.*
(1) $F \in \mathcal{F}(K_{\mathcal{H}})$ is called compatible *if $\bigcup_{b\in B} F(b) \in W_V$, meaning that F contains each variable of W as a pure literal. Let $\mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ be the collection of all compatible f-transversals of $K_{\mathcal{H}}$.*
(2) $F \in \mathcal{F}(K_{\mathcal{H}})$ is called diagonal *if for each $F' \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ holds $F \cap F' \neq \varnothing$. Let $\mathcal{F}_{\text{diag}}(K_{\mathcal{H}})$ be the collection of all diagonal f-transversals of $K_{\mathcal{H}}$.*
(3) For any $\mathcal{H}$-based formula $C \subseteq K_{\mathcal{H}}$, let $\mathcal{F}_{\text{comp}}(C) := \mathcal{F}(C) \cap \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ and $\mathcal{F}_{\text{diag}}(C) := \mathcal{F}(C) \cap \mathcal{F}_{\text{diag}}(K_{\mathcal{H}})$.

A priori it is not clear whether diagonal transversals exist at all, a question that will be addressed below. However, if there are diagonal transversals, then each fixed compatible transversal in turn meets all diagonal transversals.

We have some simple observations:

Proposition 1 *(1) $\mathcal{F}_{\text{comp}}(K_{\mathcal{H}}) \cong W_V$,*
(2) $\mathcal{F}_{\text{diag}}(K_{\mathcal{H}}) = \{F \in \mathcal{F} : \forall t \in W_V \exists b \in B : F(b) = t|_b\}$,
(3) $\mathcal{F}_{\text{comp}}(K_{\mathcal{H}})^\gamma = \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$,
(4) $\mathcal{F}_{\text{diag}}(K_{\mathcal{H}})^\gamma = \mathcal{F}_{\text{diag}}(K_{\mathcal{H}})$,
(5) $\mathcal{F}_{\text{diag}}(K_{\mathcal{H}}) \cap \mathcal{F}_{\text{comp}}(K_{\mathcal{H}}) = \varnothing$.

Proof. Assertion (1) is easily obtained by observing that

$$\varphi : \mathcal{F}_{\text{comp}}(K_{\mathcal{H}}) \ni F \mapsto \bigcup_{b\in B} F(b) \in W_V$$

is a bijection with $[\varphi^{-1}(t)](b) := t|_b$ for each $t \in W_V$, $b \in B$, recalling that by assumption $\bigcup_{b\in B} b = V$. (2) follows immediately from (1).

Assertion (3) is obvious, and implies $\varphi(F^\gamma) = \varphi(F)^\gamma$, for $F \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$.

Let $F \in \mathcal{F}_{\text{diag}}(K_{\mathcal{H}})$ and assume there is $F' \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ such that $F'(b) \neq F^\gamma(b)$ for all $b \in B$ equivalent to $F'^\gamma(b) \neq F(b)$ for all $b \in B$, by (3) contradicting that F is diagonal yielding (4).

Assume $F \in \mathcal{F}_{\text{diag}}(K_{\mathcal{H}}) \cap \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$, then by (3) also $F^\gamma \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ but $F^\gamma(b) \neq F(b)$ for each $b \in B$ therefore $F \notin \mathcal{F}_{\text{diag}}(K_{\mathcal{H}})$ yielding a contradiction implying (5). □

The next assertion essentially states that a formula C is satisfiable if and only if its complement formula admits a compatible f-transversal:

Theorem 1 *For $\mathcal{H} = (V, B)$, and $K_{\mathcal{H}}$ let $C \subset K_{\mathcal{H}}$ be a $\mathcal{H}$-formula such that $\bar{C} \subset K_{\mathcal{H}}$ also is a $\mathcal{H}$-formula (hence $B(C) = B = B(\bar{C})$), we have:*
(i) $C \in \text{SAT}$ if and only if $\mathcal{F}_{\text{comp}}(\bar{C}) \neq \varnothing$.
(ii) If $C \in \text{SAT}$ then $M(C) \cong \mathcal{F}_{\text{comp}}(\bar{C})$.

[2] $K_{\mathcal{H}}$ can be viewed as a hypergraph having all literals over V as vertex set. So, a fibre-transversal should not be mixed up with a hypergraph-transversal which, as usually defined, is a subset of its vertex set meeting all its edges, thus is a hitting set.

PROOF. We claim that if $C \in \text{SAT}$, hence $W_V \supseteq M(C) \neq \emptyset$, then $\mathcal{F}_{\text{comp}}(\bar{C}) = \varphi^{-1}(M(C)^\gamma)$, where φ is defined as in the proof of Prop. 1 (1). From this claim (ii) follows, as obviously $M(C)^\gamma \cong M(C)$ and φ is a bijection. Further (i) follows: If $M(C)$ is empty then also $\mathcal{F}_{\text{comp}}(\bar{C})$ must be empty, otherwise by the claim holds $\varphi(F)^\gamma \in M(C)$, for any $F \in \mathcal{F}_{\text{comp}}(\bar{C})$, yielding a contradiction. The reverse direction of (i) is immediately implied by the claim.

So it remains to verify the claim: Let $t \in M(C)$ be chosen arbitrarily. Clearly, $F_t := \varphi^{-1}(t^\gamma) \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ by definition of φ. Suppose there is $b \in B$ such that $F_t(b) = t^\gamma|_b \in C$. Clearly, $t|_b$ is a total truth assignment of the hypercube formula W_b satisfying all of its clauses except $(t|_b)^\gamma = t^\gamma|_b \in C$ thus $t \notin M(C)$ contradicting the assumption. Therefore $F_t(b) \in \bar{C}$, for all $b \in B$, hence $F_t \in \mathcal{F}_{\text{comp}}(\bar{C})$ thus $\varphi^{-1}(M(C)^\gamma) \subseteq \mathcal{F}_{\text{comp}}(\bar{C})$. Conversely, let $F \in \mathcal{F}_{\text{comp}}(\bar{C})$ then we claim that $t_F := \varphi(F)^\gamma \in M(C) \subseteq W_V$. Indeed, supposing the contrary, there is $b \in B$ with $t_F^\gamma|_b \in C$ equivalent to $F(b) = \varphi(F)|_b \notin \bar{C}$ contradicting the assumption and finishing the proof because $\varphi^{-1}(t_F^\gamma) = F$. □

Proposition 2 *Let $F \in \mathcal{F}(K_{\mathcal{H}})$, then holds*

$$(1) \quad F \in \mathcal{F}_{\text{diag}}(K_{\mathcal{H}}) \Leftrightarrow F \in \text{UNSAT}$$
$$(2) \quad F \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}}) \Rightarrow F \in \text{SAT}$$

PROOF. By Theorem 1 (i), we have $F \in \text{UNSAT}$ iff $\mathcal{F}_{\text{comp}}(\bar{F}) = \emptyset$ iif $\forall F' \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ there is $b \in B$ such that $F'(b) = F(b) \in F$ iif $F \in \mathcal{F}_{\text{diag}}(K_{\mathcal{H}})$, hence (1). (2) is implied by (1) due to Prop. 1 (5); moreover for $F \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$, $\varphi(F) \in W_V$ specifically satisfies F. □

Thus, we have three types of possible f-transversals composing $\mathcal{F}(K_{\mathcal{H}})$, namely compatible f-transversals which always are satisfiable formulas, diagonal ones (which may not exist) which always are unsatisfiable, and, finally, f-transversals that are neither compatible nor diagonal but always are satisfiable.

Definition 2 *A formula $D \subseteq K_{\mathcal{H}}$ is called a* diagonal *formula if for each $F \in \mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ holds $F \cap D \neq \emptyset$.*

Obviously each $F \in \mathcal{F}_{\text{diag}}(K_{\mathcal{H}})$ (if existing) is a diagonal formula. Since a diagonal formula D contains a member of each compatible f-transversal the complement formula $\bar{D}$ cannot have a compatible f-transversal. Therefore $D \in \text{UNSAT}$ due to Theorem 1, and we have:

Proposition 3 *A formula is unsatisfiable iff it contains a subformula that is diagonal.* □

Consider a simple application of the concepts above: Recall that a hypergraph is called *Sperner* (or sometimes also called *simple*) if no hyperedge is contained in another hyperedge [2]. Clearly, a formula C, regarded as a hypergraph $(L(C), C)$ over its literal set, that is non-Sperner can be turned into a SAT-equivalent one having that property: For clauses c, c' with $c \subset c'$ we can remove c' from C because c already has to be satisfied implying satisfiability of c'. Let C be Sperner then its base hypergraph $\mathcal{H}(C) = (V(C), B(C))$ can either be Sperner or non-Sperner, assume $\mathcal{H}(C) = \mathcal{H}(\bar{C})$. Clearly, if $\mathcal{H}(C)$ is Sperner then so is $\bar{C}$. Specifically, all these objects are Sperner if all clauses have the same length. However, if $\mathcal{H}(C)$ is non-Sperner, $\bar{C}$ can be Sperner or non-Sperner. For the first case, i.e., C and $\bar{C}$ Sperner but $\mathcal{H}(C) = \mathcal{H}(\bar{C})$ non-Sperner, consider the following example (simply representing clauses as strings of the literals contained):

$$\begin{aligned} C &= \{xy, x\bar{y}z, \bar{x}yz, \bar{x}\bar{y}z, x\bar{y}\bar{z}, \bar{x}y\bar{z}, \bar{x}\bar{y}\bar{z}\} \\ \bar{C} &= \{x\bar{y}, \bar{x}y, \bar{x}\bar{y}, xyz, xy\bar{z}\} \\ B(C) &= \{xy, xyz\} \end{aligned}$$

Theorem 2 *Let $C \in$ CNF be Sperner such that its base hypergraph $\mathcal{H}(C)$ is non-Sperner but the complement formula $\bar{C}$ is Sperner: Then both C and $\bar{C}$ are unsatisfiable.*

Proof. According to Theorem 1 we show that $\bar{C}$ cannot have a compatible f-transversal under the assumptions stated above. Because $\mathcal{H}(C)$ non-Sperner there are $b, b' \in B(C)$ with $b \subset b'$ and $b \neq b'$. Now for each f-transversal $F \in \mathcal{F}(\bar{C})$ holds $F(b) \not\subset F(b')$ as $\bar{C}$ is assumed to be Sperner. That means there is $x \in b$ such that $x \in F(b)$, $\bar{x} \in F(b')$ or vice versa, hence $F(b) \cup F(b') \supset \{x, \bar{x}\}$ is not compatible implying that $C \in$ UNSAT. By exchanging the roles of C and $\bar{C}$ we also obtain that $\bar{C}$ cannot be satisfiable. □

Corollary 1 *If C is Sperner and satisfiable then either*
(i) $\mathcal{H}(C)$ and $\bar{C}$ both are Sperner or
(ii) $\mathcal{H}(C)$ and $\bar{C}$ both are non-Sperner and for each two $b_1 \subset b_2 \in B(C)$ there are $c_1 \subset c_2 \in \bar{C}$ such that $V(c_i) = b_i$, $i = 1, 2$.

Remark 1 *The criterion in (ii) of the Corollary is not sufficient for satisfiability of C: Let $b_1 \subset b \in B(C)$ such that $c_1 \subset c \in \bar{C}$ and moreover let $b'_1 \subset b' \in B(C)$ such that $c'_1 \subset c' \in \bar{C}$ where $V(c) = b$, $V(c') = b'$, $V(c_1) = b_1$, and $V(c'_1) = b'_1$. Now assume that $b \cap b' \neq \varnothing$, and that c, c' are the only clauses over b, b' in $\bar{C}$. Clearly, if $c|_{b\cap b'} \neq c'|_{b\cap b'}$ then there is no compatible f-transversal of $\bar{C}$ hence no model of C.*

Returning to the general discussion, let $\mathcal{H} = (V, B)$ be a non-empty base hypergraph, then clearly $\mathcal{F}_{\text{comp}}(K_{\mathcal{H}})$ is not empty we even have $|\mathcal{F}_{\text{comp}}(K_{\mathcal{H}})| = 2^{|V|}$ due to Prop. 1 (1). However, a priori it is not clear whether also holds $\mathcal{F}_{\text{diag}}(K_{\mathcal{H}}) \neq \varnothing$ in any case. It turns out that this depends strongly on the structure of the base hypergraph $\mathcal{H}$: To that end, let us consider an interesting and guiding example regarding satisfiability of certain formulas over *(exactly) linear* base hypergraphs. In [13, 14] linear formulas (variable sets of distinct clauses have at most one member in common) are discussed in more detail and satisfiability of exactly linear formulas is shown by simple matching techniques.

Lemma 1 *[13] Any exactly linear formula C is satisfiable.*

From the last result we immediately conclude that if the base hypergraph $\mathcal{H} = (V, B)$ is exactly linear then for the corresponding total clause set holds $\mathcal{F}_{\text{diag}}(K_{\mathcal{H}})$

$= \varnothing$. Indeed, then no unsatisfiable f-transversal can exist, as each is exactly linear and we are done by Proposition 2. This answers the earlier stated question whether there are hypergraphs admitting no diagonal f-transversal. The reverse question, namely are there hypergraphs at all such that the total clause sets has diagonal f-transversals, also is answered positive: In [13, 14] it is shown that there are unsatisfiable linear formulas, these formulas must be f-transversals hence correspond to diagonal f-transversals of the total clause set over the underlying base hypergraph:

Fact 1 *The notion of (diagonal) f-transversals immediately generalizes the notion of (unsatisfiable) linear formulas.*

However formulas having an exactly linear base hypergraph in general may not be satisfiable, because they can contain a diagonal subformula (which cannot be a f-transversal):

Theorem 3 *Let $\mathcal{H} = (V, B)$ be an exactly linear base hypergraph with corresponding total clause set $K_{\mathcal{H}}$. Let $C \subseteq K_{\mathcal{H}}$ be any $\mathcal{H}$-formula. Then we can check in polynomial time whether C contains a diagonal subformula, i.e., whether $C \in$ UNSAT.*

PROOF. Recall that $C_b \subset W_b$ denotes the fibre subformula of C over $b \in B$, and that $C_b(l) \subset C_b$ is the subformula of C_b of all clauses containing literal l, where $V(l) \in b$. Let l be an arbitrary literal occuring in C, and first observe that, if $b \in B$ is an edge containing the underlying variable $V(l) \in b$, then the clauses in C_b cover (i.e., have intersection with) exactly $\mu(l) := \mu(l, b) := |C_b(l)| \cdot 2^{n-|b|}$ of all 2^{n-1} truth assignments containing l, where $n := |V|$.

We intend to determine the number of truth assignments met by the clauses in the input formula C. This essentially is organized by performing two independent runs of a Procedure `ComputeCoverNumber`(l, p), one for $l = x$ and a second one for $l = \bar{x}$. Here x is the *maximum variable* that together with the determined edges $b_1, b_1' \in B$ (smallest index if ambigous), has to be computed first according to

$$\mu(x, b_1) + \mu(\bar{x}, b_1') = \max\{\mu(y, b) + \mu(\bar{y}, b') : y \in L(C_b), \bar{y} \in L(C_{b'}), b, b' \in B\}$$

here $\mu(l, b) = |C_b(l)| \cdot 2^{n-|b|}$ is computed for all $(l, b) \in L(V) \times B$ such that $C_b(l) \neq \varnothing$. It is possible that $b_1 = b_1'$.

Both executions of Procedure `ComputeCoverNumber`(l, p) are initiated only if $\mu(l) < 2^{n-1}$ meaning that the fibre subformula corresponding to the maximum does not cover all 2^{n-1} possible truth assignments containing l. Finally, the corresponding cover numbers returned in p are added, and the algorithm returns unsatisfiable iff the total value equals 2^n. Clearly, the runs of the procedure for x and $\bar{x}$ can be processed independently because both compute coverings in different ranges in the set of all truth assignments

Now procedure `ComputeCoverNumber`(l, p) consists of two main subprocedures. A first is entered only if there is at least one fibre subformula C_b containing l besides C_{b_1} and computes all additional truth assignments containing

x covered by these fibre subformulas. The second subprocedure is entered only in case there are any remaining fibre subformulas not containing l, and the subprocedure is devoted to determine all additionally covered truth assignments containing l covered by these subformulas.

The first subprocedure proceeds as follows: W.l.o.g. (otherwise relabel the members in B) let $\{C_{b_2}, \ldots, C_{b_s}\}$, for $s \geq 1$, denote the collection of all remaining fibre subformulas with $V(l) \in b_i$, $2 \leq i \leq s$. Assume that its members are ordered due to decreasing cardinalities of its subsets $|C_{b_j}(l)|$ containing l, for $2 \leq j \leq s$.

For simplicity let $m_j := |C_{b_j}(l)|$ and $m'_j := |W_{b_j}(l) - C_{b_j}(l)| = 2^{|b_j|-1} - |C_{b_j}(l)|$, for $1 \leq j \leq s$. Then the number of truth assignments containing l covered by the subformulas in C_l is given by:

$$(*) \quad m'_1 \sum_{j=2}^{s} \left[m_j \cdot 2^{n+(j-1)-\sum_{q=1}^{j} |b_q|} \cdot \prod_{k=2}^{j-1} m'_k \right]$$

where, as usual, $\prod_{i=b}^{k} a_i := 1$, for $k < b$.

Clearly, number $(*)$ can be determined performing a simple loop recalling that by assumption $m_j > 0$, for all $1 \leq j \leq s$:

$z \leftarrow m'_1 \cdot m_2 \cdot 2^{n+1-|b_1|-|b_2|}$
$p \leftarrow z$
for $j = 2$ **to** $s - 1$ **do**
  $z \leftarrow z \cdot m'_j \cdot \frac{m_{j+1}}{m_j} \cdot 2^{1-|b_{j+1}|}$
  $p \leftarrow p + z$
od

So finally, we have to check whether the resulting value $p = 2^{n-1}$. In order to avoid calculations with possibly large number 2^n it is sufficient instead to compute $p' := p/2^n$ and finally checking whether $p' = 1/2$. Observe that the second subprocedure needs to be started only if the answer is negative.

For explaining the second subprocedure, let c be any clause of a fibre subformula over $b \in B - B(x)$, then c covers a truth assignment containing l if and only if for each $b_i \in B(x)$ there are $c_i \in W_{b_i} - C_{b_i}(x)$ with $c \cap c_i \neq \emptyset$. Observe that none of these truth assignments is covered by those computed in the first subprocedure, because each of the latter ones fixes all literals of at least one complete clause in any x-fibre subformula whereas each of the newly as covered determined truth assignments are composed of missing clauses in each hypercube formula $W_b(l) - C_b(l)$, for all $e \in B(x)$. So each corresponding truth assignment is different to each detected in the first subprocedure in at least one position.

W.l.o.g. (which always can be achieved via relabeling), let $\mathcal{C}(x) := \{C_{b_{s+1}}, \ldots, C_{b_{s+r}}\}$, for $r \geq 1$, be the collection of all fibre subformulas neither containing x nor $\bar{x}$, hence it has exactly one member for each edge in $B - B(x)$. For $C_{b_{s+1}} \in C'_x$ and $c \in C_{b_{s+1}}$, let $\{y_i\} = V(c) \cap C_{b_i}$, $1 \leq i \leq s$, which are uniquely determined because of exact linearity. Assume that $l_i \in c$ is the corresponding literal with $V(l_i) = y_i \neq x$, where clearly $|c| \geq s$ and each variable in c different from y_i, $1 \leq i \leq s$, cannot occur in any member of C_l.

Let $n_l := \sum_{q=1}^{j} |b_q| - (s-1)$ be the number of variables already fixed by $b_i, 1 \leq i \leq s$. Let $\lambda_i(c) := |c \cap [W_{b_i}(l) - C_{b_i}(l)]|$ be the number of occurences of literal l_i in $W_{b_i}(l) - C_{b_i}(l)$ which is the fibre complement of $C_{b_i}(l)$. Clearly l_i occurs in exactly $2^{|b_i|-2}$ clauses in $W_{b_i}(l)$. So, if l_i occurs t_i times in $C_{b_i}(l)$, we obvioulsy have

$$\lambda_i(c) = 2^{|b_i|-2} - t_i$$

Now the clauses in $C_{b_{s+1}}$ exactly cover the following number of additional truth assignments containing l:

$$2^{n-n_l-(|b_{s+1}|-s)} \sum_{c \in C_{b_{s+1}}} \prod_{j=1}^{s} \lambda_j(c)$$

Therefore, we obtain for the number of covered truth assignments containing l by all members of C'_x,

$$\sum_{k=1}^{r} \left[2^{n-n_l-\sum_{j=1}^{k} f(j)} \sum_{c \in C_{b_{s+k}}} \left(\prod_{j=1}^{s+k-1} \lambda_j(c) \right) \right]$$

where

$$f(j) := |b_{s+j}| - \left| \bigcup_{i=1}^{s+j-1} (b_{s+j} \cap b_i) \right| \in \{0, \ldots, |b_{s+j}| - s\}$$

$1 \leq j \leq r$.

Having processed `ComputeCoverNumber`(l,p) for $l := x$ we again check whether $p' = 1/2$ and only in the positive case we run `ComputeCoverNumber`(l,p) for $l := \bar{x}$, because otherwise not all truth assignments containing x are covered immediately enabling us to conclude that $C \in$ SAT. □

Obviously, the method above is not able to solve the search problem, we only obtain a decision whether C is satisfiable, but in positive case we are not aware of a model.

So, there are cases where no diagonal f-transversal of the total clause set exists, but unsatisfiable formulas $C \subset K_{\mathcal{H}}$ can exist although, so we conclude that, despite of Proposition 2, in general $C \in$ UNSAT is not equivalent to $\mathcal{F}(C) \neq \varnothing$. However, things may be different if H is structured such that $\mathcal{F}_{\text{diag}}(K_{\mathcal{H}}) \neq \varnothing$. So, we next pose the question whether under this assumption holds $C \in$ UNSAT iff $\mathcal{F}_{\text{diag}}(C) \neq \varnothing$. Observe that the implication $\Leftarrow$ holds because if C admits a diagonal f-transversal then $\bar{C}$ cannot have a compatible f-transversal therefore $C \in$ UNSAT due to Theorem 1 (i).

Definition 3 *Let $\mathcal{H} = (V, B)$ be a base hypergraph.*
We call $\mathcal{H}$ a diagonal base hypergraph *if $\mathcal{F}_{\text{diag}}(K_{\mathcal{H}}) \neq \varnothing$.*
$\mathcal{H}$ is called strictly diagonal *if it is diagonal, and additionally:*

$$(*) : \forall C \subset K_{\mathcal{H}} : B(C) = B = B(\bar{C}) : C \in \text{UNSAT} \Leftrightarrow \mathcal{F}_{\text{diag}}(C) \neq \varnothing$$

We first consider the question whether the class of strictly diagonal base hypergraphs coincides with the class of all diagonal base hypergraphs. To give an answer constructively: Start with a linear hypergraph $\mathcal{H} = (V, B)$ that admits an unsatisfiable polarization hence admits a diagonal f-transversal and therefore $\mathcal{H}$ is diagonal. Assume that $\mathcal{H}$ results by a block construction over a base block hypergraph $\mathcal{H}' = (V', B')$ with $V' \subset V, B' \subset B$ as shown in [13]. Then we claim that we can construct a (small) unsatisfiable formula $C' \subset K_{\mathcal{H}'}$ with $\pi'(C') = B' = \pi'(\bar{C}')$. Now we claim that it is possible to add to C' exactly one member of each fibre of $\pi^{-1}(b)$, forall $b \in B - B'$ such that the resulting formula C has the property that each of its f-transversals is satisfiable, hence cannot be diagonal. Thereofore the above stated question gets a negative answer. The question whether there exist strictly diagonal hypergraphs is still open.

Lemma 2 *For $\mathcal{H}$ strictly diagonal holds that each f-transversal meeting all diagonal f-transversals is compatible, formally:*

$$\mathcal{F}_{\mathrm{comp}}(K_{\mathcal{H}}) = \{F \in \mathcal{F}(K_{\mathcal{H}}) : \forall F' \in \mathcal{F}_{\mathrm{diag}}(K_{\mathcal{H}}) : F \cap F' \neq \varnothing\}$$

Proof. Let $F \in \mathcal{F}(K_{\mathcal{H}})$ meeting all members of $\mathcal{F}_{\mathrm{diag}}(K_{\mathcal{H}})$ then clearly F cannot be diagonal as F^{γ} is diagonal. If F is compatible we are done. So assume that F is neither compatible nor diagonal, then specifically $\bar{F} \in$ UNSAT due to Theorem 1. Since $\mathcal{H}$ is strictly diagonal it is implied that $\mathcal{F}_{\mathrm{diag}}(\bar{F}) \neq \varnothing$ meaning there is $F' \in \mathcal{F}_{\mathrm{diag}}(K_{\mathcal{H}})$ such that $F \cap F' = \varnothing$. □

We next provide some further considerations, besides the discussion of diagonality: For $\mathcal{H} = (V, B)$, again let $C \subset K_{\mathcal{H}}$ such that $B(C) = B = B(\bar{C})$. If $C \in$ SAT then due to Prop. 1 (1) each $t \in M(C)$ satisfies $\varphi^{-1}(t) \in \mathcal{F}_{\mathrm{comp}}(K_{\mathcal{H}})$. We now address the question in which case each model t of C even satisfies $\varphi^{-1}(t) \in \mathcal{F}_{\mathrm{comp}}(C)$, i.e., corresponds to a compatible f-transversal of the formula itself.

Lemma 3 *If $C \in$ CNF$\cap$SAT, $B(C) = B(\bar{C})$, such that for each $t \in M(C)$ holds $\varphi^{-1}(t) \in \mathcal{F}_{\mathrm{comp}}(C)$ then $\bar{C} \in$ SAT and $\varphi(F) \in M(\bar{C})$ for each $F \in \mathcal{F}_{\mathrm{comp}}(\bar{C})$; and vice versa.*

Proof. Let $F \in \mathcal{F}_{\mathrm{comp}}(\bar{C})$ be arbitrary with $t := \varphi(F) \in W_V$, then according to the proof of Theorem 1 (ii) t^{γ} is a model of C. By assumption there is $F' \in \mathcal{F}_{\mathrm{comp}}(C)$ such that $t^{\gamma} = \varphi(F')$. Hence, again by Theorem 1 (ii), t is a model of $\bar{C}$ as claimed, specifically $\bar{C} \in$ SAT. The vice versa assertion follows by exchanging the roles of C and $\bar{C}$. □

Next we provide a formula class admitting the assumption of the last lemma. Recall that a symmetric formula satisfies $C = C^{\gamma}$ Clearly, if $C \in$ Sym then also $\bar{C} \in$ Sym, since for $c \in \bar{C}$ holds $c \notin C$ thus $c^{\gamma} \notin C$ implying $c^{\gamma} \in \bar{C}$.

Lemma 4 *Let $C \in$ CNF such that $B(C) = B(\bar{C})$ and $\bar{C} \in$ Asym. Then $C \in$ SAT implies $\bar{C} \in$ SAT and each $t \in M(C)$ satisfies $\varphi^{-1}(t) \in \mathcal{F}_{\mathrm{comp}}(C)$; and vice versa.*

Proof. Let $t \in M(C) \neq \varnothing$ then for each $b \in B(C)$ $t|_b$ satisfies all of W_b except for $(t|_b)^\gamma$ which thus must be a clause of $\bar{C}$. And $\bar{C} \in \text{Asym}$ implies that $t|_b \in C$ for each $b \in B(C)$. Hence $\{t|_b : b \in B(C)\}$ is a compatible f-transversal of C. It follows that $\bar{C} \in \text{SAT}$ and that for each $t \in M(C)$ holds $\varphi^{-1}(t) \in \mathcal{F}_{\text{comp}}(C)$. The vice versa assertion follows by exchanging the roles of C and $\bar{C}$. □

Corollary 2 *Let $C \in \text{Asym}$ such that also $\bar{C} \in \text{Asym}$ and $B(C) = B(\bar{C})$. Then $C \in \text{SAT}$ if and only if $\bar{C} \in \text{SAT}$.* □

A formula is satisfiable if and only if the complement formula admits a compatible f-transversal. Therefore the specific class of formulas C such that every f-transversal of C is compatible is of interest, because then any f-transversal gives rise to a model of $\bar{C}$, and vice versa. To provide a characterization of that very specific class, for $C \in \text{CNF}$, let $I(C) := \{r = b \cap b' : b \neq b' \in B(C)\}$, and let $C[r] = \{c \in C : r \subseteq V(c)\}$, for each $r \in I(C)$.

Lemma 5 *Let $C \in \text{CNF}$ such that $B(C) = B(\bar{C})$. Then $\mathcal{F}_{\text{comp}}(C) = \mathcal{F}(C)$, i.e., any f-transversal of C is compatible iff $(*)$: for each fixed $r \in I(C)$ holds $c|_r = c_0|_r, \forall c \in C[r]$ and arbitrary fixed $c_0 \in C[r]$.*

Proof. Consider the hypergraph $\tilde{B} := B(C) \cup I(C)$ having $V(C)$ as vertex set. Similarly consider $\tilde{C} := C \cup \{c \cap r : c \in C, r \in I(C)\}$ then it is easy to see that $(*)$ is equivalent to: For each $r \in I(C)$ holds $|\tilde{C}_r| = 1$ where $\tilde{C}_r := \{c \in \tilde{C} : V(c) = r\}$ is the fibre of $\tilde{C}$ over r, from which the assertion immediately follows. □

So we obtain a class of satisfiable formulas recognizable in polynomial time: Let CNF_{comp} denote the class of all formulas $C \in \text{CNF}$ with $B(C) = B(\bar{C})$, and such that $\mathcal{F}(\bar{C}) = \mathcal{F}_{\text{comp}}(\bar{C})$. As an example for $C \in \text{CNF}_{\text{comp}}$, let $\mathcal{H} = (V, B)$ with $V = \{q, r, s, t, u, v, x, y\}$, $B = \{b_1 = xy, b_2 = yuv, b_3 = vxr, b_4 = rst, b_5 = txq\}$ where brackets for edges are omitted, then the following (linear) formula is maximal w.r.t. to membership in CNF_{comp}, i.e., any additional clause over any $b \in B$ disturbs that membership, of course polarity of variables in $V(I(C))$ can be chosen differently:

$$\begin{array}{lllll} \bar{C} = x\bar{y} & \bar{y}uv & vxr & rs\bar{t} & \bar{t}xq \\ & \bar{y}\bar{u}v & & r\bar{s}\bar{t} & \bar{t}x\bar{q} \end{array}$$

arranged fibrewise. It is obvious that any f-transversal of $\bar{C}$ is compatible, hence $C = K_{\mathcal{H}} - \bar{C} \in \text{CNF}_{\text{comp}}$.

Theorem 4 *We can check in polynomial time whether an input formula $C \in \text{CNF}$ belongs to $\text{CNF}_{\text{comp}} \neq \varnothing$ and in positive case, implying that C is satisfiable, a model can be provided in polynomial time.*

Proof. Clearly, if $C \in \text{CNF}_{\text{comp}}$, then we only need to select a clause $c_b \in W_b - C_b$ for each $b \in B(C)$ ensuring that $\bigcup_{b \in B(C)} c_b^\gamma \in M(C)$ due to Theorem 1 (ii). For fixed $b = \{b_{i_1}, \ldots, b_{i_{|b|}}\}$, the selection can be performed e.g. by ordering the members $c = \{b_{i_1}^{\varepsilon_{i_1}(c)}, \ldots, b_{i_{|b|}}^{\varepsilon_{i_{|b|}}(c)}\}$ in C_b by lexicographic order of the vectors $(\varepsilon_{i_1}(c), \ldots, \varepsilon_{i_{|b|}}(c)) \in \{0, 1\}^{|b|}$.

To decide whether $C \in \mathrm{CNF}_{\mathrm{comp}}$, according to the discussion above, first compute $I(C)$ and $V(I(C))$ i.e., all variables occuring in members of $I(C)$. Then check whether each $x \in V(I(C))$ occurs in $\bar{C}$ with a fixed polarity, only in the positive case holds $C \in \mathrm{CNF}_{\mathrm{comp}}$. □

3 Some Aspects of Autarkies

The concept of autarkies has been introduced in [11]. Since then it has been used in automated theorem proving and was also subject of theoretical investigations. However there are many questions open concerning the structure of autarkies in a given CNF formula and also concerning their algorithmical usefulness. This section is devoted to provide some structural aspects regarding autark variable sets.

Given $C \in \mathrm{CNF}$ and $U \subset V(C)$, let C_U be the substructure of C, called the *U-retract* of C, defined by

$$C_U := \{c|_U : c \in C\}$$

So, C_U consists of the restrictions $c|_U$ of the clauses $c \in C$ to literals over U.

In contrast to $\hat{C}(U)$, C_U, in general, is no subformula of C. However, clearly, $C_U = \hat{C}(U)$ is a subformula if and only if $U = V(\hat{C}(U))$. Let us say a retract C_U is a *k-retract* if for each $c \in C_U$ holds $|c| \leq k$ and if there is at least one $c \in C_U$ with $|c| = k$.

Informally, an autark set of variables can be removed from a formula without affecting its satisfiability status, more precisely:

Definition 4 *For $C \in \mathrm{CNF}$, we call $U \subseteq V(C)$ an* autark set*, if there exists a (partial) truth assignment $\alpha : U \to \{0,1\}$ satisfying $\hat{C}(U)$; in which case α is called an* autarky*.*
A family $U_1, \ldots, U_r \subseteq V(C)$ of autark sets is called an autark cover *for C if $\bigcup_i \hat{C}(U_i) = C$. If in addition, for $k \in \mathbb{N}$, holds $|U_i| \leq k$, $1 \leq i \leq r$, we speak of a* k-autark cover *of C. Let* At *(resp.* At_k*) denote the set of all formulas for which an autark cover (resp. k-autark cover) exists.*

Clearly, for each $C \in \mathrm{At}$ with covering $U_1, \ldots, U_r$ there exists a smallest $k \geq 1$ such that $C \in \mathrm{At}_k$. By definition holds $\mathrm{At}_k \subset \mathrm{At}_j$ for each $j \geq k$. For the monotone fomula classes holds $\mathrm{CNF}_\epsilon \subset \mathrm{At}_1$ for $\epsilon \in \{+,-\}$.

The following assertions provide some basic properties of autarkies that are not hard to verify, so the proofs are omitted.

Lemma 6 *For $C \in \mathrm{CNF}$, let $\alpha_i : U_i \to \{0,1\}$, $i = 1,2$, be autarkies of C. Then $\alpha_1|\alpha_2 : U_1 \cup U_2 \to \{0,1\}$ and $\alpha_2|\alpha_1 : U_1 \cup U_2 \to \{0,1\}$ defined by*

$$\alpha_i|\alpha_j(x) := \begin{cases} \alpha_i(x); x \in U_i \\ \alpha_j(x); x \in U_j - U_i \end{cases}$$

where $i \neq j \in \{1,2\}$, also are autarkies of C. □

Lemma 7 *Let $C \in \text{CNF}$ and $U \subseteq V(C)$ be arbitrary. If $\alpha : W \to \{0,1\}$ is an autarky of C, then the restriction $\alpha|_{W \cap [C - \hat{C}(U)]}$ is an autarky of $C - \hat{C}(U)$.* □

Lemma 8 *For $C \in \text{At}_k$ and $U \subseteq V(C)$ holds $C - \hat{C}(U) \in \text{At}_k$.* □

Lemma 9 $\text{At}_k \subset \text{SAT}$ *for $k \geq 1$.* □

Lemma 10 *For $C \in \text{CNF}$, $U \subseteq V(C)$ is an autark set iff C_U is satisfiable.* □

Proposition 4 *Given $C \in \text{CNF}, k \in \mathbb{N}$, $k < |V(C)| =:$, in time $O(k\|C\|n^k T(k\text{-SAT}_k))$, one can test whether $C \in \text{At}_k$ and if, find a model for C, where $T(k\text{-SAT}_k)$ denotes the time for solving the k-SAT problem for an input instance of k variables.*

Proof. For each set $U \subset V(C)$ with $|U| \leq k$ compute the retract C_U in time $O(k \cdots \|C\|)$. Obviously, $C_U \in k\text{-CNF}$ and $|V(C_U)| = |U| \leq k$, hence we can check in time $T(k\text{-SAT}_k)$ whether $C_U \in \text{SAT}$. There are $O(n^k)$ subsets of $V(C)$ of at most k elements yielding the claim. □

Problem: The bound stated above is polynomial for fixed k but clearly yields no fixed-parameter tractability characterization [5] for SAT restricted to At_k with respect to parameter k. So, the question arises whether $\text{At}_k\text{-SAT} \in \text{FPT}$.

Below we intend to provide some structural features concerning autarkies which could be used to approach an answer to this question, which however is not given in this paper.

First regarding autarkies in linear formulas [13] a first simple observation is:

Lemma 11 *For linear $C \in \text{CNF}$ containing no unit clauses and no pure literals, a set $U \subset V(C)$ can be autark only if $|U| \geq 1 + \min\{|C(l)| : l \in L(C)\}$.*

Proof. Assume the assertion does not hold, and let $\alpha : U \to \{0,1\}$ be an autarky of C with $|U| < 1 + \min\{|C(l)| : l \in L(C)\}$. For convenience we regard each value $\alpha(x)$ as a literal, namely $\alpha(x) := x$ if x is assigned true and $\alpha(x) := \bar{x}$ otherwise. Since, by assumption, there is no pure literal in C we must have $|U| \geq 2$. For $x \in U$ all clauses in $C(\alpha(x))$ are satisfied. Since x is no pure literal each clause in $C(\overline{\alpha(x)}) \neq \emptyset$ still has to be satisfied by literals over variables in $U - \{x\}$. Because all these clauses have x in common we need $|C(\overline{\alpha(x)})| \geq 1 + \min\{|C(l)| : l \in L(C)\}$ further variables in U, but $|U| - 1 < \min\{|C(l)| : l \in L(C)\}$. Hence α cannot be an autarky yielding the lemma via contradiction. □

3.1 Autarky Closures

In this section we study which subsets of $V(C)$ need to be tested for autarky, for which a certain hull concept, defined next, turns out to be useful. Given $C \in \text{CNF}$, then for every $U \subseteq V(C)$ we define the set $H_C(U) \subseteq V(C)$ as

$$H_C(U) := \{x \in V(\hat{C}(U)) : x \notin V(C - \hat{C}(U))\}$$

Recall that $\hat{C}(U) = \{c \in C : V(c) \cap U \neq \varnothing\}$. Obviously, we also have:

$$(*) \quad H_C(U) = V(C) - V(C - \hat{C}(U))$$

We call $H_C(U)$ the *autarky closure or autarky hull* of U, which is justified due to the following observation.

Lemma 12 *For each fixed $C \in$ CNF, the map $H_C : 2^{V(C)} \to 2^{V(C)}$ as defined above is well formulated and is a (finite) closure operator.*

PROOF. Clearly each subset $U \subseteq V(C)$ yields the unique subformula $\hat{C}(U)$ of the fixed formula $C \in$ CNF. Moreover for every $x \in V(\hat{C}(U))$ either $x \in V(C - \hat{C}(U))$ or not, hence $H_C(U)$ is a unique subset of $V(C)$ hence H_C is a well formulated total map, where $H_C(\varnothing) = \varnothing$.
To verify that H_C is a closure operator, recall that a closure operator $\sigma : 2^M \to 2^M$ has the following defining properties: (i) $\forall S \subseteq M$ holds $S \subseteq \sigma(S)$, (ii) $\forall S_1, S_2 \subseteq M$ with $S_1 \subseteq S_2$ holds $\sigma(S_1) \subseteq \sigma(S_2)$, and (iii) $\forall S \subseteq M$ we have $\sigma(\sigma(S)) = \sigma(S)$.
(i) obviously holds true for H_C. Let $U_1, U_2 \subseteq V(C)$ with $U_1 \subseteq U_2$, then $\hat{C}(U_1) \subseteq \hat{C}(U_2)$, hence $V(C - \hat{C}(U_2)) \subseteq V(C - \hat{C}(U_1))$. Now suppose there is $x \in H_C(U_1)$ with $x \notin H_C(U_2)$, then by definition $x \in V(C - \hat{C}(U_2))$ and therefore $x \in V(C - \hat{C}(U_1))$ contradicting the assumption, thus (ii) holds. Finally, let $Q := H_C(U)$, for $U \in 2^{V(C)}$. We have $\hat{C}(Q) = \hat{C}(U)$ since no variable of Q occurs outside $\hat{C}(U)$, yielding $H_C(Q) = Q$ which is (iii). □

Lemma 13 *Given $C \in$ CNF.*
1.) For $U_1, U_2 \subset V(C)$, $U_1 \sim U_2 :\Leftrightarrow \hat{C}(U_1) = \hat{C}(U_2)$ defines an equivalence relation on $2^{V(C)}$ with classes $[U]$.
2.) The quotient space $2^{V(C)}/\sim$ is in 1:1-correspondence to $\{H_C(U) : U \in 2^{V(C)}\}$.

PROOF. The first part is obvious. For proving the second part we claim that for each $U_1, U_2 \in 2^{V(C)}$ holds

$$H_C(U_1) = H_C(U_2) \quad \Leftrightarrow \quad U_1 \sim U_2$$

from which 2.) obviously follows. Now $U_1 \sim U_2$ means $\hat{C}(U_1) = \hat{C}(U_2)$ implying $H_C(U_1) = H_C(U_2)$. For the reverse direction we observe that $\hat{C}(H_C(U)) = \hat{C}(U)$, for each $U \subseteq V(C)$. Therefore, $H_C(U_1) = H_C(U_2)$ implies $\hat{C}(U_1) = \hat{C}(H_C(U_1)) = \hat{C}(H_C(U_2)) = \hat{C}(U_2)$ thus $U_1 \sim U_2$. □

Lemma 14 *Let $C \in$ CNF and $U \subseteq V(C)$, then $H_C(U)$ is autark if U is autark.*

PROOF. Suppose U is autark. Since $\hat{C}(U) = \hat{C}(H_C(U))$ any truth assignment $\alpha : U \to \{0,1\}$ satisfying $\hat{C}(U)$ also satisfies $\hat{C}(H_C(U))$, thus $H_C(U)$ is an autark set, for any extension α^* of α to $H_C(U)$. □

Therefore, instead of checking all subsets of $V(C)$ for autarky, it suffices to check the hulls only. Indeed, there can be left no autark set because if U is autark then also $H_C(U)$ is. Supposing no autarky hull is autark, then there is no autark set at all, because otherwise its hull must have been checked positive for autarky.

Proposition 5 *Given* $C \in \mathrm{CNF}$*,* $k \in \mathbb{N}$*, for checking whether* C *has an autark set it suffices to test the members in* $\{H_C(U) : U \in 2^{V(C)}\}$*.* □

But this yields an improvement only in cases where $\{H_C(U) : U \in 2^{V(C)}\}$ can be enumerated without enumerating $2^{V(C)}$.

Given $C \in \mathrm{CNF}$ and $U = \{x_{i_1}, \ldots, x_{i_k}\} \subset V(C)$, for simplicity let $H(U) = H(x_{i_1}, \ldots, x_{i_k}) := H_C(U)$.

Lemma 15 *Let* $x, y \in V(C)$ *and* $U, Q \subset V(C)$ *then:*
(1) $H(x) = H(y)$ *iff for each* $c \in C$ *holds* $x \in V(c) \Leftrightarrow y \in V(c)$ *then* $H(x) = H(x, y) = H(y)$*.*
(2) $H(H(U) \cup H(Q)) = H(U \cup Q)$*.*
(3) If $H(U) = H(Q)$ *then* $H(U) = H(U \cup Q) = H(Q)$*.*

Proof. $\Leftarrow$ direction of the first equivalence in (1) is obvious. For the reverse direction assume $H(x) = H(y)$ from which follows $y \in H(x)$ and $x \in H(y)$ implying that for each $c \in C$ holds $x \in V(c) \Leftrightarrow y \in V(c)$. The latter is equivalent to $\hat{C}(x) = \hat{C}(x, y) = \hat{C}(y)$ implying $H(x) = H(x, y) = H(y)$ finishing the proof of (1).

(2) is an immediate consequence of closure operator properties of H: Since $U \cap Q \subseteq H(U) \cup H(Q)$ it follows $H(U \cup Q) \subseteq H(H(U) \cup H(Q))$. Similarly, because $H(U) \subseteq H(U \cup Q)$ and $H(Q) \subseteq H(U \cup Q)$ we have $H(U) \cup H(Q) \subseteq H(U \cup Q)$ therefore $H(H(x) \cup H(y)) \subseteq H(H(U \cup Q)) = H(U \cup Q)$ using idempotency of closures yielding (2).

With (2) holds $H(U \cup Q) = H(H(U) \cup H(Q)) = H(H(U)) = H(U)$, if $H(U) = H(Q)$, hence (3). □

We call an autarky hull in $C \in \mathrm{CNF}$ *free* if it does not contain a non-empty subset that is a hull, called *subhull* for short. A hull U is called *i-hull* if there is a smallest set $Q \subset V(C)$ with $|Q| = i$ such that $H(Q) = U$. Observe that often holds $|H(Q)| > i$.

Lemma 16 *A hull* $U \subset V(C)$ *in* C *is free iff* $H(x) = U$ *for each* $x \in U$*.*

Proof. For each $x \in U$ we have $H(x) \subseteq H(U)$, so if U is free we have $H(x) = H(U) = U$. Reversely, assume there is a non-empty hull $Q \subset U$ in C and $H(x) = U$ for each $x \in U$. Because $x \in Q$ implies $H(x) \subset H(W) = Q \neq U$ we have $H(x) \neq U$ yielding a contradiction. □

Because $H(x) = U$ for a free hull U and each $x \in U$ we have:

Lemma 17 *Each free hull in* $C \in \mathrm{CNF}$ *is a 1-hull. There exist at most* $|V(C)|$ *distinct free hulls in* C*.* □

Theorem 5 *For* $C \in \mathrm{CNF}$*, we can check in polynomial time which hulls in* C *are free or whether there is none. Moreover, a free hull* $U \subset V(C)$ *in* C *can be checked for autarky in linear time* $O(\|C\|)$*.*

Proof. Since each free hull is a 1-hull we focus on 1-hulls and first compute $H(x)$ for all $x \in V(C)$ which can be done in time $O(n \cdot \|C\|)$. Since $|H(x)| \in O(n)$

for each $x \in V(C)$ where $n := |V(C)|$, in $O(n^2)$ we can check whether a given hull contains one of the $n-1$ other hulls yielding an overall time of $O(n \cdot \|C\| + n^3)$.

For the second assertion it suffices to show that if U is a free hull then the retract C_U is a subset of the hypercube formula W_U. Indeed, in that case we have $C_U \notin \mathrm{SAT}$ if and only if $C_U = W_U$. So, testing a free hull for autarky simply means checking, whether $|C_U| = 2^{|U|}$ holds. Computing C_U from $\hat{C}(U)$ can be achieved by inspecting C due to appropriate data structures in linear time.

It remains to verify $C_U \subseteq W_U$ for which we claim that $U \subseteq V(c)$, for each $c \in \hat{C}(U)$. Suppose this does not hold, then there is $c \in \hat{C}(U)$ and $x \in U$ not contained in $V(c)$. Hence $c \notin \hat{C}(x)$ but $c \in \hat{C}(U)$, therefore $\hat{C}(x) \neq \hat{C}(U)$. On the other hand, as U is free we must have $H(y) = U$, for each $y \in U$, thus $\hat{C}(x) = \hat{C}(U)$ yielding a contradiction. □

Free subhulls in a hull are disjoint:

Lemma 18 *Let $C \in \mathrm{CNF}$ and U be a hull in C. If U_1, U_2 are two distinct free subhulls of U in C then $U_1 \cap U_2 = \varnothing$.*

Proof. Assume $S := U_1 \cap U_2 \neq \varnothing$ then $H(S) \subseteq H(U_i) = U_i$, $i = 1, 2$. Because $U_1 \neq U_2$ we have $H(S) \neq U_i, i = 1, 2$ contradicting that U_i are free for $i = 1, 2$. □

Lemma 19 *For fixed positive integer j, let $C \in \mathrm{CNF}$ be such that each $U \subset V(C)$ with $|U| \leq j$ is a hull in C. Then, for each $Q \subset V(C)$ with $|Q| \geq n - j$, the subformula $\hat{C}(Q')' = C - \hat{C}(Q')$ consists of* exactly *the variables in Q, where $Q' := V(C) - Q$ and $n = |V(C)|$. Moreover $\hat{C}(Q')' = C_Q$ iff $\hat{C}(Q)$ and $\hat{C}(Q')$ are independent components of C.*

Proof. Clearly, no $y \in V(Q')$ can occur in $\hat{C}(Q')'$. Moreover, for $Q \subset V(C)$ with $|Q| \geq n - j$ by assumption holds $H(Q') = Q'$ because then $|Q'| \leq j$. So, if there is $y \in Q$ not occuring in $\hat{C}(Q')'$ then y must occur in $\hat{C}(Q')$ hence $y \in Q'$ as Q' is a hull in C yielding a contradiction.

Now, subformula $\hat{C}(Q')'$ equals the retract C_Q if and only if $Q \cap V(\hat{C}(Q')) = \varnothing$ because $V(\hat{C}(Q')') = Q$ meaning $\hat{C}(Q')' = \hat{C}(Q)$ hence $V(\hat{C}(Q)) = Q$ and $V(\hat{C}(Q')') = Q'$ are distinct components of C. □

Lemma 20 *If U is an autarky closure in C, and $\mathcal{C}\ell(C)$ denotes the set of all autarky closures in C, then $\mathcal{C}\ell(C - \hat{C}(U)) = \{Q - U : Q \in \mathcal{C}\ell(C)\}$.*

Proof. The proof follows immediately from the fact that $V(C - \hat{C}(U)) = V(C) - U$, if U is a hull in C. □

Lemma 21 *Let U be a hull in $C \in \mathrm{CNF}$ that is not autark, and let Q be a hull in C containing U then Q also is not autark if*
(1) $(Q - U) \cap V(\hat{C}(U)) = \varnothing$ or
(2) $Q - U$ is not autark for $\hat{C}(Q) - \hat{C}(U)$.

PROOF. In case (1), no variable of $Q-U$ appears in $\hat{C}(U)$ thus $\hat{C}(U)$ cannot be satisfied by variables in W outside U. In case (2), let $\tilde{C} := \hat{C}(U)' = C - \hat{C}(U)$ then by Lemma 20, $Q' := Q - U$ is a hull in C'. Clearly $\hat{\tilde{C}}(Q) = \hat{C}(Q) - \hat{C}(U)$, thus if Q' is not autark then $\hat{\tilde{C}}(Q)$ cannot be satisfied over Q', hence there is no way for satisfying $\hat{C}(Q)$ over Q. □

Clearly, if $U \subset V(C)$ is autark for C and is a hull then there may be proper subsets Q of U which already are autark sets. In the specific case that U, in addition, is a free hull we even have $Q \in [U]$ for any $Q \subseteq U$, because $H(Q) = H(U)$. So we then ask for the complexity of deciding the autarky of a set $Q \in [U]$, where U is free. For $Q = U$ we are done by Theorem 5.

For each $Q \subseteq U$, we first claim that the retract C_Q is a subset of the hypercube formula W_Q. Again it suffices to show $V(c) = Q$, for each $c \in C_Q$. This can be done easily, as C_Q is obtained from C_U by taking $c \in C_U$ and removing all literals over variables in $U - Q$ yielding $c' \in C_Q$, thus $V(c') = V(c) \cap Q$. As argued in the proof of Theorem 5, we have $V(c) = U$, for each $c \in C_U$, so we get $V(c') = Q$, for each $c' \in C_Q$. As earlier this means that C_Q is satisfiable iff $|C_Q| \leq 2^{|Q|}$

Lemma 22 *Let U be a free hull in $C \in$ CNF that is autark. Then for each $i \in \{0, \ldots, j-1\}$ there are autark sets $Q \in [U]$ with $|Q| = |U| - i$ if and only if $|W_Q| - |C_U| = j$, for $j \in \mathbb{N}, 1 \leq j \leq 2^{|U|} - 1$.*

Due to the last result, we immediately obtain:

Corollary 3 *Given an integer k, there is an autark subset $Q \subset U$ of size at most k in an autark free hull U of C if and only if $|C_U| \leq 2^{|U|} - k - 1$.*

Assume there are hulls $Q, U, U \subset Q$ such that U is free and not autark, $Q' := Q - U$ is free and autark in $C' := C - \bar{C}(U)$ and $Q' \cap V(C_U) \neq \varnothing$, then the problem remains whether Q is autark in C according to Lemma 21.

Let $D_U := \{d^\gamma|_{Q' \cap V(C_U)} : d \in W_{Q'} - C'_{Q'}\} \subset W_{Q' \cap V(C_U)}$, and, for each $c \in C_U$, let $\pi^{-1}(c) := \{\tilde{c} \in \bar{C}(U) : \tilde{c}|_U = c\}$ be the fibre over c. Finally, let $\pi^{-1}_{Q'}(c) = \{\tilde{c}|_{Q'} : \tilde{c} \in \pi^{-1}(c)\}$ be the collection of fibre clauses over c restricted to the relevant part, since in general $V(C_U) \subseteq Q$ does not hold. Observe that whenever $c \in \pi^{-1}(c)$ meaning $c \in \bar{C}(U)$, then $c|'_Q \in \pi^{-1}_{Q'}(c)$ is the empty clause and hence $\pi^{-1}_{Q'}(c)$ is unsatisfiable.

Lemma 23 *Under circumstances and definitions mentioned above, Q is autark in C iff there is $d \in D_U$ and $c \in C_U$ such that d satisfies $\pi^{-1}_{Q'}(c)$, i.e., $d \cap c^\gamma_{Q'} \neq \varnothing$, for all $c_{Q'} \in \pi^{-1}_{Q'}(c)$.*

4 Hyperjoin Formulas

Next we define a class of formulas whose members are built by combining a finite number of hc formulas as follows:

Definition 5 *For $k \in \mathbb{N}$, $k \geq 2$, arbitrarily fixed, let $V_i, i \in [k]$, be sets of propositional variables with $V_i \cap V_j = \emptyset$, for distinct $i, j \in [k]$, and $V := \bigcup_{i \in [k]} V_i$. Let $W_i := W_{V_i}$ denote the hc formula over V_i. A* hyperjoin (formula) *$\mathfrak{H}$ over $\{W_i : i \in [k]\}$ is the* CNF *formula defined as follows: Each clause $c_i \in W_i$ either is a clause of $\mathfrak{H}$ or it is part of exactly one clause of $\mathfrak{H}$, for all $i \in [k]$ such that there is no clause in $\mathfrak{H}$ containing two clauses of the same W_i. In other words, to construct a $\mathfrak{H}$ out of $\{W_i : i \in [k]\}$ proceed as follows: Arbitrarily choose m_i clauses from W_i, where $0 \leq m_i \leq |W_i|$, for all $i \in [k]$, and by arbitrary unions compose new clauses, such that each chosen clause occurs in exactly one new clause, which are called the* joined *clauses in $\mathfrak{H}$, and such that each joined clause contains at most one member of W_i, for $i \in [k]$.*

For a hyperjoin $\mathfrak{H}$ over $\{W_i : i \in [k]\}$, we define for each W_i the joined part *$J_{\mathfrak{H}}(W_i)$ of W_i in $\mathfrak{H}$ by $J_{\mathfrak{H}}(W_i) := \{c \in W_i : c \notin \mathfrak{H}\}$.*

As example consider W_1, for $V_1 = \{x, y, z\}$, and W_2, for $V_2 = \{s, t, u, v, w\}$, let $D_1 = \{x\bar{y}\bar{z}, xy\bar{z}\}$, $D_2 = \{s\bar{t}\bar{u}v\bar{w}, \bar{s}\bar{t}\bar{u}\bar{v}\bar{w}\}$ and as hyperjoin over W_1, W_2 let

$$(W_{V_1} - D_1) \cup (W_{V_2} - D_2) \cup \{x\bar{y}\bar{z}s\bar{t}\bar{u}v\bar{w}, xy\bar{z}\bar{s}\bar{t}\bar{u}\bar{v}\bar{w}\}$$

with joined parts $J_{\mathfrak{H}}(W_i) = D_i$, $i = 1, 2$.

Clearly, a hyperjoin $\mathfrak{H}$ is unsatisfiable if there is $i \in [k]$ such that $J_{\mathfrak{H}}(W_i) = \emptyset$. So, from now on (except we explicitly state the contrary) we only consider hyperjoins such that $J_{\mathfrak{H}}(W_i) \neq \emptyset$, for all $i \in [k]$. Moreover, it is easy to see that such a hyperjoin $\mathfrak{H}$ always satisfies $\mathcal{H} := \mathcal{H}(\mathfrak{H}) = \mathcal{H}(\bar{\mathfrak{H}})$, i.e., both $\mathfrak{H}$ and its based complement formula are $\mathcal{H}$-formulas. However, observe that $\bar{\mathfrak{H}}$, in general, is no hyperjoin. It is not hard to see that a hyperjoin is always Sperner when regarded as a hypergraph over its literal set. Thus due to Theorem 2 we have:

Lemma 24 *Let $\mathfrak{H}$ be a hyperjoin over W_i, $i \in [k]$, such that its $\mathcal{H}(\mathfrak{H})$-based complement $\bar{\mathfrak{H}}$ also is a hyperjoin then $\mathfrak{H}$ and $\bar{\mathfrak{H}}$ are unsatisfiable if $\mathcal{H}(\mathfrak{H})$ is non-Sperner.*

From the general point of view, the relevant questions are:

(1) Given $C \in$ CNF, can efficiently (and constructively) be decided whether C is a hyperjoin, and in that can we efficiently reveal its *construction rule*, that is, can we determine the corresponding hc formulas $W_i, i \in [k]$, and its joined parts in C?
(2) If we know that a hyperjoin $\mathfrak{H}$ is constructed over W_i with $J_{\mathfrak{H}}(W_i)$, $i \in [k]$, can we then decide efficiently whether $\mathfrak{H}$ is satisfiable and can we find a model in the positive case?

Regarding (1), a hyperjoin $\mathfrak{H}$ behaves somewhat determined w.r.t. free autarky hulls in the following sense:

Lemma 25 *For a hyperjoin $\mathfrak{H}$ over W_i, $i \in [k]$, let $x \in V := \bigcup_{i \in [k]} V_i$ be arbitrary, then exactly one of the following holds:*

(1) $H(x)$ is a free hull, and either

(a) there is $i \in [k]$ such that $x \in V_i$ and $H(x) = H(V_i) = V_i$, then the retract $\mathfrak{H}_{H(x)} = W_i$, and $H(x)$ admits no autark assignment in $\mathfrak{H}$, or
(b) there is $s \in \mathbb{N}$, $I \subseteq [k]$ such that $x \in V_I := \bigcup_{i \in I} V_i$, $|V_i| = s$, for each $i \in [k]$, and $H(x) = H(V_I) = V_I$, then the retract $\mathfrak{H}_{H(x)}$ is a proper subset of W_{V_I}, and $H(x)$ admits an autark assignment in $\mathfrak{H}$.

Moreover, each free hull U in $\mathfrak{H}$ is of type (a) or (b), and is refered to, correspondingly.

(2) $H(x) = V_I$ is no free hull, where $I \subseteq [k]$, and $x \in V_I$, then there are $y_1, \ldots, y_r \in V_I$, with $r \leq |I|$, such that $H(y_i)$ is a free hull in $\mathfrak{H}_i$ of type (1) or (2), where $\mathfrak{H}_1 := \mathfrak{H}$, and $\mathfrak{H}_i := \mathfrak{H}_{i-1} - \mathfrak{H}_{H(y_{i-1})}$, are hyperjoins, for each $2 \leq i \leq r$.

Proof. Clearly, for each $x \in V$, $H(x)$ either is a free hull in $\mathfrak{H}$ or not. In the second case, $H(x)$ must contain a free hull U, which is disjoint to other free hulls, due to Lemma 18. Hence, assuming that (1) holds, it is not hard to see that removing the retract $\mathfrak{H}_U$ from $\mathfrak{H}$ yields a smaller hyperjoin that, inductively, can be treated analogously implying that (2) holds. So it remains to verify (1).

Suppose U is a free hull in $\mathfrak{H}$ then due to Lemma 16, we have $H(x) = U$, for each $x \in U$. Any clause of $\mathfrak{H}$ containing a variable $x \in V_i$, by construction, already contains the whole set V_i (disregarding negations). Therefore, for each hull U there exists a subset $I \subseteq [k]$ such that $U = V_I := \bigcup_{i \in I} V_i$. However, since U is free, we have the case $|I| = 1$ implying that U is of type (1), (a), and obviously the retract $\mathfrak{H}_{V_i} = W_i$ being unsatisfiable. The reverse direction is trivial. Or we have $|I| \geq 2$. Since U is free there cannot exist a clause in $\mathfrak{H}$ that does not contain all variables in V_I. So, each clause $c \in \mathfrak{H}$ such that $V_I \subseteq V(c)$ contains exactly one member of W_i, for all $i \in I$. It is not hard to see that this can happen if and only if $|V_i| = s$, for appropriate $s \in \mathbb{N}$. Moreover, then the retract $\mathfrak{H}_{V_I}$ is a proper subset of W_{V_I}, and therefore admits a model, cf. the proof of Theorem 5. □

Now, we can answer the first question stated above:

Theorem 6 *Given $C \in$ CNF in polynomial time we can check whether C is a hyperjoin, and in positive case reveal its construction rule, up to satisfiable substructures corresponding to free hulls in C of type (1), (b).*

Proof. Take any $x \in V$ and compute $H(x)$ in linear time, in polynomial time, due to Theorem 5 decide whether $H(x)$ is free. In the latter case check whether (1) (a) or (b) of Lemma 5 holds. If not, reject the input formula and stop.

Otherwise, $H(x)$ is no free hull in C, then repeat the last step for each variable in $H(x)$ distinct from x until either a rejection occured or a free hull is found. Then remove the corresponding retract from C, and continue with the remaining formula as described above until a rejection is found or we have a decomposition of free hull retracts of C each of type (1), (a), or (b), due to Lemma 25.

Finally, computing the joined parts between the determined retracts, can be performed in polynomial time via comparing the retracts with C. □

Addressing question (2) posed above, first observe that two clauses c_1, c_2 with the same base element $V(c_1)$. either belong to the same hc formula W_i or each is composed out of two or more clauses of the same hc formulas.

Lemma 26 *A hyperjoin $\mathfrak{H}$ over W_1, W_2 is unsatisfiable if and only if $|J_{\mathfrak{H}}(W_1)| = |J_{\mathfrak{H}}(W_2)| \in \{0, 1\}$.*

Proof. Observe that $r := |J_{\mathfrak{H}}(W_1)| = |J_{\mathfrak{H}}(W_2)|$, since otherwise a clause of a hc formula was part of more than one clause of $\mathfrak{H}$. The base hypergraph $\mathcal{H}(\mathfrak{H})$, has at most three edges, namely $B := B(\mathfrak{H}) := \{V_1, V_2, V := V_1 \cup V_2\}$. $\mathfrak{H}$ is unsatisfiable iff there is no compatible f-transversal of the complement formula $\bar{\mathfrak{H}} = \bigcup_{b\in B}(W_b - \mathfrak{H}_b)$ due to Theorem 1. First, we show that there is no compatible f-transversal of $\bar{\mathfrak{H}}$ if $r \in \{0, 1\}$. In case $r = 0$, the assertion certainly is true. If $r = 1$, let $c = c_1 \cup c_2 \in \mathfrak{H}$ where $c_i \in J_{\mathfrak{H}}(W_i)$, $i = 1, 2$. Hence, $\pi^{-1}_{\bar{\mathfrak{H}}}(V_i) = \{c_i\}$, $i = 1, 2$, and each f-transversal of $\bar{\mathfrak{H}}$ must contain c_1, c_2, as well as any element clause $c' \in \pi^{-1}_{\bar{\mathfrak{H}}}(V) - \{c\}$. But the only claus of W_V that can yield a compatible f-transversal together with c_1, c_2 obviously is $c \in \mathfrak{H}$ itself. Hence there is no compatible f-transversal of $\bar{\mathfrak{H}}$ at all.

For the reverse direction, assume $r \geq 2$, and let $c = c_1 \cup c_2, d = d_1 \cup d_2 \in \mathfrak{H}$ with $c_i, d_i \in W_i, i = 1, 2$. Now we claim that $F := \{c_1, d_2, c_1 \cup d_2\}$ is a compatible f-transversal of $\bar{\mathfrak{H}}$. Indeed, compatibility is obvious, because the union of the elements in F yields $c_1 \cup d_2 \in W_V$. It remains to verify that F indeed is a f-transversal of $\bar{\mathfrak{H}}$, obviously it is a f-transversal of the total clause set over B. Obviously, $c_1, d_2 \in \bar{\mathfrak{H}}$, because these are members of joined parts. Moreover, because each element of $J_{\mathfrak{H}}(W_i), i = 1, 2$, is used only once to yield a joined clause in $\mathfrak{H}$ there can be no other joined clauses containing c_1 or d_2. Thus $c_1 \cup d_2 \in \bar{\mathfrak{H}}$ and therefore $c_1^{\gamma} \cup d_2^{\gamma}$ is a model of $\mathfrak{H}$. □

Proposition 6 *A hyperjoin $\mathfrak{H}$ over W_i, $i \in [k]$, is satisfiable if and only if there is $\{c_i \in W_i : i \in [k]\}$ such that $c \not\subset c_1 \cup \cdots \cup c_k$, for each $c \in \mathfrak{H}$.*

Proof. Let $\mathcal{H} := \mathcal{H}(\mathfrak{H}) = (V, B)$ be the base hypergraph of $\mathfrak{H}$ where $V := \bigcup_{i\in[k]} V_i$. $\mathfrak{H}$ is satisfiable iff there exists a compatible f-transversal of the based complement formula $\bar{\mathfrak{H}}$ according to Theorem 1. We claim that there exists such a f-transversal iff there is $\{c_i \in W_i : i \in [k]\}$ such that $c \not\subset \bigcup_{i\in[k]} c_i$ for each $c \in \mathfrak{H}$. Clearly, each member $b \in B$ can be represented as a symmetric difference $b = \bigoplus_{i\in[k]} \alpha_i(b)V_i$, with $\alpha_i \in \{0, 1\}$, such that $\alpha_i V_i := \varnothing$ iff $\alpha_i = 0$ and $\alpha_i V_i := V_i$ iff $\alpha_i = 1$. By definition of $\mathfrak{H}$, every V_i, $i \in [k]$, occurs in the symmetric difference of exactly one $b \in B$. Each compatible f-transversal F of the total clause set $K_{\mathcal{H}}$ corresponds to a fixed set $\{c_i \in W_i : i \in [k]\}$ determined $F(b) = \bigoplus_{i\in[k]} \alpha_i(b)c_i, \alpha_i(b) \in \{0, 1\}$, for each $b \in B$. Indeed, let $\{F(b) : b \in B\}$ be a fixed compatible f-transversal of $K_{\mathcal{H}}$, then clearly, there can occur from each W_i $(i \in [k])$ at most one member in $\bigcup_{i\in[k]} F(b)$, otherwise F was not compatible. And since each V_i occurs in a base element it must be exactly one member of each W_i from which the claim follows immediately. So, there is a compatible f-transversal of $\bar{\mathfrak{H}} \subset K_{\mathcal{H}}$ if and only if there is a set $\{c_1, \ldots, c_k\}$ with the property of the Lemma completing the proof. □

Though the last result yields an equivalent characterization of satisfiability of hyperjoins, it yields no obvious polynomial time algorithm which is provided in the following from a slightly different point of view.

We call a formula C *connected* if the intersection graph of its base hypergraph $\mathcal{H}(C)$ is connected.

Proposition 7 *Let $\mathfrak{H}$ over W_i, $i \in [k]$, $k \geq 2$, be a connected hyperjoin. Then $\mathfrak{H}$ is satisfiable if there is a pair $(i,j) \in [k]^2$, $i \neq j$, such that for the retract $\mathfrak{H}(i,j) := \mathfrak{H}_{V_i \cup V_j}$ holds $|J_{\mathfrak{H}(i,j)}(W_i)| = |J_{\mathfrak{H}(i,j)}(W_j)| \geq 2$.*

Proof. If $k = 2$ we are done because then $\mathfrak{H}$ is satisfiable iff $|J_{\mathfrak{H}}(W_1)| = |J_{\mathfrak{H}}(W_2)| \geq 2$ due to Lemma 26. Let $k \geq 3$. Clearly, since $\mathfrak{H}$ is connected there must exist a permutation π of $\{1, 2, 3, \ldots, k\}$ such that the hyperjoin $\mathfrak{H}(r) := \mathfrak{H}(1, \ldots, r)$, i.e., the retract of $\mathfrak{H}$ constructed over the r first W_i w.r.t. π, is connected, for each fixed $2 \leq r \in k$. In other words, it then is possible to join the W_i's step by step until reaching $\mathfrak{H}$ such that each intermediate hyperjoin is connected.

Now let $V_i, V_j, i \neq j$, be such that $\mathfrak{H}(i,j)$ fulfills $|J_{\mathfrak{H}(i,j)}(W_i)| = |J_{\mathfrak{H}(i,j)}(W_j)| \geq 2$, and therefore is connected. Clearly, w.l.o.g. we can assume that $(i,j) = (1,2)$ and moreover that $\{1, \ldots, k\}$ is a permutation as described above. Again, for $r = 2$ we are done by Lemma 26 meaning $\mathfrak{H}(2) \in \mathrm{SAT}$. Clearly, because $\mathfrak{H}(3)$ by construction is connected, at least one member of W_3 is joined to a clause of $\mathfrak{H}(2)$, hence $J_{\mathfrak{H}(3)}(W_3) \neq \varnothing$. Let t be a model of $\mathfrak{H}(2)$, and let $c \in J_{\mathfrak{H}(3)}(W_3)$ be arbitrarily chosen then we claim that $t \cup c^{\gamma}$ is a model of $\mathfrak{H}(3)$. From which the desired implication of the theorem follows inductively. To verify the claim, first observe that t specifically already satisfies all clauses containing the members of $J_{\mathfrak{H}(3)}(W_3)$ because they only are enlarged. Hence taking $c \in J_{\mathfrak{H}(3)}(W_3)$ arbitrarily then c^{γ} satisfies all fragments of W_3 except for c but the corresponding joined clause in $\mathfrak{H}(3)$ is already satisfied, and the claim holds true. □

Note that the reverse implication of the assertion above does not hold. Clearly, if a hyperjoin is disconnected we can consider each component independently as decribed previously. However, from the last proof we immediately can deduce:

Corollary 4 *If there is a retract in a connected component of a hyperjoin that already is found to be satisfiable, then the whole component is satisfiable.* □

To a hyperjoin $\mathfrak{H}$ over W_i, $i \in [k]$, construct an edge.weighted graph $G(\mathfrak{H}), w$ as follows: For each W_i build a vertex x_i and two distinct vertices x_i, x_j are joined by an edge iff for the retract $\mathfrak{H}(i,j) := \mathfrak{H}_{V_i \cup V_j}$ holds $|J_{\mathfrak{H}(i,j)}(W_i)| = |J_{\mathfrak{H}(i,j)}(W_j)| > 0$, meaning that there exist $w \geq 1$ joined clauses in $\mathfrak{H}$ each containing a (distinct) member of W_i and of W_j, correspondingly. As weight assign to the edge the number w. Clearly, a hyperjoin $\mathfrak{H}$ is a connected formula iff $G(\mathfrak{H})$ is connected.

Proposition 8 *For $k \geq 2$, a connected hyperjoin $\mathfrak{H}$ over hc formulas $W_i := W_{V_i}$, $i \in [k]$, is satisfiable if and only if the sum of edge weights in $G(\mathfrak{H})$ is at least k.*

Proof. We proceed by induction on $k \geq 2$. The induction base is established due to Lemma 26. Now let the assertion hold, for each connected hyperjoin composed of at most k hc formulas, for a fixed $k \geq 2$, and assume that $\mathfrak{H}$ is connected and is composed over $k+1$ hc formulas; set $V := \bigcup_{i \in [k+1]} V_i$.

$\underline{\Rightarrow}$: So let $\mathfrak{H}$ be satisfiable, and suppose that the sum of edge weights in $G := G(\mathfrak{H})$ is at most k. Because of connectedness this implies that each edge weight is exactly one, so G has exactly k edges, and therefore is a tree because of connectedness. W.l.o.g. let W_{k+1} be a hc formula corresponding to a leaf in G having father vertex W_k then $\mathfrak{H}' := \mathfrak{H} - \mathfrak{H}_{V_{k+1}}$ is a hyperjoin over k hc formulas such that $G(\mathfrak{H}')$ is a tree having total edge weight $k-1$, and therefore, by induction hypothesis, $\mathfrak{H}'$ cannot be satisfiable. On the other hand, let t be a model of $\mathfrak{H}$, and let c corresponds to the unique edge of weight one in G joining W_{k+1} to W_k. Then clearly, $c = c_k \cup c_{k+1}$ where $c_k \in W_k$, $c_{k+1} \in W_{k+1}$ are fixed members, otherwise G would contain a cycle. Since W_k having at least two clauses is a leaf in G, at least one clause of W_{k+1} appears as a clause in $\mathfrak{H}$. Therefore, in order to satisfy these clauses, t has to contain the part c_{k+1}^{γ}. Hence, for satisfying c, t clearly has to satisfy c_k. It follows that restricting t to $V - V_{k+1}$ yields a model of $\mathfrak{H}'$ and therefore a contradiction.

$\underline{\Leftarrow}$: If $G(\mathfrak{H})$ which by assumption is connected has an edge of weight at least two, then $\mathfrak{H}$ is satisfiable due to Prop. 26. In the remaining case $G(\mathfrak{H})$ has at least $k+1$ edges and each edge has weight exactly one. Then it contains a cycle having vertices $W_{i_0}, \ldots, W_{i_{r-1}}$, for appropriate $r \leq k+1$, corresponding to the retract $\mathfrak{H}^r := \mathfrak{H}_{V(r)}$, where $V(r) := \bigcup_{j=0}^{r-1} V_{i_j}$. For all $j, l \in \{0, \ldots, r-1\}$, $j \neq l$, we then have $|J_{\mathfrak{H}^r(i_j,i_l)}(W_{i_j})| = |J_{\mathfrak{H}^r(i_j,i_l)}(W_{i_l})| = 1$ if $l = j+1 (\mod r)$ and 0 otherwise; let the corresponding joined clauses be $c_{i_0,i_1}, \ldots, c_{i_{r-1},i_0}$ meaning that $c_{i_j,i_{j+1}}$ contains a member $c_{i_j} \in W_{i_j}$ and a member $d_{i_{j+1}} \in W_{i_{j+1}} (\mod r)$, for $0 \leq j \leq r-1$. Now, it is not hard to see that a model for $\mathfrak{H}^I$ is provided by $\bigcup_{j=1}^{r-1} d_{i_j}^{\gamma} (\mod r)$. Therefore due to Cor. 4 also $\mathfrak{H}$ is satisfiable completing the proof. □

The last result immediately yields a polynomial time algorithm deciding SAT for hyperjoins and also to find a model, provided we know that the input formula is a hyperjoin and its construction rule which both can be determined in the sense of Theorem 6. So we are ready to state the main result of this section

Theorem 7 *Given a formula $C \in$ CNF, we can decide whether it is a hyperjoin in polynomial time. In the latter case we are able to decide satisfiability and in positive case to provide a model both in linear time.*

Proof. Suppose we have found in polynomial time, due to Theorem 6, that C is a hyperjoin. Then we know its construction rule up to free hulls of type (1), (b). Then we can build the graph $G(C)$ and compute its components in linear time. Clearly each component containing a detected free hull of type (1), (b), is satisfiable by Cor. 4, and a model for each such component is given by c^{γ}, where c is an arbitrary clause in the based-complement of the corresponding hull due to Theorem 1 that can be found appropriately scanning the hull in linear time. For each remaining component proceed due to Prop. 8 for deciding its satisfiability, and in positive case, a model can be provided in linear time depth-first traversing the graph and taking c^{γ} for determined fragments of hc formulas as executed in the proof of Prop. 7. □

References

1. B. Aspvall, M. R. Plass, and R. E. Tarjan, A linear-time algorithm for testing the truth of certain quantified Boolean formulas, Inform. Process. Lett. 8 (1979) 121-123.
2. C. Berge, Hypergraphs, North-Holland, Amsterdam, 1989.
3. E. Boros, Y. Crama, and P. L. Hammer, Polynomial time inference of all valid implications for Horn and related formulae, Annals of Math. Artif. Intellig. 1 (1990) 21-32.
4. E. Boros, P. L. Hammer, and X. Sun, Recognition of q-Horn formulae in linear time, Discrete Appl. Math. 55 (1994) 1-13.
5. R. G. Downey and M. R. Fellows, Parameterized Complexity, Springer-Verlag, New York, 1999.
6. J. Franco, A. v. Gelder, A perspective on certain polynomial-time solvable classes of satisfiability, Discrete Appl. Math. 125 (2003) 177-214.
7. H. Kleine Büning and T. Lettman, Propositional logic, deduction and algorithms, Cambridge University Press, Cambridge, 1999.
8. D. E. Knuth, Nested satisfiability, Acta Informatica 28 (1990) 1-6.
9. H. R. Lewis, Renaming a Set of Clauses as a Horn Set, J. ACM 25 (1978) 134-135.
10. M. Minoux, LTUR: A Simplified Linear-Time Unit Resolution Algorithm for Horn Formulae and Computer Implementation, Inform. Process. Lett. 29 (1988) 1-12.
11. B. Monien, and E. Speckenmeyer, Solving satisfiability in less than 2^n steps, Discrete Appl. Math. 10 (1985) 287-295.
12. S. Porschen, and E. Speckenmeyer, Worst case bounds for some NP-complete modified Horn-SAT problems, in: "Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing (SAT'04), Vancouver, British Columbia, Canada", Lect. Notes in Comp. Science, Vol. 3542, pp. 251-262, 2005.
13. S. Porschen, E. Speckenmeyer, and B. Randerath, On linear CNF formulas, in: "A. Biere, C. P. Gomes (Eds.), Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT 2006), Seattle, WA, USA", Lect. Notes in Comp. Science, Vol. 4121, pp. 221-225, Springer-Verlag, Berlin, 2006.
14. S. Porschen, E. Speckenmeyer, Linear CNF formulas and satisfiability, Techn. Report zaik2006-520, Univ. Köln, 2006.
15. C. A. Tovey, A Simplified NP-Complete Satisfiability Problem, Discrete Appl. Math. 8 (1984) 85-89.