SIAM JOURNAL ON COMPUTING, 24(6):1278–1304, 1995.

IDENTIFYING THE MINIMAL TRANSVERSALS OF A HYPERGRAPH AND RELATED PROBLEMS

THOMAS EITER† AND GEORG GOTTLOB†

Abstract. The paper considers two decision problems on hypergraphs, hypergraph saturation and recognition of the transversal hypergraph, and discusses their significance for several search problems in applied computer science. Hypergraph saturation, i.e., given a hypergraph $\mathcal{H}$, decide if every subset of vertices is contained in or contains some edge of $\mathcal{H}$, is shown to be co-**NP**-complete. A certain subproblem of hypergraph saturation, the saturation of simple hypergraphs (i.e. Sperner families), is shown to be under polynomial transformation equivalent to transversal hypergraph recognition, i.e., given two hypergraphs $\mathcal{H}_1, \mathcal{H}_2$, decide if the sets in $\mathcal{H}_2$ are all the minimal transversals of $\mathcal{H}_1$. The complexity of the search problem related to the recognition of the transversal hypergraph, the computation of the transversal hypergraph, is an open problem. This task needs time exponential in the input size; it is unknown whether an output-polynomial algorithm exists. For several important subcases, for instance if an upper or lower bound is imposed on the edge size or for acyclic hypergraphs, output-polynomial algorithms are presented. Computing or recognizing the minimal transversals of a hypergraph is a frequent problem in practice, which is pointed out by identifying important applications in database theory, Boolean switching theory, logic, and AI, particularly in model-based diagnosis.

Key words. hypergraphs, hypergraph transversals, hitting sets, independent sets, graph algorithms, NP-complete, polynomial time, output-polynomial time, database design, dependency inference, key computation, distributed databases, coteries, model-based diagnosis, satisfiability, prime implicants

AMS subject classifications. 05C65, 05C85, 05C90, 06E30, 68P15, 68Q20, 68Q25, 68R10, 68T30, 94C10

1. Introduction.

Hypergraph theory [7] is an important subfield of discrete mathematics with many relevant applications in both theoretical and applied computer science. In this paper, we study complexity issues of relevant computational problems on hypergraphs.

A hypergraph $\mathcal{H}$ is a set of subsets (edges) of a finite set of vertices. A hypergraph is simple if none of its edges is contained in any other of its edges. We say that a hypergraph is saturated if every subset of the vertex set is contained in an edge or contains an edge of the hypergraph.

The first problem we consider is the test whether a given hypergraph $\mathcal{H}$ is saturated. This decision problem will be referred to as HYPERGRAPH SATURATION (H-SAT). We will also deal with a restricted version of this problem where $\mathcal{H}$ is supposed to be a simple hypergraph. This subproblem is called SIMPLE HYPERGRAPH SATURATION (SIMPLE-H-SAT).

The second main problem – and probably the more important one from the application viewpoint – concerns hypergraph transversals. A transversal (or hitting set) of a hypergraph $\mathcal{H}$ is a subset of the vertices of $\mathcal{H}$ which intersects each edge of $\mathcal{H}$. A transversal is minimal if it does not properly contain any other transversal. The set $Tr(\mathcal{H})$ of all minimal transversals of a hypergraph $\mathcal{H}$ is itself a hypergraph called the transversal hypergraph of $\mathcal{H}$.

Our second problem can thus be formulated as follows: Given two hypergraphs $\mathcal{G}$ and $\mathcal{H}$, decide whether $\mathcal{G}$ is the transversal hypergraph of $\mathcal{H}$. This decision problem, referred to as TRANSVERSAL HYPERGRAPH (TRANS-HYP), is closely related to the search problem of computing $Tr(\mathcal{H})$ for a given hypergraph $\mathcal{H}$.

The complexity of TRANS-HYP is to date an open issue. The problem is in co-**NP**, but there is no proof of co-**NP**-completeness. On the other hand, while the problem is polynomial-time solvable if $\mathcal{G}$ or $\mathcal{H}$ is a graph, no polynomial algorithm for the solution of the general version of TRANS-HYP is currently known. Similarly, it is an open problem, whether

†Christian Doppler Laboratory for Expert Systems, Information Systems Department, Technical University of Vienna, Paniglgasse 16, A-1040 Wien, Austria.
Email: (eiter|gottlob)@vexpert.dbai.tuwien.ac.at

$Tr(\mathcal{H})$ can be computed in output-polynomial total time (i.e., in time polynomial in the combined sizes of the input and the output). This complexity problem was posed independently by several researchers [41, 14, 29]. Note that the existence of an output-polynomial algorithm for computing $Tr(\mathcal{H})$ would imply the polynomial-time solvability of TRANS-HYP; vice versa, if TRANS-HYP is co-**NP**-complete, then no output-polynomial algorithm for the computation of $Tr(\mathcal{H})$ is likely to exist.

This paper presents an extensive study of the above-mentioned complexity and computation problems. We derive some new complexity results and exhibit relationships to other relevant problems such as satisfiability or hypergraph 2-colorability. Particular attention is paid to restricted versions of the general problem classes which are either polynomial-time solvable or $\leq_m^p$-equivalent (i.e., equivalent under polynomial-time transformations) to TRANS-HYP. The considered problems have important applications, e.g., in database theory, switching theory, and artificial intelligence, from which it will become clear that computing the minimal transversals of a hypergraph is a frequently encountered problem of computer science. In the rest of this section, we outline the structure of the paper and highlight the most important results.

Our study of hypergraph problems starts in Section 2 with the definitions of basic hypergraph concepts and a brief overview of results known so far. In Section 3 we show that H-SAT is co-**NP**-complete and relate this problem to the well-known problem of hypergraph 2-colorability. SIMPLE-H-SAT turns out to be $\leq_m^p$-equivalent to the problem of 2-coloring a hypergraph whose edges are mutually intersecting. In Section 4 it is shown that SIMPLE-H-SAT is $\leq_m^p$-equivalent to TRANS-HYP. A restricted version of the latter problem is the question whether a given hypergraph is equal to its own transversal hypergraph, i.e., $Tr(\mathcal{H}) = \mathcal{H}$. This problem, which we call SELFTRANSVERSALITY, has attracted much interest by mathematicians [36, 6, 7] but no complexity results have been derived so far. We show that SELFTRANSVERSALITY is $\leq_m^p$-equivalent to TRANS-HYP.

Section 5 is dedicated to the investigation of polynomial subcases of the main problems. The most interesting results are briefly summarized as follows:

1. H-SAT becomes solvable in polynomial time if the cardinalities of the edges of the input hypergraph $\mathcal{H}$ differ by at most a constant k. In other words, if all edges of $\mathcal{H}$ have approximately the same size.

2. TRANS-HYP is decidable in polynomial time if the cardinalities of the edges of one of the input hypergraphs, say $\mathcal{H}$, are bounded by a constant. We present an output-polynomial algorithm for computing $Tr(\mathcal{H})$ in this case. Note that this result immediately leads to a relevant generalization of well-known output-polynomial methods for computing the maximal independent sets of graphs [52, 34, 29]. (Indeed, the maximal independent sets of a graph or hypergraph are exactly the complements of its minimal transversals.)

3. TRANS-HYP is decidable in polynomial time if one of the input hypergraphs is acyclic. We present an output-polynomial algorithm for computing $Tr(\mathcal{H})$ for acyclic hypergraphs $\mathcal{H}$. The type of acyclicity we consider is β-acyclicity as defined by Fagin in [16]. Note that β-acyclicity is among the weakest types of acyclicity; our results and methods also hold if we replace β-acyclicity by any of the several stronger types of acyclicity from the literature, cf. [16, 7].

Several applications where recognizing or computing the transversal hypergraph plays a fundamental role are described in Section 6. Our results can be profitably used in those applications. Section 7 concludes the paper; we state some open problems and give directions for further research. For convenient readability, a cross-reference list of definitions is provided in the appendix.

2. Preliminaries and previous results.

DEFINITION 2.1. *A hypergraph is a pair $(V, \mathcal{E})$ of a finite set $V = \{v_1, \ldots, v_n\}$ and a set $\mathcal{E}$ of subsets of V. The elements of V are called vertices, the elements of $\mathcal{E}$ edges.*

Note that some authors, e.g. [7], state that the edge-set as well as each edge must be nonempty and that the union of all edges yields the vertex set.

For notational convenience, we will identify a hypergraph with its edge-set and vice versa if there is no danger of ambiguity. Thus for hypergraphs $\mathcal{H} = (V, \mathcal{E})$ and $\mathcal{H}' = (V', \mathcal{E}')$, we write $E \in \mathcal{H}$ for $E \in \mathcal{E}$, $\mathcal{H} \cup \mathcal{H}'$ for $(V \cup V', \mathcal{E} \cup \mathcal{E}')$, $\min(\mathcal{H})$ for $(V, \min(\mathcal{E}))$ etc. Moreover, if not stated otherwise, it is presupposed that hypergraphs have the same set of vertices which is denoted by V, that n denotes the number of vertices, and that m denotes the number of edges.

A hypergraph $\mathcal{H}$ is called *simple* if for each pair H, H' of distinct edges of $\mathcal{H}$ it holds that $H \not\subseteq H'$. Simple hypergraphs are also called *Sperner families* according to [49], where it is proved that the cardinality $|\mathcal{H}|$ of a simple hypergraph $\mathcal{H}$ is bounded by $\binom{n}{[n/2]}$, which is asymptotic to $\left(\frac{2}{\pi}\right)^{1/2} 2^n n^{-1/2}$ [3]. For example, the set of minimal models for a finite set of propositional clauses or the minimal keys of a database relation form a Sperner system.

Let $\mathcal{H}$ be a hypergraph. The set $V_e(\mathcal{H}) = \bigcup_{H \in \mathcal{H}} H$ are the *essential vertices* of $\mathcal{H}$. The *rank* $r(\mathcal{H})$ of $\mathcal{H}$ is defined by $r(\mathcal{H}) = \max\{\, |E| : E \in \mathcal{H}\}$, and its *anti-rank* $ar(\mathcal{H})$ by $ar(\mathcal{H}) = \min\{|E| : E \in \mathcal{H}\}$. For example, every graph is a hypergraph of rank 2. $\mathcal{H}$ is called *intersecting* if no pair of its edges is disjoint, i.e., for all $E_1, E_2 \in \mathcal{H}$, $E_1 \cap E_2 \neq \emptyset$. Singleton edges in a hypergraph are referred to as *loops*.

EXAMPLE: Consider the hypergraph $\mathcal{H} = \{\{1\}, \{2,3\}, \{1,3,4\}\}$ on $V = \{1,2,3,4\}$. We note that $V_e(\mathcal{H}) = V$, i.e., all vertices are essential, and that $\mathcal{H}$ is not simple. Moreover, $ar(\mathcal{H}) = 1$, $r(\mathcal{H}) = 3$, and $\mathcal{H}$ is not intersecting.

DEFINITION 2.2. *Let $\mathcal{H}$ be a hypergraph. Then $\min(\mathcal{H})$ (resp. $\max(\mathcal{H})$) denotes the set of minimal (resp. maximal) edges of $\mathcal{H}$ with respect to set inclusion, i.e., $\min(\mathcal{H}) = \{E \in \mathcal{H} :$ there exists no $E' \in \mathcal{H}$ with $E' \subset E\}$ (resp. $\max(\mathcal{H}) = \{E \in \mathcal{H} :$ there exists no $E' \in \mathcal{H}$ with $E' \supset E\}$).*

Clearly, $\min(\mathcal{H})$ and $\max(\mathcal{H})$ are simple hypergraphs on V. For the hypergraph $\mathcal{H} = \{\{1\}, \{2,3\}, \{1,3,4\}\}$ of the last example, we have $\min(\mathcal{H}) = \{\{1\}, \{2,3\}\}$ and $\max(\mathcal{H}) = \{\{2,3\}, \{1,3,4\}\}$.

DEFINITION 2.3 (Transversal). *Let $\mathcal{H}$ be a hypergraph. A set $T \subseteq V$ is a transversal of $\mathcal{H}$ if for each $E \in \mathcal{H}$, $T \cap E \neq \emptyset$. A transversal T is minimal if no proper subset T' of T is a transversal.*

EXAMPLES: Consider the hypergraph $\mathcal{H} = \{\{1\}, \{2,3\}, \{1,3,4\}\}$ on $V = \{1,2,3,4\}$ again. $\mathcal{H}$ has two minimal transversals: $\{1,2\}$ and $\{1,3\}$. Note that $\emptyset$ is a minimal transversal of $\mathcal{H} = \emptyset$, since every $T \subseteq V$ vacuously fulfills the transversal criterion, and that $\mathcal{H} = \{\emptyset\}$ has no transversal, since no $T \subseteq V$ has nonempty intersection with $\emptyset$.

Transversals are also called *hitting sets* and in case of a graph *vertex covers*. The minimal vertex covers of a graph as well as the maximal independent sets are output-efficiently computable. An *independent set* of a graph $\mathcal{G}$ is a vertex subset $S \subseteq V$ that contains no edge of $\mathcal{G}$, and it is maximal if no proper superset of it has this property. Clearly, S is a maximal independent set of $\mathcal{G}$ iff $V - S$ is a minimal vertex cover of $\mathcal{G}$. Finding the lexicographically first maximal independent set is LOGSPACE-hard [11], while finding the lexicographically last maximal independent set is **NP**-hard [29]. (Given that V is ordered, $V_1 \subseteq V$ comes lexicographically before $V_2 \subseteq V$ if the first element at which V_1 and V_2 disagree is in V_1 [29].) The maximal independent sets of a graph can be output with polynomial-time delay [52, 34], even if they have to be in lexicographic order [29]. Moreover, fast parallel algorithms are

known, cf. [31].

Notice that transversals are closely related to edge covers. An *edge cover* of a hypergraph $\mathcal{H}$ is an edge subset $\mathcal{H}' \subseteq \mathcal{H}$ such that $\bigcup_{E \in \mathcal{H}'} E$ yields V, and is minimal if no proper subset is an edge cover. The computation of minimal transversals and minimal edge covers is easily transformable into each other, cf. [35], thus results on one of these problems apply to the other.

Finding a minimal transversal of a hypergraph $\mathcal{H}$ is efficiently possible: If $\mathcal{H} = \emptyset$, then $\emptyset$ is the only minimal transversal of $\mathcal{H}$. If $\mathcal{H} \neq \emptyset$, then $V = \{v_1, \ldots, v_n\}$ is a transversal of $\mathcal{H}$. If we define

$$T_0 = V, \quad \text{and for all } i,\ 1 \leq i \leq n,$$
$$T_i = \begin{cases} T_{i-1}, & \text{if } T_{i-1} - \{v_i\} \text{ is not a transversal;} \\ T_{i-1} - \{v_i\}, & \text{if } T_{i-1} - \{v_i\} \text{ is a transversal,} \end{cases}$$

then T_n is a minimal transversal of $\mathcal{H}$. Thus a minimal transversal of $\mathcal{H}$ can be found in time $O(m \cdot n)$. Finding a minimum cardinality transversal, however, is **NP**-hard, cf. [21].

DEFINITION 2.4. *Let $\mathcal{H}$ be a hypergraph on V. The transversal hypergraph $Tr(\mathcal{H})$ of $\mathcal{H}$ is the hypergraph on V whose edges are the minimal transversals of $\mathcal{H}$.*

The following propositions capture important relations between a hypergraph $\mathcal{H}$ and $Tr(\mathcal{H})$, cf. [7].

PROPOSITION 2.1. *For every hypergraph $\mathcal{H}$, $Tr(\mathcal{H})$ is a simple hypergraph and $Tr(\mathcal{H})$ = $Tr(\min(\mathcal{H}))$.*

PROPOSITION 2.2. *Let $\mathcal{G}$ and $\mathcal{H}$ be simple hypergraphs. Then,*

(i) *$\mathcal{G}$ = $Tr(\mathcal{H})$ if and only if $\mathcal{H}$ = $Tr(\mathcal{G})$;*
(ii) *$Tr(\mathcal{G}) = Tr(\mathcal{H})$ iff $\mathcal{G} = \mathcal{H}$;*
(iii) *$Tr(Tr(\mathcal{H})) = \mathcal{H}$.*

The following simple algorithm to determine $Tr(\mathcal{H})$ is given in [7] (cf. also [38, 35]). Let $\mathcal{H}$ = $\{E_1, \ldots, E_m\}$, $m \geq 0$. Note that $Tr(\{E_i\}) = \{\{e\} : e \in E_i\}$. Define a sequence $\mathcal{T}_0, \ldots, \mathcal{T}_m$ by

$$\mathcal{T}_0 = \{\emptyset\}, \quad \text{and for all } i,\ 1 \leq i \leq m, \quad \mathcal{T}_i = \min\{T \cup \{e\} : T \in \mathcal{T}_{i-1} \text{ and } e \in E_i\}.$$

Then, $\mathcal{T}_j = Tr(\{E_1, \ldots, E_j\})$, $0 \leq j \leq m$, and thus $\mathcal{T}_m = Tr(\mathcal{H})$.

It is easy to see that the computation of the transversal hypergraph is inherently exponential. For example, consider the hypergraph $\mathcal{F}_n = \{\{2i-1, 2i\} : 1 \leq i \leq n\}$ on $V_n = \{1, \ldots, 2n\}$. Then, $Tr(\mathcal{F}_n) = \{\{x_1, \ldots, x_n\} : x_i \in \{2i, 2i-1\}, 1 \leq i \leq n\}$. Since $Tr(\mathcal{F}_n)$ has 2^n edges, the computation of $Tr(\mathcal{F}_n)$ needs space (hence also time) exponential in the input size.

Since an algorithm for computing $Tr(\mathcal{H})$ that runs in time polynomial in the input size IS is not possible, an algorithm is desirable which works at least in polynomial time if the number of minimal transversals (resp. the output size OS) is taken into account. There are various possibilities to define appealing notions of output-polynomiality, e.g. output-polynomial total time, incremental-polynomial time, polynomial-time delay computability [29], or P-enumerability [54].

The most general of these concepts is *output-polynomial total time*, which requires that an algorithm works in time bounded by a polynomial in IS *and* OS. Note that an output-polynomial total time algorithm may run for exponentially many steps in IS without producing any output. The same holds for the notion of *P-enumerability*, where a search algorithm computes all solutions of a problem in time $p(IS)N$, where p is a polynomial and N is the number of solutions [54]. To consider incremental computations, an algorithm meets the

incremental-polynomial time criterion if it outputs all solutions $s_1, \ldots, s_N$ to the respective problem one by one as follows. The time until the first output and the time between output of solutions s_i, s_{i+1}, $1 \leq i < N$, is bounded by a polynomial in the combined sizes of the input and all solutions $s_1, \ldots, s_i$, and the algorithm stops in polynomial time after the output of the last solution. A more restrictive criterion is output with *polynomial-time delay*. An algorithm with this property generates the solutions one by one such that the time until the first solution is output and between the output of consecutive solutions is bounded by a polynomial in IS. Clearly, any such algorithm P-enumerates the solutions.

It is easy to show that the above algorithm does not stop in output-polynomial total time. Consider the complete graph $\mathcal{K}_n$ on vertices $\{1, \ldots, n\}$, i.e., $\mathcal{K}_n = \{\, \{i, j\} : 1 \leq i < j \leq n \,\}$, where $n \geq 4$ is even. Let the edges $E_1, \ldots, E_m$, $m = |\mathcal{K}_n|$, of $\mathcal{K}_n$ be ordered such that $E_1 = \{1, 2\}, E_2 = \{3, 4\}, \ldots, E_{n/2} = \{n-1, n\}$. Computing $Tr(\mathcal{K}_n)$ by $\mathcal{T}_0, \ldots, \mathcal{T}_m$, we find that $|\mathcal{T}_{n/2}| = 2^{n/2}$, because $\mathcal{T}_{n/2} = Tr(\{E_1, \ldots, E_{n/2}\})$ and $\{E_1, \ldots, E_{n/2}\}$ is just the hypergraph $\mathcal{F}_{n/2}$ from above. As $Tr(\mathcal{K}_n)$ is given by $\mathcal{T}_m = \{\{1, \ldots, n\} - \{i\} : 1 \leq i \leq n\}$ and $m = n(n-1)/2$, we have $n = |Tr(\mathcal{K}_n)| = \Theta(|\mathcal{K}_n|^{1/2})$. Relating $|\mathcal{T}_{n/2}|$ to the j-th power, j arbitrarily fixed, of $|\mathcal{K}_n| = \max\{\, |\mathcal{K}_n|, \, |Tr(\mathcal{K}_n)| \,\}$ we get

$$|\mathcal{T}_{n/2}| = 2^{\Theta(|\mathcal{K}_n|^{1/2})}, \quad \text{thus} \quad |\mathcal{T}_{n/2}| > |\mathcal{K}_n|^j$$

for sufficiently large n. Hence, the size of $\mathcal{T}_{n/2}$ is not bounded by any polynomial in the input and output size. Consequently, the above algorithm is not efficient with respect to output-polynomiality.

Though there are several algorithms which involve, in a more or less obvious way, the computation of transversal hypergraphs, (e.g., [38, 35, 46, 14, 40, 42, 45]), unfortunately no output-polynomial transversal hypergraph algorithm is known today; moreover, it is even an open question whether such an algorithm is likely to exist at all, cf. [14, 29, 43]. We will show in Section 4 that the complexity of the closely related decision problem TRANS-HYP, which is, given two hypergraphs $\mathcal{G}$ and $\mathcal{H}$, to decide if $\mathcal{G} = Tr(\mathcal{H})$, is of crucial importance to answer this question.

The second kind of problem in our study is the complexity of hypergraph saturation, a covering problem for the powerset $\mathcal{P}(S)$ of a finite set S considered in [51, 10]. We introduce some notation first.

DEFINITION 2.5. *Let S be a finite set, and let $X \subseteq S$. Then $Cov_S(X) = \{Y \subseteq S : Y \subseteq X \text{ or } Y \supseteq X\}$. For any hypergraph $\mathcal{H}$ on V, $Cov(\mathcal{H}) = \bigcup_{H \in \mathcal{H}} Cov_V(H)$.*

In words, $Cov(\mathcal{H})$ is the set of sets $X \subseteq V$ that are *covered* by an edge E of $\mathcal{H}$ *as subset* ($X \subseteq E$) or *as superset* ($X \supseteq E$).

DEFINITION 2.6. *A hypergraph $\mathcal{H}$ on V is called saturated if and only if $Cov(\mathcal{H}) = \mathcal{P}(V)$.*

According to the definition, a hypergraph $\mathcal{H}$ is saturated iff every set $V' \subseteq V$ is contained in at least one of edge of $\mathcal{H}$ or contains at least one edge of $\mathcal{H}$.

EXAMPLE: Consider the hypergraph $\mathcal{H} = \{\{1\}, \{2, 3\}, \{1, 3, 4\}\}$ on $V = \{1, 2, 3, 4\}$ again. Then, $Cov(\mathcal{H}) = \mathcal{P}(V) - \{2, 4\}$, i.e., $\mathcal{H}$ covers all subsets of V but $\{2, 4\}$. Hence, $\mathcal{H}$ is not saturated. If we add $\{2, 4\}$ to $\mathcal{H}$, the resulting hypergraph $\mathcal{H}' = \{\{1\}, \{2, 3\}, \{1, 3, 4\}, \{2, 4\}\}$ is saturated. Note that saturation of a hypergraph $\mathcal{G}$ does not imply saturation of either $\min(\mathcal{G})$ or $\max(\mathcal{G})$. As an example consider $\mathcal{H}'$. Neither $\min(\mathcal{H}') = \{\{1\}, \{2, 3\}, \{2, 4\}\}$ nor $\max(\mathcal{H}') = \{\{2, 3\}, \{1, 3, 4\}, \{2, 4\}\}$ is saturated, because $\{3, 4\}$ and $\{1, 2\}$ respectively are not covered.

The complexity of checking if a hypergraph is saturated has, to our best knowledge, not been studied yet in the literature.

We assume that the reader is familiar with the basic concepts of the theory of **NP**-completeness (cf. [21]). co-Π denotes the problem complementary to decision problem Π; $\leq_m^p$ denotes polynomial-time transformability. Decision problems Π and Π' are called $\leq_m^p$-*equivalent* iff $\Pi \leq_m^p \Pi'$ and $\Pi' \leq_m^p \Pi$. Problems that are solvable in polynomial time are referred to as *tractable*, and those which are most likely not solvable in polynomial time, amongst them all **NP**-hard problems, as *intractable*.

3. Hypergraph saturation. Imagine you are a friend of Toni who manages a pizza restaurant that is famous for its rich variety of pizze.[1]

The pizze differ in their topping, and accordingly every pizza has its special name. For example, on a *Pizza Margherita* there are tomatoes, onions, and some cheese, on a *Pizza al Tonno* tuna fish and onions, and on a *Pizza Provinciale* tomatoes, ham, corn, onions, mushrooms, salami, pepperonies, and cheese. Now Toni wants to enlarge his pizza offer by a new pizza, but the new pizza should be neither the "Grande" version of another pizza, i.e., have all the food of another pizza on it and some additional, nor the "Mini" version, i.e., all the food on it is also on some other pizza. Toni has been testing quite a number of food collections for the new pizza yet, but each of his compositions turned out to be a Grande or a Mini version of one of the many pizze. This makes Toni wondering if there is any food collection for a new pizza at all. Since you are an expert in computer science, he asks you to write a computer program for answering this question. Under the principle of *quot capita tot gustus*, which means that every collection of food is appropriate for a pizza, you find that Toni's problem is just the hypergraph saturation problem if the food available is considered as set V of vertices and the pizze already offered, described by food collections, are considered as edges of a hypergraph. In a more precise formulation,

Problem: HYPERGRAPH SATURATION (H-SAT)
Instance: A hypergraph $\mathcal{H} = (V, \mathcal{E})$ on vertices $V = \{v_1, \ldots, v_n\}$.
Question: Is $\mathcal{H}$ saturated ?

A simple brute force algorithm would be to test all possible 2^n vertex sets subsequently until some collection is found that is not covered by $\mathcal{H}$; this collection would be a new pizza which satisfies the requirements. However, if $\mathcal{H}$ is saturated, the algorithm recognizes this only after 2^n covering checks – too many if the number of food items on Toni's pizze is taken into account, as then the algorithm would take weeks to run. Thus a more subtle one than the naive exponential algorithm is needed to solve the problem efficiently. However, chances are very low that there is some algorithm substantially faster than the brute force algorithm, because the problem is co-**NP**-complete; this is shown in this section. First, we need some additional definitions.

DEFINITION 3.1. *Let V be a set and let $V' \subseteq V$. Then $\overline{V'} = V - V'$. For every hypergraph $\mathcal{H}$ on V, the complemented hypergraph of $\mathcal{H}$ is the hypergraph $\overline{\mathcal{H}} = \{\overline{H} : H \in \mathcal{H}\}$ on V.*

The following relationships between $\mathcal{H}$ and $\overline{\mathcal{H}}$ are straightforward, but nevertheless important:

PROPOSITION 3.1. *Let $\mathcal{H}$ be a hypergraph. Then $\overline{\overline{\mathcal{H}}} = \mathcal{H}$, and $\mathcal{H}$ is simple iff $\overline{\mathcal{H}}$ is simple.*

DEFINITION 3.2. *A hypergraph $\mathcal{H}$ is called self-complemented if and only if $\mathcal{H} = \overline{\mathcal{H}}$.*

Notice that if a hypergraph $\mathcal{H}$ is self-complemented, then for each $X \subseteq V$ we have $X \in Cov(\mathcal{H})$ *iff* $\overline{X} \in Cov(\mathcal{H})$.

2-colorability of a hypergraph, also known as set splitting, is defined as follows.

[1] Problems on pizze are studied elsewhere, cf. [24, p. 4].

DEFINITION 3.3. *Let $\mathcal{H}$ be a hypergraph. A partitioning (A, B) of V, i.e., $A \cup B = V$ and $A \cap B = \emptyset$, is a 2-coloring of $\mathcal{H}$ iff for every $E \in \mathcal{H}$, it holds that $E \not\subseteq A$ and $E \not\subseteq B$. $\mathcal{H}$ is 2-colorable iff there exists a 2-coloring for $\mathcal{H}$.*

In words, a hypergraph is 2-colorable if and only if there is an assignment of one of 2 "colors" to each vertex such that each edge has two colors. Deciding whether a hypergraph $\mathcal{H}$ is 2-colorable (HYPERGRAPH 2-COLORABILITY, HP2C) is an **NP**-complete problem [36, 21]. We prove that this problem remains **NP**-complete for the following restriction:

THEOREM 3.2. HP2C *remains* **NP***-complete even if the hypergraph $\mathcal{H}$ is self-complemented.*

Proof. Consider the hypergraph $\mathcal{H}'$ =$(V', \mathcal{E}')$, where $V' = V \cup \{e, f\}$ for some $e, f \notin V$, and $\mathcal{E}' = \mathcal{H} \cup \{\overline{E} \cup \{e, f\} : E \in \mathcal{H}\}$. It is clear that $\mathcal{H}'$ can be constructed from $\mathcal{H}$ in polynomial time; moreover, $\mathcal{H}'$ is self-complemented. We claim that $\mathcal{H}$ is 2-colorable if and only if $\mathcal{H}'$ is 2-colorable. To prove the *only if* direction, let $(A, \overline{A})$ be any 2-coloring of $\mathcal{H}$. If e,f are colored differently, only an edge of $\mathcal{H}'$ that is also an edge of $\mathcal{H}$ can prevent a 2-coloring of $\mathcal{H}'$. Hence $(A \cup \{e\},\ (V - A) \cup \{f\})$ is a 2-coloring of $\mathcal{H}'$. For the *if* direction, if $(B, \overline{B})$ is any 2-coloring of $\mathcal{H}'$, it does not color any edge from $\mathcal{H}$ monochromatically because $\mathcal{H} \subseteq \mathcal{H}'$, and since no edge of $\mathcal{H}$ contains e or f, clearly $(B - \{e, f\},\ \overline{B} - \{e, f\})$ is a 2-coloring of $\mathcal{H}$. □

Using the relationship between 2-colorable hypergraphs and saturated hypergraphs, we show now the intractability of H-SAT. We refer to the following lemma.

LEMMA 3.3. *Let $\mathcal{H}$ be a self-complemented hypergraph. Then $\mathcal{H}$ is not 2-colorable iff $\mathcal{H}$ is saturated.*

Proof. If $\mathcal{H}$ is not 2-colorable, for every partitioning $(A, \overline{A})$ of V there exists $H \in \mathcal{H}$ such that $H \subseteq A$ or $H \subseteq \overline{A}$. As self-complementarity of $\mathcal{H}$ implies that $A \in Cov(\mathcal{H})$ iff $\overline{A} \in Cov(\mathcal{H})$, it follows that $A, \overline{A} \in Cov(\mathcal{H})$; consequently, $\mathcal{H}$ is saturated, which proves the *only if* direction. For the *if* direction, assume $\mathcal{H}$ is 2-colored by $(A, \overline{A})$. We show that for every $H \in \mathcal{H}$, it holds that $A \not\supseteq H$ and $A \not\subseteq H$; hence, $A \notin Cov(\mathcal{H})$, which means that $\mathcal{H}$ is not saturated. Consider $H \in \mathcal{H}$. Since $(A, \overline{A})$ is a 2-coloring of $\mathcal{H}$, we have $A \not\supseteq H$. If $A \subseteq H$, then $\overline{A} \supseteq \overline{H}$, which by self-complementarity of $\mathcal{H}$ contradicts that $(A, \overline{A})$ is a 2-coloring of $\mathcal{H}$. Thus $A \not\subseteq H$. This proves the *if* direction. □

THEOREM 3.4. H-SAT *is co-***NP***-complete, even if $\mathcal{H}$ is self-complemented.*

Proof. Membership in co-**NP** is easy to show: Guess a subset $V' \subseteq V$ and check if $V' \in Cov(\mathcal{H})$ holds. This can clearly be done in time polynomial in n and m. co-**NP**-hardness under the asserted restriction follows immediately from Theorem 3.2 and Lemma 3.3. □

This result can be strengthened as follows.

THEOREM 3.5. H-SAT *remains co-***NP***-complete even if $\mathcal{H}$ is self-complemented and all edges have size k or $n - k$, for fixed $k \geq 3$.*

Proof. HP2C remains **NP**-complete even if no edge of $\mathcal{H}$ contains more than three vertices [21]. Applying the transformation in the proof of Theorem 3.2 on such a hypergraph $\mathcal{H}$, we obtain a self-complemented hypergraph $\mathcal{H}' = (V', \mathcal{E}')$ such that every $E \in \mathcal{H}'$ contains at most 3 vertices or at least $n' - 3$ vertices, where $n' = |V'| = n + 2$, with the property that $\mathcal{H}'$ is saturated iff $\mathcal{H}$ is not 2-colorable. Hence, H-SAT remains co-**NP**-complete if $\mathcal{H}$ has the form of $\mathcal{H}'$. Therefore, we assume in the following without loss of generality that $\mathcal{H}$ is self-complemented and that every edge contains at most 3 vertices or at least $n - 3$ vertices.

We show first the result for $k = 3$. Without loss of generality assume that $n > 2k = 6$. One can test in polynomial time if for each $X \subseteq V$ with $|X| \leq 3$, $X \in Cov(\mathcal{H})$ holds. If $\mathcal{H}$ does not cover all the sets X, then $\mathcal{H}$ is not saturated. Otherwise, replace in $\mathcal{H}$ every edge E with $|E| < 3$ by all sets E' of size 3 such that $E \subset E' \subseteq V$, and every edge F with $|F| > n - 3$ by all subsets $F' \subset F$ of size $n - 3$. Let $\mathcal{H}''$ be the resulting hypergraph. Notice

that $\mathcal{H}''$ is self-complemented and contains only edges of size 3 or $n-3$; furthermore, it follows from the construction of $\mathcal{H}''$ that $X \in Cov(\mathcal{H}'')$ for each $X \subseteq V$ such that $|X| \leq 3$.

We claim that $\mathcal{H}''$ is saturated iff $\mathcal{H}$ saturated. To show this, assume that $\mathcal{H}$ is not saturated. Then, since $\mathcal{H}$ is self-complemented, there exists $X \subseteq V$ such that $3 < |X| < n-3$ and $X \notin Cov(\mathcal{H})$. By construction of $\mathcal{H}''$, it follows that $X \notin Cov(\mathcal{H}'')$. Thus, $\mathcal{H}''$ is not saturated. The proof of the converse direction is analogous.

Since $\mathcal{H}''$ can be constructed in polynomial time, H-SAT remains co-**NP**-complete under the asserted restriction for $k = 3$.

For $k > 3$, the proof is analogous, where in each place "3" is replaced by "k". Notice since k is a constant, it can be checked in polynomial time if for each $X \subseteq V$ with $|X| \leq k$, it holds that $X \in Cov(\mathcal{H})$ (there are fewer than kn^k such X), and $\mathcal{H}''$ can be constructed in polynomial time. □

The following result is an immediate consequence of Theorem 3.5 and Lemma 3.3.

COROLLARY 3.6. HP2C *is* **NP***-complete even if $\mathcal{H}$ is self-complemented and all edges have size k or $n-k$, for fixed $k \geq 3$.*

The case $k = 3$ in Theorem 3.5 marks the intractability frontier, as the case $k = 2$ is polynomial.

THEOREM 3.7. H-SAT *is polynomial if $\mathcal{H}$ is self-complemented and all edges have size* 2 *or* $n-2$.

Proof. Let $\mathcal{G}$ be the graph on V with edges $\{E \in \mathcal{H} : |E| = 2\}$. We reduce deciding saturation of $\mathcal{H}$ to testing certain properties of $\mathcal{G}$, which can be done in polynomial time. In what follows, we assume without loss of generality that $\mathcal{G}$ is not empty (i.e., $\mathcal{H}$ is not empty) and that $n \geq 4$.

CLAIM A: *Let $v \in V$ be arbitrary. If v belongs to all edges of $\mathcal{G}$, then $\mathcal{H}$ is not saturated iff $V_e(\mathcal{G}) \subset V$.*

To prove the *if* direction of this claim, assume that $V_e(\mathcal{G}) \subset V$. Let $v' \in V - V_e(\mathcal{G})$ be arbitrary. It is readily checked that the partitioning $(\{v, v'\}, V - \{v, v'\})$ is a 2-coloring of $\mathcal{H}$. Thus, by Lemma 3.3, $\mathcal{H}$ is not saturated. This proves the *if* direction. For the *only if* direction, assume that $\mathcal{H}$ is not saturated. Then, by Lemma 3.3 there must exist a 2-coloring of $\mathcal{H}$; let $(B, \overline{B})$ be an arbitrary such 2-coloring. Since $\mathcal{G} \subseteq \mathcal{H}$, $(B, \overline{B})$ is a 2-coloring of $\mathcal{G}$. Assume that $V_e(\mathcal{G}) = V$. Then, it follows that $B = \{v\}$ or $\overline{B} = \{v\}$; assume without loss of generality that $B = \{v\}$. Let $E \in \mathcal{G}$ be arbitrary. Since $v \in E$, it follows $\overline{E} \subseteq V - \{v\} = \overline{B}$. Since $\mathcal{H}$ is self-complemented, we have $\overline{E} \in \mathcal{H}$; this implies that $(B, \overline{B})$ is not a 2-coloring of $\mathcal{H}$, which is a contradiction. Thus, it follows that $V_e(\mathcal{G}) \neq V$, and hence $V_e(\mathcal{G}) \subset V$.

CLAIM B: *If no $v \in V$ belongs to all edges of $\mathcal{G}$, then $\mathcal{H}$ is not saturated iff $\mathcal{G}$ is 2-colorable*.

The *only if* direction is immediate from Lemma 3.3. For the *if* direction, let $(B, \overline{B})$ be any 2-coloring of $\mathcal{G}$. Since no $v \in V$ belongs to all edges of $\mathcal{G}$, the 2-coloring must satisfy $|B \cap V_e(\mathcal{G})| \geq 2$ and $|\overline{B} \cap V_e(\mathcal{G})| \geq 2$. It follows that $B \notin Cov(\mathcal{G})$ and $\overline{B} \notin Cov(\mathcal{G})$. Since the latter is equivalent to $B \notin Cov(\overline{\mathcal{G}})$ and $\mathcal{H} = \mathcal{G} \cup \overline{\mathcal{G}}$, we have $B \notin Cov(\mathcal{H})$. This means that $\mathcal{H}$ is not saturated, and the *if* direction is shown.

In summary, by Claims A and B, the problem of deciding whether $\mathcal{H}$ is saturated is efficiently reducible to checking whether $V_e(\mathcal{G}) \not\subset V$ or to deciding whether $\mathcal{G}$ is not 2-colorable. The former is trivially polynomial, and the latter is polynomial since checking 2-colorability of a graph is polynomial [22].

It follows that H-SAT is polynomial under the asserted restriction. □

Simple hypergraph saturation. Toni has a brother Luigi, who also manages a big pizzeria. Luigi is proud that each of his pizze is "originale" which means that in his restaurant there are no Grande or Mini versions of pizzas. As he hears from the plans of his

brother, Luigi also intends to enlarge his pizza offer by a new pizza, which has, of course, to be a pizza originale. It is clear that Luigi's problem is a restricted version of Toni's problem. Since all pizze are "originali", in the corresponding hypergraph saturation problem the hypergraph of pizze is simple. Thus in terms of hypergraphs, we have

Problem: SIMPLE HYPERGRAPH SATURATION (SIMPLE-H-SAT)
Instance: A simple hypergraph $\mathcal{H} = (V, \mathcal{E})$ on vertices $V = \{v_1, \ldots, v_n\}$.
Question: Is $\mathcal{H}$ saturated ?

The question is if this restriction on the input makes H-SAT tractable. Though SIMPLE-H-SAT is clearly in co-**NP**, a precise classification is still open.

A saturation test for two simple hypergraphs is easily reducible to a test for one hypergraph.

DEFINITION 3.4. *Let $\mathcal{H}$ be a hypergraph on V and let v be a (possibly new) vertex. The hypergraph $\mathcal{H}^v$ is defined by $\mathcal{H}^v = (V \cup \{v\}, \{H \cup \{v\} : H \in \mathcal{H}\})$. Let $\mathcal{G}$ and $\mathcal{H}$ be simple hypergraphs on V, and let e, f be two (possibly new) vertices. Then $\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H}) = (V \cup \{e, f\}, \mathcal{G}^e \cup \mathcal{H}^f \cup \{V, \{e, f\}\})$.*

PROPOSITION 3.8. *Let $\mathcal{G}$ and $\mathcal{H}$ be simple hypergraphs which do not contain $\emptyset$ or V as edges, and let $e, f \notin V$ be distinct. Then $\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H})$ is simple, and $\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H})$ is saturated iff $\mathcal{G}$ and $\mathcal{H}$ are saturated.*

Proof. It is readily verified that $\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H})$ is simple. To show the other property, assume first that $\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H})$ is not saturated. That is, there exists $X \subseteq V \cup \{e, f\}$ such that $X \notin Cov(\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H}))$. Since $V \in \mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H})$ and $\{e, f\} \in \mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H})$, X must contain exactly one of e and f. If $e \in X$, it follows that $X - \{e\} \notin Cov(\mathcal{G})$; hence, $\mathcal{G}$ is not saturated. Analogously, if $f \in X$, it follows that $\mathcal{H}$ is not saturated. Thus the *if* direction holds. For the *only if* direction, assume without loss of generality that $\mathcal{G}$ is not saturated. Let $X \subseteq V$ such that $X \notin Cov(\mathcal{G})$. Then, we have that $X \cup \{e\} \notin Cov(\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H}))$, i.e., $\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H})$ is not saturated. □

We obtain that SIMPLE-H-SAT remains for the following subcase as hard as in the general case:

PROPOSITION 3.9. SIMPLE-H-SAT *is $\leq_m^p$-equivalent to the subcase where $\mathcal{H}$ is self-complemented.*

Proof. Notice that a hypergraph $\mathcal{H}$ is saturated if and only if $\overline{\mathcal{H}}$ is saturated: Indeed, for every $X \subseteq V$, we have $X \in Cov(\mathcal{H})$ iff $\overline{X} \in Cov(\overline{\mathcal{H}})$. Since the hypergraph $\mathcal{C}_{e,f}(\mathcal{G}, \mathcal{H})$ is self-complemented for $\mathcal{H} = \overline{\mathcal{G}}$, the result follows from Proposition 3.8. □ In terms of colorability, the following result can be formulated as an immediate consequence of Proposition 3.9 and Lemma 3.3. Observe that HP2C for simple hypergraphs is still **NP**-hard, as a hypergraph $\mathcal{H}$ is 2-colorable iff $\min(\mathcal{H})$ is 2-colorable.

COROLLARY 3.10. HP2C *for simple and self-complemented $\mathcal{H}$ is $\leq_m^p$-equivalent to co-*SIMPLE-H-SAT.

4. Relating hypergraph transversals and saturation. Problems involving hypergraph transversals appear in many practical applications (cf. Section 6), so their computational complexity is of major interest. Moreover, several computational problems on hypergraphs can be reformulated in terms of transversals; for example, finding a 2-coloring for a hypergraph $\mathcal{H}$ is equivalent to finding a transversal T of $\mathcal{H}$ such that $\overline{T}$ is also a transversal.

In this section we give first a characterization of saturated hypergraphs in terms of a transversal relation and then we will consider the problem of transversal recognition, where we will show that this problem is $\leq_m^p$-equivalent to SIMPLE-H-SAT. Moreover, we will investigate into necessary and sufficient criterions for simple hypergraph saturation and transversal hypergraph recognition, and we will show how the latter problem reduces to a very specific subcase.

4.1. Definitions and basic facts. We need some additional definitions first.

DEFINITION 4.1 (cf. [7]). *Let $\mathcal{H}$ and $\mathcal{G}$ be two hypergraphs. Then $\mathcal{H} \geq \mathcal{G}$ iff for every $H \in \mathcal{H}$ there exists $G \in \mathcal{G}$ such that $H \supseteq G$,* [2] *and $\mathcal{H} \leq \mathcal{G}$ iff for every $H \in \mathcal{H}$ there exists $G \in \mathcal{G}$ such that $H \subseteq G$.*

For every hypergraph $\mathcal{H}$, $\emptyset \geq \mathcal{H}$, and $\{\emptyset\} \geq \mathcal{H}$ iff $\emptyset \in \mathcal{H}$. Note that $\mathcal{H} \geq \mathcal{G}$ (resp. $\mathcal{H} \leq \mathcal{G}$) does not imply $\mathcal{G} \leq \mathcal{H}$ (resp. $\mathcal{G} \geq \mathcal{H}$); for example, let $\mathcal{H} = \{\{1\}\}$ and $\mathcal{G} = \{\{1\}, \{2\}\}$. $\leq$ is in some sense dual to $\geq$ with respect to complementation. For every hypergraph $\mathcal{H}$, $\mathcal{H} \leq \{V\}$, and $\{V\} \leq \mathcal{H}$ iff $V \in \mathcal{H}$. Notice that $\geq$ and $\leq$ are reflexive, transitive, and define partial orders on the simple hypergraphs on V. We observe the following properties:

PROPOSITION 4.1. *Let $\mathcal{G}$ and $\mathcal{H}$ be hypergraphs. Then,*

$$\begin{aligned} \mathcal{G} \geq \mathcal{H} \quad &\Longleftrightarrow \quad \mathcal{G} \geq \min(\mathcal{H}) \quad \Longleftrightarrow \quad \min(\mathcal{G}) \geq \min(\mathcal{H}); \\ \mathcal{G} \leq \mathcal{H} \quad &\Longleftrightarrow \quad \mathcal{G} \leq \max(\mathcal{H}) \quad \Longleftrightarrow \quad \max(\mathcal{G}) \leq \max(\mathcal{H}). \end{aligned}$$

We have the following characterization of saturated hypergraphs.

THEOREM 4.2. *Let $\mathcal{H} \neq \emptyset$ be a hypergraph. Then the following statements (i)–(iii) are equivalent:*

(i) *$\mathcal{H}$ is saturated.* $\quad (ii)$ $Tr(\overline{\max(\mathcal{H})}) \geq \min(\mathcal{H})$ $\quad (iii)$ $Tr(\overline{\mathcal{H}}) \geq \mathcal{H}$

Proof. $(i) \Rightarrow (ii)$. Assume $\mathcal{H}$ is saturated. Consider $T \in Tr(\overline{\max(\mathcal{H})})$. We have $T \in Cov(\mathcal{H})$, but since T is a transversal of $\overline{\max(\mathcal{H})}$, for every $E \in \mathcal{H}$ it holds that $T \not\subseteq E$. But then $E' \subseteq T$ for some $E' \in \mathcal{H}$, which implies $T \supseteq E''$ for some $E'' \in \min(\mathcal{H})$. It follows that $Tr(\overline{\max(\mathcal{H})}) \geq \min(\mathcal{H})$.
$(ii) \Rightarrow (iii)$. Check that $\overline{\max(\mathcal{H})} = \min(\overline{\mathcal{H}})$. Hence, $Tr(\overline{\max(\mathcal{H})}) = Tr(\min(\overline{\mathcal{H}})) = Tr(\overline{\mathcal{H}})$. Since $Tr(\overline{\max(\mathcal{H})}) \geq \min(\mathcal{H})$, it follows from Proposition 4.1 that $Tr(\overline{\mathcal{H}}) \geq \mathcal{H}$.
$(iii) \Rightarrow (i)$. Assume $\mathcal{H}$ is not saturated, i.e., there exists $X \subseteq V$ such that $X \notin Cov(\mathcal{H})$. Since $X \not\subseteq E$ for every $E \in \mathcal{H}$, X is a transversal of $\overline{\mathcal{H}}$. Let $X' \subseteq X$ be a minimal transversal of $\overline{\mathcal{H}}$. Now consider any $E \in \mathcal{H}$; since $E \not\subseteq X$, we have $E \not\subseteq X'$. Consequently, $Tr(\overline{\mathcal{H}}) \not\geq \mathcal{H}$. □

COROLLARY 4.3. *To test if $Tr(\mathcal{H}) \geq \mathcal{H}$ holds for a hypergraph $\mathcal{H}$ is co-***NP***-complete, even if $\mathcal{H}$ is self-complemented and contains only edges of size k and $n-k$, for fixed $k \geq 3$.*

Proof. Membership in co-**NP** follows since a guess for $T \in Tr(\mathcal{H})$ such that for every $E \in \mathcal{H}$, $T \not\supseteq E$ can be verified in polynomial time. co-**NP**-hardness under the asserted restriction follows from Theorems 4.2 and 3.5. □

We remark that the equivalence of (i) and (iii) in Theorem 4.2, restricted to simple hypergraphs, is implicit in a result of [51], which investigates Sperner families in the context of relational database theory. Notice that this criterion does not describe the transversal hypergraph *explicitly*. We will provide such a description later in this section.

4.2. The main problem. Besides SIMPLE-H-SAT the recognition of the transversal hypergraph is the second issue of central concern in our study. We will show now that the complexity of this decision problem is of importance for computing the transversal hypergraph, and then we will demonstrate $\leq_m^p$-equivalence of SIMPLE-H-SAT and transversal hypergraph recognition. Let us start with a precise problem statement:

Problem: TRANSVERSAL HYPERGRAPH (TRANS-HYP)
Instance: Two hypergraphs $\mathcal{G} = (V, \mathcal{E}_1)$ and $\mathcal{H} = (V, \mathcal{E}_2)$ on vertices $V = \{v_1, \ldots, v_n\}$.
Question: Does $\mathcal{H} = Tr(\mathcal{G})$ hold ?

[2] On [7, p. 45] $\geq$ is denoted by $\prec$.

The problem is in co-**NP**, since a guess for $T \in Tr(\mathcal{G})$ such that $T \notin \mathcal{H}$ can be verified in polynomial time. However, there is no trivial evidence that the problem is in **NP**, even less that it is in **P**. We observe that TRANS-HYP efficiently reduces to the following subproblem:

PROPOSITION 4.4. TRANS-HYP *is* $\leq_m^p$*-equivalent to the subcase in which* $\mathcal{G}$ *and* $\mathcal{H}$ *are simple.*

Proof. Note that $\mathcal{H} = Tr(\mathcal{G})$ implies that $\mathcal{H}$ is simple. Furthermore, $Tr(\mathcal{G}) = Tr(\min(\mathcal{G}))$. Since $\min(\mathcal{G})$ is a simple hypergraph and computable in polynomial time, the result follows. □

The significance of TRANS-HYP on the related search problem, namely the computation of the transversal hypergraph, is captured by the following theorem.

THEOREM 4.5. *If* TRANS-HYP *is not in* **P***, then no output-polynomial total time algorithm for computing the transversal hypergraph exists.*

Proof. Assume that an algorithm A computes $Tr(\mathcal{F})$ from $\mathcal{F}$ in output-polynomial total time. Let its runtime be bounded by a polynomial $p(IS(\mathcal{F}), N)$, where IS denotes the input size and $N = |Tr(\mathcal{F})|$. To solve TRANS-HYP, construct an algorithm A' which works as follows: A' simulates A applied on $\mathcal{G}$, but does this for at most $p(IS(\mathcal{G}), |\mathcal{H}|)$ steps. If A did not terminate within $p(IS(\mathcal{G}), |\mathcal{H}|)$ steps, then $\mathcal{H} \neq Tr(\mathcal{G})$. Otherwise, A' compares the hypergraph output by A to $\mathcal{H}$; these hypergraphs are identical iff $\mathcal{H} = Tr(\mathcal{G})$. It is easy to see that the runtime of A' can be bounded by a polynomial $q(IS(\mathcal{G}), |\mathcal{H}|)$. (The book-keeping between simulation steps can be done in polynomial time.) Hence, A' is polynomial with respect to the input size. The result follows. □

4.3. Characterizations of the transversal hypergraph. Since one can test in polynomial time for every $X \subseteq V$ whether X is a minimal transversal for a given hypergraph $\mathcal{H}$, we wonder if we can infer that $\mathcal{G} = Tr(\mathcal{H})$ if all edges of $\mathcal{G}$ are minimal transversals of $\mathcal{H}$ and vice versa, since this can be checked in polynomial time. The answer is no:

PROPOSITION 4.6. *Let* $\mathcal{G}$ *and* $\mathcal{H}$ *be simple hypergraphs. Then* $\mathcal{H} \subseteq Tr(\mathcal{G})$ *and* $\mathcal{G} \subseteq Tr(\mathcal{H})$ *does not imply* $\mathcal{H} = Tr(\mathcal{G})$.

Proof. We give a counterexample. Let $V = \{a, \ldots, g\}$. Consider the hypergraphs

$$\begin{aligned}\mathcal{H} = \{ &\{a,b,c\}, \{a,d,g\}, \{a,e,f\}, \{b,d,f\}, \{b,e,g\}, \{c,d,e\}, \{c,f,g\},\\ &\{d,e,f,g\}, \{b,c,e,f\}, \{b,c,d,g\}, \{a,c,e,g\}, \{a,c,d,f\}, \{a,b,f,g\} \},\end{aligned}$$

$$\begin{aligned}\mathcal{G} = \{ &\{a,b,c,d\}, \{a,b,c,e\}, \{a,b,c,f\}, \{a,b,c,g\}, \{a,b,d,g\}, \{a,c,d,g\},\\ &\{a,d,e,g\}, \{a,d,f,g\}, \{a,b,e,f\}, \{a,c,e,f\}, \{a,d,e,f\}, \{a,e,f,g\},\\ &\{a,b,d,f\}, \{b,c,d,f\}, \{b,d,e,f\}, \{b,d,f,g\}, \{a,b,e,g\}, \{b,c,e,g\},\\ &\{b,d,e,g\}, \{b,e,f,g\}, \{a,c,d,e\}, \{b,c,d,e\}, \{c,d,e,f\}, \{c,d,e,g\} \}.\end{aligned}$$

It holds that $\mathcal{H} \subseteq Tr(\mathcal{G})$ and $\mathcal{G} \subseteq Tr(\mathcal{H})$, but $\mathcal{G} \neq Tr(\mathcal{H})$ as $\{c,f,g\} \in Tr(\mathcal{H})$ and $\{c,f,g\} \notin \mathcal{G}$.[3] □

For further characterizations of the transversal hypergraph of a simple hypergraph we introduce a special delete operator which removes from every edge of a hypergraph a vertex in all possible ways:

DEFINITION 4.2. *Let* V *be a finite set, and let* $Hyp(V) = 2^{2^V}$ *denote the set of all hypergraphs on* V*. The mapping* $\delta : Hyp(V) \to Hyp(V)$ *is defined by* $\delta(\mathcal{H}) = (V, \{H - \{v\} : H \in \mathcal{H}, v \in H\})$.

[3] $\mathcal{H} = \mathcal{P}_7 \cup \overline{\mathcal{P}_7} - \{ \{a,b,d,e\} \}$ and $\mathcal{G} = \{E \cup \{v\} : E \in \mathcal{P}_7 - \{\{c,f,g\}\}, v \in \overline{E}\}$ for the hypergraph $\mathcal{P}_7 = \{\{a,b,c\}, \{c,d,e\}, \{b,d,f\}, \{a,e,f\}, \{a,d,g\}, \{b,e,g\}, \{c,f,g\}\}$ on $V = \{a, \ldots, g\}$ from [7, p.47,Fig.1].

EXAMPLES: For the hypergraph $\mathcal{H}$ = {{1},{2,3},{1,3,4}} on $V = \{1,2,3,4\}$, we have $\delta(\mathcal{H}) = \{\emptyset, \{2\}, \{3\}, \{1,3\}, \{1,4\}, \{3,4\}\}$. Note in particular that $\delta(\emptyset) = \emptyset$ and that $\delta(\{\emptyset\}) = \emptyset$.

We observe the following relationships.

PROPOSITION 4.7. *Let $\mathcal{G}$ and $\mathcal{H}$ be simple hypergraphs such that $V_e(\mathcal{G}) \neq \emptyset$ and $V_e(\mathcal{H}) \neq \emptyset$. Then*

(i) *$\mathcal{H} \subseteq Tr(\mathcal{G})$ implies $\mathcal{H} \leq \overline{\delta(\mathcal{G})}$, but the converse does not hold.*

(ii) *$\overline{\delta(\mathcal{H})} \geq Tr(\mathcal{H})$, i.e., every edge of $\overline{\delta(\mathcal{H})}$ is a transversal of $\mathcal{H}$.*

(iii) *$Tr(\mathcal{H}) \leq \overline{\delta(\mathcal{H})}$, i.e., every minimal transversal of $\mathcal{H}$ is contained in some edge of $\overline{\delta(\mathcal{H})}$.*

Proof.

(i): Consider $T \in \mathcal{H}$. Since $T \in Tr(\mathcal{G})$, it follows that for every $v \in T$ there exists $G \in \mathcal{G}$ such that $(T - \{v\}) \cap G = \emptyset$; hence, $T - \{v\} \subseteq \overline{G}$. This implies that $T \subseteq \overline{G} \cup \{v\} = \overline{G - \{v\}}$, from which it follows immediately that $\mathcal{H} \leq \overline{\delta(\mathcal{G})}$.

The following example shows that the converse does not hold. Consider $\mathcal{H} = \{\{1\}\}$ and $\mathcal{G} = \{\{1\},\{2\}\}$ on $V = \{1,2\}$. Then, $\overline{\delta(\mathcal{G})} = \{\{1,2\}\}$ and $\mathcal{H} \leq \overline{\delta(\mathcal{G})}$, but $\mathcal{H} \not\subseteq Tr(\mathcal{G}) = \{\{1,2\}\}$.

(ii): Consider $H \in \mathcal{H}$. $\overline{H}$ is not a transversal of $\mathcal{H}$, but since $\mathcal{H}$ is simple, it is a transversal of the hypergraph $\mathcal{H} - \{H\}$. Thus for every $v \in H$, $\overline{H} \cup \{v\}$ is a transversal of $\mathcal{H}$; hence, $\overline{H} \cup \{v\} = \overline{H - \{v\}}$ contains a minimal transversal of $\mathcal{H}$. The claim follows immediately from the definition of δ.

(iii): For every minimal transversal T of $\mathcal{H}$ and every vertex $v \in T$ there exists an edge H of $\mathcal{H}$ such that $T \cap H = \{v\}$; hence, $T \subseteq \overline{H} \cup \{v\}$, but $\overline{H} \cup \{v\} = \overline{H - \{v\}} \in \overline{\delta(\mathcal{H})}$. □

Note that in the above proposition (i) states a necessary, but not sufficient condition for TRANS-HYP. With the δ-operator we have the following important characterization of saturated simple hypergraphs:

THEOREM 4.8. *Let $\mathcal{H}$ be a simple hypergraph. Then $\mathcal{H}$ is saturated if and only if $Tr(\mathcal{H}) = \min(\overline{\delta(\mathcal{H})})$.*

Proof. We prove the *only if* direction first. Assume $\mathcal{H}$ is saturated, but $Tr(\mathcal{H}) \neq \min(\overline{\delta(\mathcal{H})})$. Note that this assumption implies $\mathcal{H} \neq \emptyset$ and $\mathcal{H} \neq \{\emptyset\}$, hence $V_e(\mathcal{H}) \neq \emptyset$.

CLAIM A: $Tr(\mathcal{H}) \not\subseteq \min(\overline{\delta(\mathcal{H})})$. Proof: Suppose to the contrary that $Tr(\mathcal{H}) \subseteq \min(\overline{\delta(\mathcal{H})})$; hence, since $Tr(\mathcal{H}) \neq \min(\overline{\delta(\mathcal{H})})$, there exists an edge $E \in \min(\overline{\delta(\mathcal{H})}) - Tr(\mathcal{H})$. From Proposition 4.7$(ii)$, it follows that E is a transversal of $\mathcal{H}$; thus, E contains some $T \in Tr(\mathcal{H})$. Consider any such T. Since $\min(\overline{\delta(\mathcal{H})})$ is simple, it follows that $T \notin \min(\overline{\delta(\mathcal{H})})$. Hence, T gives rise to a contradiction of $Tr(\mathcal{H}) \subseteq \min(\overline{\delta(\mathcal{H})})$. This proves the claim.

From Claim A and the assumption $Tr(\mathcal{H}) \neq \min(\overline{\delta(\mathcal{H})})$, it follows that there exists $T \in Tr(\mathcal{H})$ such that $T \in Tr(\mathcal{H}) - \min(\overline{\delta(\mathcal{H})})$.

CLAIM B: $\overline{H} \cap \overline{T} \neq \emptyset$. Proof: Assume to the contrary that $\overline{H} \cap \overline{T} = \emptyset$, i.e. that $\overline{H} \subseteq T$. Clearly, $\overline{H}$ is not a transversal of $\mathcal{H}$; however, since $\mathcal{H}$ is simple, for every $v \in H$ it holds that $\overline{H} \cup \{v\}$ is a transversal of $\mathcal{H}$. From this and $\overline{H} \subseteq T$, $T \in Tr(\mathcal{H})$ we infer that $T = \overline{H} \cup \{v'\}$ for some $v' \in H$. Hence, $T \in \overline{\delta(\mathcal{H})}$, which means that T is a superset of some edge in $\min(\overline{\delta(\mathcal{H})})$. Since $T \in Tr(\mathcal{H})$, we infer from Proposition 4.7(ii) that $T \in \min(\overline{\delta(\mathcal{H})})$. We reached a contradiction. Thus $\overline{H} \not\subseteq T$, i.e. $\overline{H} \cap \overline{T} \neq \emptyset$.

Since for each $H \in \mathcal{H}$, a vertex e exists such that $E = \overline{H} \cup \{e\} \in \overline{\delta(\mathcal{H})}$, it follows from Claim B that $\overline{T}$ is a transversal of $\overline{\mathcal{H}}$. Since $\mathcal{H}$ is saturated, we have by Theorem 4.2(iii) that $Tr(\overline{\mathcal{H}}) \geq \mathcal{H}$. Thus there exists $H \in \mathcal{H}$ such that $\overline{T} \supseteq H$, which is equivalent to $T \subseteq \overline{H}$. But this means that T is not a transversal of $\mathcal{H}$; this is a contradiction to the existence of T which followed from Claim A. Thus, the assumption $Tr(\mathcal{H}) \neq \min(\overline{\delta(\mathcal{H})})$, from which we derived

Claim A, is not consistent. This proves the *only if* direction.

To prove the *if* direction, assume that $Tr(\mathcal{H}) = \min(\overline{\delta(\mathcal{H})})$, but $\mathcal{H}$ is not saturated. Notice that $\mathcal{H} \neq \emptyset$. Since $\mathcal{H}$ is not saturated, there exists a set $T \subseteq V$ such that $T \notin Cov(\mathcal{H})$. In particular, we have for every $H \in \mathcal{H}$ that $T \not\supseteq H$, and hence $\overline{\overline{T}} \cap H \neq \emptyset$. This means that $\overline{T}$ is a transversal of $\mathcal{H}$. From the assumption $Tr(\mathcal{H}) = \min(\overline{\delta(\mathcal{H})})$, we infer that $\overline{T} \supseteq E$ for some $E \in \min(\overline{\delta(\mathcal{H})})$. By the definition of $\delta(\mathcal{H})$, we have that $E = \overline{H} \cup \{v\}$ for some H and v such that $H \in \mathcal{H}$ and $v \in H$. Hence, $\overline{T} \supseteq \overline{H} \cup \{v\}$, and thus $\overline{T} \supseteq \overline{H}$. The latter means $T \subseteq H$, which implies $T \in Cov(\mathcal{H})$. Thus we reached a contradiction to the existence of T. This proves the *if* direction. □

Given a simple hypergraph $\mathcal{H}$, one can easily compute $\mathcal{H}' = \min(\overline{\delta(\mathcal{H})})$ in polynomial time. Since checking whether each edge of $\mathcal{H}$ is a minimal transversal of $\mathcal{H}'$ and vice versa is possible in polynomial time, we wonder if this necessary condition for saturation of $\mathcal{H}$ is also sufficient. Unfortunately, this does not hold:

PROPOSITION 4.9. *Let $\mathcal{H}$ be a simple hypergraph, and let $\mathcal{H}' = \min(\overline{\delta(\mathcal{H})})$. Then $\mathcal{H} \subseteq Tr(\mathcal{H}')$ and $\mathcal{H}' \subseteq Tr(\mathcal{H})$ does not imply that $\mathcal{H}$ is saturated. The same holds if $\mathcal{H}$ is self-complemented.*

A counterexample for unrestricted $\mathcal{H}$ is not hard to find (e.g., in the proof of Proposition 4.6, $\mathcal{G} = \min(\overline{\delta(\mathcal{H})})$). Imposing self-complementarity on $\mathcal{H}$ makes this more challenging, in particular to find a counterexample with as few edges or vertices as possible; we can offer an instance of $\mathcal{H}$ with 308 edges on 22 vertices.

It follows immediately from Theorem 4.8 that SIMPLE-H-SAT is $\leq_m^p$-transformable into TRANS-HYP. The next theorem implies the converse, which establishes that SIMPLE-H-SAT and TRANS-HYP are $\leq_m^p$-equivalent. We use the following simple, but helpful lemma:

LEMMA 4.10. *If $\mathcal{H}$ is simple, then for each $E \in Tr(\mathcal{H})$, there exists no $E' \in \mathcal{H}$ such that $E' \subseteq \overline{E}$.*

Proof. Indeed, if $E \in Tr(\mathcal{H})$, then $E \cap E' \neq \emptyset$ for all $E' \in \mathcal{H}$; hence, $E' - \overline{E} \neq \emptyset$, i.e., $E' \not\subseteq \overline{E}$. □

THEOREM 4.11. *Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two simple hypergraphs on V, and let $e, f \notin V$ be distinct. Define a hypergraph $\mathcal{H} = (V', \mathcal{F}_1 \cup \mathcal{F}_2)$, where $V' = V \cup \{e, f\}$, $\mathcal{F}_1 = \mathcal{G}_1$, and $\mathcal{F}_2 = \{\overline{E} \cup \{e, f\} : E \in \mathcal{G}_2\}$. Then $\mathcal{G}_1 = Tr(\mathcal{G}_2)$ iff $\mathcal{H}$ is simple and saturated.*

Proof. For the *only if* direction, let $\mathcal{G}_1 = Tr(\mathcal{G}_2)$ hold. We first show that $\mathcal{H}$ is simple. Suppose $\mathcal{H}$ is not simple. Since $\mathcal{F}_1, \mathcal{F}_2$ constitute simple hypergraphs, there must exist $F_1 \in \mathcal{F}_1, F_2 \in \mathcal{F}_2$ such that $F_1 \subseteq F_2$ or $F_2 \subseteq F_1$. By Lemma 4.10 and the construction of $\mathcal{F}_1$ and $\mathcal{F}_2$ it follows that $F_1 \subseteq F_2$ is impossible to hold. However, $F_2 \subseteq F_1$ is also impossible because $e \in F_2$ but $e \notin F_1$. Thus we have reached a contradiction, and $\mathcal{H}$ is simple. Now we show that $\mathcal{H}$ must also be saturated. Assume this is not the case, i.e., there exists $X \subseteq V', X \notin Cov(\mathcal{H})$. Since for all $F \in \mathcal{F}_2$ we have $X \not\subseteq F$, it follows that X is a transversal of $\overline{\mathcal{F}_2} = \overline{\overline{\mathcal{G}_2}} = \mathcal{G}_2$. Since $\mathcal{F}_1$ constitutes $Tr(\mathcal{G}_2)$, this implies that $E \subseteq X$ for some $E \in \mathcal{F}_1$, which means that $X \in Cov(\mathcal{H})$, a contradiction. Thus $\mathcal{H}$ is saturated; the *only if* direction is proved.

To prove the *if* direction, assume that $\mathcal{H}$ is simple and saturated, but $\mathcal{G}_1 \neq Tr(\mathcal{G}_2)$. We show that $\mathcal{G}_1 \not\supset Tr(\mathcal{G}_2)$. Assume that $\mathcal{G}_1 \supset Tr(\mathcal{G}_2)$. Hence, there exists a set $E \in \mathcal{G}_1 - Tr(\mathcal{G}_2)$. Since $\mathcal{G}_1 \supset Tr(\mathcal{G}_2)$ and $\mathcal{G}_1$ is simple, it follows that E is not a transversal of $\mathcal{G}_2$. Since E is not a transversal of $\mathcal{G}_2$, there exists some $E' \in \mathcal{G}_2$ such that $E \cap E' = \emptyset$; hence, $E \subset \overline{E'} \cup \{e, f\}$. By construction of $\mathcal{H}$, we have that $E, \overline{E'} \cup \{e, f\} \in \mathcal{H}$. This means that $\mathcal{H}$ is not simple, which is a contradiction. Thus it follows that $\mathcal{G}_1 \not\supset Tr(\mathcal{G}_2)$.

Since $\mathcal{G}_1 \neq Tr(\mathcal{G}_2)$ and $\mathcal{G}_1 \not\supset Tr(\mathcal{G}_2)$, there exists a $T \in Tr(\mathcal{G}_2) - \mathcal{G}_1$. Consider $T \cup \{e\}$. Because T is a transversal of $\mathcal{G}_2 = \overline{\mathcal{F}_2}$, it is clear that there is no $F \in \mathcal{F}_2$ such that $T \cup \{e\} \subseteq F$. There is no $F \in \mathcal{F}_2$ such that $T \cup \{e\} \supseteq F$ since $f \in F, f \notin T$, and

there is no $F \in \mathcal{F}_1$ with the property $T \cup \{e\} \subseteq F$ because $e \notin F$. Since $\mathcal{H}$ is saturated, this implies that there is some $F \in \mathcal{F}_1$ such that $F \subseteq T \cup \{e\}$, from which clearly $F \subseteq T$ follows. Because $\mathcal{H}$ is simple and $\mathcal{F}_2 \subseteq \mathcal{H}$, it holds for every $G \in \mathcal{F}_2$ that $F \not\subseteq G$; thus, F is a transversal of $\overline{\mathcal{F}_2} = \mathcal{G}_2$. Since $T \in Tr(\mathcal{G}_2)$, it follows that $F = T$. Hence, $T \in \mathcal{G}_1$, which is a contradiction to $T \in Tr(\mathcal{G}_2) - \mathcal{G}_1$. This proves the *if* direction. □

Now the above stated relationship between SIMPLE-H-SAT and TRANS-HYP is easy to establish:

THEOREM 4.12. SIMPLE-H-SAT *is* $\leq_m^p$*-equivalent to* TRANS-HYP.

Proof. Theorems 4.8 and 4.11 imply that SIMPLE-H-SAT $\leq_m^p$-reduces to TRANS-HYP and vice versa. □

4.4. Selftransversal hypergraphs. As announced above, we show that recognizing the transversal hypergraph $\leq_m^p$-reduces to the SELFTRANSVERSALITY problem, which is as follows.

Problem: SELFTRANSVERSALITY
Instance: A hypergraph $\mathcal{H} = (V, \mathcal{E})$ on vertices $V = \{v_1, \ldots, v_n\}$.
Question: Does $\mathcal{H} = Tr(\mathcal{H})$ hold ?

Selftransversal hypergraphs have attracted much interest by mathematicians, cf. [36, 6, 7]. The simplest example of such a hypergraph is $\mathcal{H} = (\{x\}, \{\{x\}\})$. Another example is the hypergraph $\mathcal{H} = \{\{x, y\}, \{x, y\}, \{y, z\}\}$. For a thorough study of selftransversal hypergraphs, the reader is referred to [7, chapter 2]. We have the following result.

THEOREM 4.13. *To decide if a hypergraph* $\mathcal{H}$ *satisfies* $Tr(\mathcal{H}) = \mathcal{H}$ *is* $\leq_m^p$*-equivalent to* SIMPLE-H-SAT.

Proof. We show that TRANS-HYP is $\leq_m^p$-reducible to SELFTRANSVERSALITY, from which by Theorem 4.12 the result follows. Let without loss of generality be $\mathcal{G}$ and $\mathcal{H}$ two simple, nonempty hypergraphs on V and let $e, f \notin V$ be two new distinct vertices. Define a hypergraph $\mathcal{D}$ on $V' = V \cup \{e, f\}$ by $\mathcal{D} = \mathcal{G}^e \cup \mathcal{H}^f \cup \{\{e, f\}\}$. We claim that $\mathcal{G} = Tr(\mathcal{H})$ iff $\mathcal{D} = Tr(\mathcal{D})$.

To prove this, consider $Tr(D)$. We have that $\{e, f\} \in Tr(\mathcal{D})$ and also $\{e, f\} \in \mathcal{D}$. Every $T \in Tr(\mathcal{D}) - \{\{e, f\}\}$ distinct from $\{e, f\}$ contains either e or f, but not both. Assume that T contains e, but not f. Since no edge in $\mathcal{H}^f$ contains e and since all other edges in $\mathcal{D} - \mathcal{H}^f$ contain e, we infer that $T - \{e\}$ must be a transversal of $\mathcal{H}$. Conversely, if T' is a transversal of $\mathcal{H}$, then $T' \cup \{e\}$ is easily shown to be a transversal of $\mathcal{D}$. This implies that the minimal transversals of $\mathcal{D}$ which contain e but not f are given by $\mathcal{T}_e = \{E \cup \{e\} : E \in Tr(\mathcal{H})\} = Tr(\mathcal{H})^e$. From this it is immediately verified that $\mathcal{G} = Tr(\mathcal{H})$ if and only if $\mathcal{G}^e = \mathcal{T}_e$. In the same way one can show that the set $\mathcal{T}_f$ of minimal transversals of $\mathcal{D}$ which contain f but not e satisfies $\mathcal{T}_f = \mathcal{H}^f$ if and only if $\mathcal{H} = Tr(\mathcal{G})$. Since $\mathcal{G}$ and $\mathcal{H}$ are simple, by Proposition 2.2 $\mathcal{G} = Tr(\mathcal{H})$ iff $\mathcal{H} = Tr(\mathcal{G})$. Hence, since $Tr(\mathcal{D}) = \mathcal{T}_e \cup \mathcal{T}_f \cup \{\{e, f\}\}$, we have that $\mathcal{G} = Tr(\mathcal{H})$ iff $\mathcal{D} = Tr(\mathcal{D})$; the claim is proved. □

From this result, we obtain by the close relation between 2-colorable intersecting hypergraphs and selftransversal hypergraphs [7] the following result on deciding 2-colorability of intersecting hypergraphs.

COROLLARY 4.14. HP2C *for intersecting hypergraphs is* $\leq_m^p$*-equivalent to co-*SIMPLE-H-SAT.

Proof. Notice that $\mathcal{H}$ is intersecting iff $\min(\mathcal{H})$ is intersecting, and that $\mathcal{H}$ is 2-colorable iff $\min(\mathcal{H})$ is 2-colorable. Thus without loss of generality we may assume that $\mathcal{H}$ is simple and intersecting, and in addition, that $\mathcal{H}$ contains no loops, i.e. singleton edges. A simple intersecting hypergraph $\mathcal{H}$ without loops satisfies $\mathcal{H} = Tr(\mathcal{H})$ if and only if it is not 2-colorable [7]. Thus, if $\mathcal{H}$ is intersecting, 2-colorability of $\mathcal{H}$ is $\leq_m^p$-reducible to co-SELFTRANSVERSALITY, which by Theorem 4.13 is $\leq_m^p$-equivalent to co-SIMPLE-H-SAT. □

5. Polynomial cases. Although we do not know whether SIMPLE-H-SAT and TRANS-HYP are intractable in their general problem statement, in practical occurrences there are (natural) restrictions on the instances of a problem such that the problem reduces to a subproblem of the general problem. Since a subproblem might be computationally easier, there is hope to get efficient algorithms for it even if the more general problem is intractable. For example, in the pizza baker's problem it seems quite reasonable to assume that all pizze have approximately the same number of food items; in this case, however, the problem becomes polynomial-time solvable for both Toni and Luigi. Narrowing the "frontier" between the intractable general problem and tractable subcases [21], we identify some important subcases of SIMPLE-H-SAT and TRANS-HYP which are in **P**, among them the recognition of the minimal transversals of a hypergraph with bounded edge size and hypergraph saturation if the difference between rank and antirank is bounded. In particular, we present algorithms for output-efficient computation of the transversal hypergraph, if the edge-size is small (probably the most important of the restrictions treated for practice) and if it is large, or if the hypergraph is acyclic.

5.1. Restrictions on the edge size. Let us consider restrictions on the size of the edges of a hypergraph. A straightforward restriction of this kind which is important in practice is a constant upper bound. Fortunately, the transversal hypergraph is output-efficiently computable in this case, which implies that the minimal transversals of such a hypergraph can be recognized in polynomial time and that also saturation of a simple hypergraph of this form is efficiently decidable.

DEFINITION 5.1. *Let $\mathcal{H}$ be a hypergraph on vertices V, and let $x \in V$. The degree $d(x, \mathcal{H})$ of vertex x in hypergraph $\mathcal{H}$ is defined as $d(x, \mathcal{H}) = |\{E \in \mathcal{H} : x \in E\}|$.*

In the proof we apply a result from combinatorial studies by Berge and Duchet:

LEMMA 5.1 ([7, p.58, Corollary 1]). *Let $\mathcal{G}$ be a simple hypergraph, and let $k \geq 2$ be an integer. Then $r(Tr(\mathcal{G})) \leq k$ iff for all $\mathcal{G}' \subseteq \mathcal{G}$ with $|\mathcal{G}'| = k + 1$ there exists an $E \in \mathcal{G}$ s.t. $E \subseteq \{v \in V : d(v, \mathcal{G}') > 1\}$.*

THEOREM 5.2. *Let $\mathcal{H}$ be a hypergraph. If the size of the edges of $\mathcal{H}$ is bounded by some constant k, i.e., $r(\mathcal{H}) \leq k$, then $Tr(\mathcal{H})$ is computable in incremental-polynomial time.*

Proof. We describe an algorithm which has the desired property. The following characterization of $Tr(\mathcal{H})$ for a simple, nonempty hypergraph $\mathcal{H}$ is obtained from Lemma 5.1. Let $r = r(\mathcal{H})$, and for any hypergraph $\mathcal{F}$ and integer $i \geq 0$ denote by $\mathcal{F}|_i$ the hypergraph $\mathcal{F}_i = \{E \in \mathcal{F} : |E| \leq i\}$. Then,

$$\mathcal{G} = Tr(\mathcal{H}) \iff (\mathcal{G} \subseteq Tr(\mathcal{H})) \wedge \mathbf{P} \wedge \mathbf{Q}\,, \tag{1}$$

where
$\mathbf{P} \equiv Tr(\mathcal{G})|_r \subseteq \mathcal{H}$,
$\mathbf{Q} \equiv$ there exists no $\mathcal{G}' \subseteq \mathcal{G}$ with $|\mathcal{G}'| = r + 1$ such that for all $E \in \mathcal{G}$, $E \not\subseteq \{x \in V : d(x, \mathcal{G}') > 1\}$.

It is readily checked that (1) holds if $r \leq 1$. We show that (1) also holds if $r \geq 2$. If $\mathcal{G}$ $=Tr(\mathcal{H})$ and $\mathcal{H}$ is simple, then $\mathcal{H}$ = $Tr(\mathcal{G})$, and the *only if* direction follows by Lemma 5.1. For the *if* direction, if $\mathcal{H}$ is simple, $\mathcal{G} \subseteq Tr(\mathcal{H})$, and **P** holds, then $\mathcal{H} \subseteq Tr(\mathcal{G})$. To show this, consider $H \in \mathcal{H}$. Since $\mathcal{G} \subseteq Tr(\mathcal{H})$, H is a transversal of $\mathcal{G}$. Let $H' \subseteq H$ be an arbitrary minimal transversal of $\mathcal{G}$. Since $|H'| \leq r$, it follows from **P** that $H' \in \mathcal{H}$; as $\mathcal{H}$ is simple, it follows that $H' = H$. Hence, $\mathcal{H} \subseteq Tr(\mathcal{G})$.

Since by validity of **Q** and Lemma 5.1 $r(Tr(\mathcal{G})) \leq r$, it follows from **P** that $\mathcal{H}$ = $Tr(\mathcal{G})$.

We derive now from (1) a method for incrementally computing $Tr(\mathcal{H})$, where $\mathcal{H}$ is simple and nonempty. If $\mathcal{G} \subseteq Tr(\mathcal{H})$ and $\mathcal{G} \neq Tr(\mathcal{H})$, then **P** or **Q** is false. In this case, some $T_1 \in Tr(\mathcal{H})$ such that $T_1 \notin \mathcal{G}$ can be found as follows.

(i) Assume that **P** is false. Hence, there exists $T \in Tr(\mathcal{G})|_r - \mathcal{H}$.

CLAIM A: *$\overline{T}$ is a transversal of $\mathcal{H}$*. Proof: Notice that $\overline{T}$ is not a transversal of $\mathcal{H}$ iff there exists an $H \in \mathcal{H}$ such that $\overline{T} \cap H = \emptyset$, which is equivalent to $H \subseteq T$. Since $\mathcal{G} \subseteq Tr(\mathcal{H})$, H is a transversal of $\mathcal{G}$. As $T \in Tr(\mathcal{G})|_r - \mathcal{H}$, T is a minimal transversal of $\mathcal{G}$. Since $H \subseteq T$, it follows that $H = T$; hence, $T \in \mathcal{H}$. This is a contradiction. Consequently, $\overline{T}$ is a transversal of $\mathcal{H}$.

CLAIM B: $\overline{T} \notin Cov(\mathcal{G})$. Proof: Assume to the contrary that $\overline{T} \in Cov(\mathcal{G})$. Assume furthermore that there exists $G \in \mathcal{G}$ such that $G \subseteq \overline{T}$. This implies $G \cap T = \emptyset$, which means that T is not a transversal of $\mathcal{G}$. However, by definition $T \in Tr(\mathcal{G})|_r - \mathcal{H}$, and thus T is a transversal of $\mathcal{G}$. We reached a contradiction, and thus G does not exist. Assume otherwise that there exists $G \in \mathcal{G}$ such that $\overline{T} \subset G$. This implies that $G \notin Tr(\mathcal{H})$, which contradicts $\mathcal{G} \subseteq Tr(\mathcal{H})$. This shows that $\overline{T} \notin Cov(\mathcal{G})$.

By Claims A and B, $\overline{T}$ is a transversal of $\mathcal{H}$ such that $\overline{T} \notin Cov(\mathcal{G})$. Thus $\overline{T}$ contains some $T_1 \in Tr(\mathcal{H})$ such that $T_1 \notin \mathcal{G}$; in fact, every $T_1 \in Tr(\mathcal{H})$ such that $T_1 \subseteq \overline{T}$ has this property.

(ii) Assume that **Q** is false. Hence, there exists $\mathcal{G}' \subseteq \mathcal{G}$ with $|\mathcal{G}'| = r + 1$ such that for every $E \in \mathcal{G}$ we have $E \not\subseteq \{x \in V : d(x, \mathcal{G}') > 1\}$. Note that this is only possible if $|\mathcal{G}| > r(\mathcal{H})$, which implies that $r(\mathcal{H}) \geq 2$. By Lemma 5.1, it follows that $r(Tr(\mathcal{G})) > r$, thus $\mathcal{H} \neq Tr(\mathcal{G})$.

CLAIM C: *$T = \{x \in V : d(x, \mathcal{G}') > 1\}$ is a transversal of $\mathcal{H}$ and $T \notin Cov(\mathcal{G})$*. Proof: Note that if T is a transversal of $\mathcal{H}$, then $T \notin Cov(\mathcal{G})$, as no edge of $\mathcal{G}$ is contained in T, and, on the other hand, for no $E \in \mathcal{G}$ it holds that $T \subset E$ as $E \in Tr(\mathcal{H})$. Thus to prove our claim, it remains to show that T is a transversal of $\mathcal{H}$. Assume that T is not a transversal of $\mathcal{H}$. Hence, there exists some $E \in \mathcal{H}$ such that $E \cap T = \emptyset$, i.e., $E \subseteq \overline{T}$. However, this is impossible: Since T contains no edge of $\mathcal{G}$, $\overline{T}$ is a transversal of $\mathcal{G}$, and thus also a transversal of $\mathcal{G}'$. From the definition of T, each vertex of $\overline{T}$ appears in at most one edge of $\mathcal{G}'$. Since $\mathcal{G}'$ has $r + 1$ edges, it follows that no transversal T' of $\mathcal{G}'$ with $T' \subseteq \overline{T}$ can have fewer than

---

Algorithm Trans$(\mathcal{H})$

input : simple, nonempty hypergraph $\mathcal{H}$ on V.
output: $Tr(\mathcal{H})$ edge by edge, i.e. incrementally all minimal transversals of $\mathcal{H}$.

$r \leftarrow r(\mathcal{H})$; $\mathcal{G} \leftarrow \emptyset$;
loop / $*$ $\mathcal{G} \subseteq Tr(\mathcal{H})$ $*$ /
  if (there exists a $T \in Tr(\mathcal{G})|_r - \mathcal{H}$) **then**
    minimize $\overline{T}$ to a minimal transversal T_1 of $\mathcal{H}$;
  else / $*$ $\mathcal{H} \subseteq Tr(\mathcal{G}), \mathcal{G} \subseteq Tr(\mathcal{H})$ $*$ /
    if (there exists $\mathcal{G}' \subseteq \mathcal{G}$ with $|\mathcal{G}'| = r + 1$ such that
        for all $E \in \mathcal{G}$, $E \not\subseteq \{x \in V : d(x, \mathcal{G}') > 1\}$)
    then
      minimize $\{x \in V : d(x, \mathcal{G}') > 1\}$ to a minimal transversal T_1 of $\mathcal{H}$;
    else / $*$ $\mathcal{G} = Tr(\mathcal{H})$ $*$ /
      exit loop;
    fi;
  fi;
  output(T_1); $\mathcal{G} \leftarrow \mathcal{G} \cup \{T_1\}$;
endloop;

TABLE 1
Algorithm computing $Tr(\mathcal{H})$ in incremental-polynomial time if $r(\mathcal{H}) \leq k$, for constant $k \geq 0$.

$r + 1$ vertices. Check that every $H \in \mathcal{H}$ is a transversal of $\mathcal{G}'$; since $|H| \leq r$, it follows that $E \not\subseteq \overline{T}$, which is a contradiction. Thus T is a transversal of $\mathcal{H}$, and the claim is proved.

Since $T \notin Cov(\mathcal{G})$, T contains some $T_1 \in Tr(\mathcal{H})$ such that $T_1 \notin Cov(\mathcal{G})$; in fact, every $T_1 \in Tr(\mathcal{H})$ such that $T_1 \subseteq T$ has this property.

Utilizing this result, the algorithm **Trans** in Table 1 incrementally outputs the minimal transversals of a simple, nonempty hypergraph $\mathcal{H}$. The correctness of **Trans** can be easily proved from (1) and the described additional transversal determination; note that $\mathcal{G} \subseteq Tr(\mathcal{H})$ is a loop invariant.

To prove the theorem, assume that $r(\mathcal{H}) \leq k$. If $\mathcal{H}$ is not simple, compute $\min(\mathcal{H})$ in polynomial time and proceed with that hypergraph. Since k is a constant, the conditions of both **if** statements can be checked in time polynomial in the size of $\mathcal{G}$,$\mathcal{H}$ and V. n^k is an upper bound on the number of vertex sets to check in the outer **if**, and $|\mathcal{G}|^{k+1}$ is an upper bound for the number of subhypergraphs $\mathcal{G}' \subseteq \mathcal{G}$ (each of them computable in polynomial time) to test in the inner **if**. In both cases, all tests are clearly polynomial. Furthermore, T_1 is computable in polynomial time (cf. Section 2). Thus an execution of the loop body needs time polynomial in the size of $\mathcal{G}, \mathcal{H}$ and n. Every pass of the loop but the last yields an additional minimal transversal T_1 of $\mathcal{H}$ for output. The result follows. □

The following result is an immediate consequence of the last theorem.

COROLLARY 5.3. TRANS-HYP *is polynomial if* $r(\mathcal{H}) \leq k$ *or* $r(\mathcal{G}) \leq k$, *for fixed* $k \geq 0$.

Note that Theorem 5.2 generalizes the well-known result that the maximal independent sets of a graph are output-efficiently computable to hypergraphs with edge-size bounded by a constant. (Recall that a maximal independent set is the complement of a minimal transversal). Thus the question in [29], whether there is an output-polynomial total time algorithm for generating all maximal independent sets of a hypergraph, is affirmatively answered for hypergraphs of bounded edge-size.

Let us now consider a constant upper bound on the edge size. In this case the computation of the transversal hypergraph is efficiently possible even in the input size.

THEOREM 5.4. *If* $ar(\mathcal{H}) \geq n - k$, *for fixed* $k \geq 0$, *then* $Tr(\mathcal{H})$ *is computable in input-polynomial time.*

Proof. Without loss of generality we assume that $\mathcal{H}$ is simple. We know from Proposition 4.7(iii) that $Tr(\mathcal{H}) \leq \overline{\delta(\mathcal{H})}$. Since $r(\overline{\delta(\mathcal{H})}) \leq k + 1$, only the sets $V' \subseteq V$ of up to $k + 1$ elements are candidates for minimal transversals, and there are no more than roughly n^{k+1} of them. □

From Theorems 5.4 and 4.8, we immediately have the following result.

COROLLARY 5.5. TRANS-HYP *and* SIMPLE-H-SAT *are polynomial if* $ar(\mathcal{H}) \geq n - k$, *for fixed* $k \geq 0$.

As the third restriction, let us consider hypergraphs where the edges differ in their size by at most some given constant. We note a simple, but important lemma.

LEMMA 5.6. *Let* $\mathcal{H}$ *be a hypergraph on* V, *and* $a = ar(\mathcal{H})$, $r = r(\mathcal{H})$. *Then* $\mathcal{H}$ *is not saturated if and only if there exists a set* $V' \subseteq V$, $a \leq |V'| \leq r$, *such that* $V' \notin Cov(\mathcal{H})$.

Proof. Assume there exists $V' \subseteq V$, such that $V' \notin Cov(\mathcal{H})$. If $|V'| < a$, we may add any vertices $v_1, \ldots, v_{a-|V'|}$ from $V - V'$ to V', and $V' \cup \{v_1, \ldots, v_{a-|V'|}\} \notin Cov(\mathcal{H})$ will hold. If $|V'| > r$, we may remove any vertices $v_1, \ldots, v_{|V'|-r}$ from V' without establishing $V' - \{v_1, \ldots, v_{|V'|-r}\} \in Cov(\mathcal{H})$. Thus the *only if* direction holds. The *if* direction is trivial. □

THEOREM 5.7.

H-SAT *is polynomial if* $r(\mathcal{H}) - ar(\mathcal{H}) \leq k$, *for fixed* $k \geq 0$.

Proof. Let $a = ar(\mathcal{H})$, and $r = r(\mathcal{H})$. There is a simple saturation algorithm which is polynomial for this restriction: Check for each $X \subseteq V$, $|X| = a, a + 1, \ldots, r$, whether $X \notin Cov(\mathcal{H})$, and output "no" if the first such X is found and "yes" if none is found. By Lemma 5.6, this algorithm is correct.

Let us analyze the complexity of this algorithm. Consider $Y \in \mathcal{H}$. The exact number of the sets X covered by Y is given by

$$C(Y,a,r) = \sum_{i=0}^{r-|Y|} \binom{n-|Y|}{i} + \sum_{i=0}^{|Y|-a} \binom{|Y|}{i} - 1, \tag{2}$$

where the first (resp. second) term on the right hand side of the equation equals the number of the sets X covered as supersets (resp. subsets). The number $C(Y,a,r)$ is bounded by the sum of the number of sets $X \subseteq V$ with $|X| \leq r$ that a fixed $Y' \subseteq V$ with $|Y'| = a$ covers as superset and the number of sets $X \subseteq V$ with $|X| \geq a$ that a fixed $Y' \subseteq V$ with $|Y'| = r$ covers as subset. Let $c = r - a + 1$. Thus,

$$\begin{aligned} C(Y,a,r) &\leq \sum_{i=0}^{r-a} \left[\binom{n-a}{i} + \binom{r}{i} \right] \\ &\leq 2c \cdot \max \left\{ \binom{n-a}{i}, \binom{r}{i} : 0 \leq i \leq c-1 \right\} \\ &\leq 2cn^{c-1}. \end{aligned}$$

The last inequation uses that $u \leq v$ implies $\binom{u}{w} \leq \binom{v}{w}$ and that $\binom{u}{w} \leq u^w$. As $c \leq k+1$, we have $C(Y,a,r) \leq s(n)$ for some polynomial $s(n)$. Hence, $\mathcal{H}$ covers at most $m \cdot s(n)$ of the sets X.

Note that the subsets of V of size i can be systematically output with polynomial-time delay $p(n)$ (cf. [12] for a suitable order). Moreover, testing $X \in Cov(\mathcal{H})$ can be done in polynomial time $q(m,n)$.

Hence if $\mathcal{H}$ is not saturated, the algorithm finds the first X such that $X \notin Cov(\mathcal{H})$ in time $O(m \cdot s(n)(p(n)+q(m,n)))$, i.e., in polynomial time. If $\mathcal{H}$ is saturated, then all sets X are checked. Let $f(n,a,r) = \binom{n}{a} + \binom{n}{a+1} + \cdots + \binom{n}{r}$ denote their number. Since $\mathcal{H}$ is saturated, it covers each of the sets X. On the other hand, $\mathcal{H}$ covers at most $m \cdot s(n)$ of the sets X; consequently, $f(n,a,r) \leq m \cdot s(n)$. This ensures that the algorithm again terminates in time $O(m \cdot s(n)(p(n)+q(m,n)))$. □

COROLLARY 5.8 (to Theorem 5.7). SIMPLE-H-SAT *is polynomial if* $r(\mathcal{H}) - ar(\mathcal{H}) \leq k$, *for fixed* $k \geq 0$.

5.2. Acyclic hypergraphs. Like in graph theory, the notion of acyclicity is appealing in hypergraph theory from a theoretical as well as a practical point of view. In some contexts, acyclic hypergraphs gain special attention. E.g., in relational database theory acyclic hypergraphs (also called *tree schemes*) were introduced in [8] and later used by many authors (cf. [53, 30]), especially in relational database design, cf. [2, 4, 5, 17].

Several **NP**-complete problems on hypergraphs become polynomial for acyclic hypergraphs [57]. Since it is not straightforward to carry over the definition of a cycle from graphs to hypergraphs, there are many notions of acyclicity in a hypergraph, cf. [16, 18, 7]. We refer to α-, β-, γ-, and *Berge*-acyclicity as stated in [16], where the proper inclusion hierarchy *Berge*-acyclic $\Rightarrow$ γ-acyclic $\Rightarrow$ β-acyclic $\Rightarrow$ α-acyclic is proved.

The notion of α-acyclicity came up in the context of relational database theory [23, 58, 5, 16]. A hypergraph $\mathcal{H}$ is *α-acyclic* iff $\mathcal{H} = \emptyset$ or $\mathcal{H}$ is by the GYO-reduction (due to Graham [23] and (independently) to Yu and Ozsoyoglu [58]), that is by repeated application of one of the two rules:

(1) If vertex v occurs in only one edge E, remove v from E;
(2) If distinct edges E, E' satisfy $E' \subseteq E$, remove E',

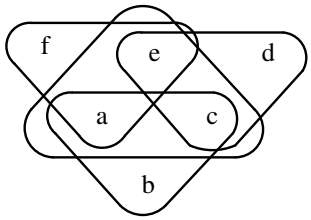

FIG. 1. *α-acyclic, but β-cyclic hypergraph* $\mathcal{H} = \{\{a, b, c\}, \{c, d, e\}, \{a, e, f\}, \{a, c, e\}\}$.

reducible to the hypergraph $\{\emptyset\}$.

For example, the hypergraph in Figure 1 is α-acyclic. Notice that α-acyclicity of a hypergraph $\mathcal{H}$ can be checked with an algorithm of Tarjan and Yannakakis [50] in time $O(n+t)$, where n is the number of vertices and t is the total size of the edges of $\mathcal{H}$.

Note that rule 2 implies that a hypergraph $\mathcal{H}$ is α-acyclic if and only if $\max(\mathcal{H})$ is α-acyclic. Consequently, an α-acyclic hypergraph $\mathcal{H}$ may contain an α-cyclic subhypergraph $\mathcal{H}' \subseteq \mathcal{H}$. For example, let in Figure 1 be $\mathcal{H}' = \{\{a, b, c\}, \{c, d, e\}, \{a, e, f\}\}$. However, this is not what one expects from acyclicity, as it is surprising that a cycle disappears by adding edges. Fagin points out this anomaly [16] and introduces the more natural concept of β-acyclicity.

DEFINITION 5.2. *A hypergraph $\mathcal{H}$ is β-acyclic iff every subhypergraph $\mathcal{H}' \subseteq \mathcal{H}$ is α-acyclic.*

The hypergraph in Figure 1 is β-cyclic, i.e., not β-acyclic. Fagin presents in [16] various equivalent definitions of β-acyclicity, among them a related acyclicity criterion by Graham [23], and he also gives a polynomial-time algorithm to test β-acyclicity.

Our next aim is to show that the transversal hypergraph of a β-acyclic hypergraph is efficiently computable with respect to the input and the output size.

DEFINITION 5.3. *Let $\mathcal{H}$ be a hypergraph on V. For every $V' \subseteq V$, the partial hypergraph of $\mathcal{H}$ generated by V' is $\mathcal{H}_{V'} = \{E \cap V' | E \in \mathcal{H}\}$.*

For convenience, let us call any vertex an *ear node* of the hypergraph $\mathcal{H}$ if it occurs in exactly one edge of $\mathcal{H}$ (cf. [53, p.698]). We observe the following rule for transversal computation.

LEMMA 5.9. *Let $\mathcal{H}$ be a simple hypergraph on V with an ear node v that occurs in edge E. Then $\{\{v\} \cup T : T \in Tr(\mathcal{H}_{\overline{E}} - \{\emptyset\})\} \cup Tr(\mathcal{H}_{V-\{v\}})$ is a partitioning of $Tr(\mathcal{H})$.*

Proof. If $\mathcal{H}$ is simple, then every essential vertex $v \in V$ occurs in at least one minimal transversal of $\mathcal{H}$. (This is readily shown from Proposition 2.2.) However, $Tr(\mathcal{H}_{\overline{E}} - \{\emptyset\})$ is never empty. Now consider $Tr(\mathcal{H})$. The minimal transversals can be partitioned into $Tr(\mathcal{H}) = \mathcal{T}_1 \cup \mathcal{T}_2$, where $\mathcal{T}_1$ contains all minimal transversals that contain v and $\mathcal{T}_2$ all others. Since $Tr(Tr(\mathcal{H})) = \mathcal{H}$ if $\mathcal{H}$ is simple (Proposition 2.2), E is a minimal transversal of $Tr(\mathcal{H})$. It follows that $E \cap T = \{v\}$ for every $T \in Tr(\mathcal{H})$ that contains v. Indeed, if $\{v\} \subset E \cap T$ would hold, then, since E is the only edge of $\mathcal{H}$ that contains v, $T - \{v\}$ would be a transversal of $\mathcal{H}$, which contradicts $T \in Tr(\mathcal{H})$. Thus, $E \cap T = \{v\}$. This implies $T - \{v\} \subseteq \overline{E}$. From this it is clear that $\mathcal{T}_1 = \{\{v\} \cup T : T \in Tr(\mathcal{H}_{\overline{E}} - \{\emptyset\})\}$. It is immediate that $\mathcal{T}_2 = Tr(\mathcal{H}_{V-\{v\}})$. □

THEOREM 5.10. *The minimal transversals of a β-acyclic hypergraph $\mathcal{H}$ are P-enumerable.*

Proof. Recall that a problem is P-enumerable if there is some algorithm which computes all solutions to the problem in time $p(IS)N$, where p is some polynomial in the input size IS and N is the number of solutions. To include the case in which the problem has *no* solution,

we slightly modify this convention from $p(IS)N$ to $p(IS)(N+1)$; this modification is not substantial.

To prove the theorem, we proceed as follows: first we note some facts regarding β-acyclic hypergraphs, and then we give an algorithm for transversal computation. We may assume without loss of generality that $\mathcal{H}$ has no inessential vertices and that $\mathcal{H}$ is simple, since $\min(\mathcal{H})$ is polynomial-time computable, $Tr(\mathcal{H}) = Tr(\min(\mathcal{H}))$, and $\min(\mathcal{H})$ is β-acyclic if $\mathcal{H}$ is β-acyclic.

There is a simple observation on ear nodes of β-acyclic hypergraphs.

FACT 1: *Every simple β-acyclic hypergraph $\mathcal{H}$ that contains a nonempty edge has an ear node.* Indeed, if $\mathcal{H}$ is β-acyclic, then it is also α-acyclic, and the GYO-reduction must succeed. However, as $\mathcal{H}$ is simple, rule 2 is not applicable, and as $\mathcal{H} \neq \{\emptyset\}$, rule 1 must apply. Hence $\mathcal{H}$ has an ear node.

FACT 2: *Let $\mathcal{H}$ be a β-acyclic hypergraph and let $V' \subseteq V$. Then $\mathcal{H}_{V'}$ is β-acyclic.* (See [16, p. 530].)

Algorithm BetaTr$(\mathcal{H})$

input : Simple β-acyclic hypergraph $\mathcal{H}$.
output: $Tr(\mathcal{H})$.

if $\mathcal{H} = \emptyset$ **then return**$(\{\emptyset\})$
else
  if $\mathcal{H} = \{\emptyset\}$ **then return**$(\emptyset)$
  else
    $v \leftarrow$ an ear node in $E \in \mathcal{H}$**;**
    $\mathcal{H}_1 \leftarrow \min(\mathcal{H}_{\overline{E}} - \{\emptyset\})$**;** $\mathcal{H}_2 \leftarrow \min(\mathcal{H}_{V-\{v\}})$**;**
    $\mathcal{T}_1 \leftarrow$ **BetaTr**$(\mathcal{H}_1)$**;** $\mathcal{T}_2 \leftarrow$ **BetaTr**$(\mathcal{H}_2)$**;**
    return$(\{T \cup \{v\} : T \in \mathcal{T}_1\} \cup \mathcal{T}_2)$
  fi;
fi;

TABLE 2
Output-polynomial algorithm for computing $Tr(\mathcal{H})$ of a simple, β-acyclic hypergraph $\mathcal{H}$.

Now consider the procedure $\mathbf{BetaTr}$ in Table 2, where the input hypergraph $\mathcal{H}$ is simple. Let for every hypergraph $\mathcal{G}$ denote $v_e(\mathcal{G}) = |V_e(\mathcal{G})|$ in what follows.

The correctness of $\mathbf{BetaTr}$ is shown by induction on $v_e(\mathcal{H})$. If $v_e(\mathcal{H}) = 0$, then $\mathcal{H} = \emptyset$ or $\mathcal{H} = \{\emptyset\}$, and the result is correct. Now consider $v_e(\mathcal{H}) > 0$ and assume the hypothesis is correct for simple hypergraphs with fewer essential nodes than $\mathcal{H}$. As $\mathcal{H}$ is simple, β-acyclic, and $v_e(\mathcal{H}) > 0$, by Fact 1 $\mathcal{H}$ has an ear node; hence, a vertex v in edge E will be found. From Lemma 5.9, we have

$$Tr(\mathcal{H}) = \{\{v\} \cup T : T \in Tr(\mathcal{H}_{\overline{E}} - \{\emptyset\})\} \cup Tr(\mathcal{H}_{V-\{v\}}),$$

which is equivalent to

$$Tr(\mathcal{H}) = \{\{v\} \cup T : T \in Tr(\min(\mathcal{H}_{\overline{E}} - \{\emptyset\}))\} \cup Tr(\min(\mathcal{H}_{V-\{v\}})).$$

Since a hypergraph is β-acyclic iff every subhypergraph of it is β-acyclic, in conjunction with Fact 2 it follows that $\mathcal{H}_1$ and $\mathcal{H}_2$ are simple, β-acyclic hypergraphs. Because $\mathcal{H}_1$ and $\mathcal{H}_2$ have fewer essential vertices than $\mathcal{H}$, by the induction hypothesis we have $Tr(\mathcal{H}_i) = \mathbf{BetaTr}(\mathcal{H}_i) = \mathcal{T}_i$, for $i = 1, 2$. Thus

$$Tr(\mathcal{H}) = \{\{v\} \cup T : T \in Tr(\mathcal{H}_{\overline{E}} - \{\emptyset\})\} \cup Tr(\mathcal{H}_{V-\{v\}}) = \{\{v\} \cup T : T \in \mathcal{T}_1\} \cup \mathcal{T}_2.$$

This is exactly what $\mathbf{BetaTr}(\mathcal{H})$ returns if $V_e(\mathcal{H}) \neq \emptyset$; hence, the claimed statement holds for $\mathcal{H}$.

It remains to prove the claim on the complexity of **BetaTr**. Denote by $calls(\mathcal{H})$ the total number of recursive calls to $\mathbf{BetaTr}$ in computing $\mathbf{BetaTr}(\mathcal{H})$. We show by induction on $v_e(\mathcal{H})$ that

$$calls(\mathcal{H}) \leq 2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H})|.$$

For $v_e(\mathcal{H}) = 0$, we have either $\mathcal{H} = \emptyset$ or $\mathcal{H} = \{\emptyset\}$; the statement clearly holds. Now consider the case $v_e(\mathcal{H}) > 0$ and assume the statement is correct for hypergraphs with fewer essential vertices. We have that

$$calls(\mathcal{H}) = calls(\mathcal{H}_1) + calls(\mathcal{H}_2) + 2.$$

By the induction hypothesis, we obtain that

$$calls(\mathcal{H}_1) \leq 2v_e(\mathcal{H}_1) \cdot |Tr(\mathcal{H}_1)| \leq 2(v_e(\mathcal{H}) - 1) \cdot |Tr(\mathcal{H}_1)| \leq 2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H}_1)| - 2;$$

the last inequation holds because $\mathcal{H}_1 \neq \{\emptyset\}$, and therefore $|Tr(\mathcal{H}_1)| > 0$. Similarly, we obtain by the induction hypothesis that

$$calls(\mathcal{H}_2) \leq 2v_e(\mathcal{H}_2) \cdot |Tr(\mathcal{H}_2)| \leq 2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H}_2)|.$$

Thus

$$\begin{aligned} calls(\mathcal{H}) &= calls(\mathcal{H}_1) + calls(\mathcal{H}_2) + 2 \\ &\leq 2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H}_1)| - 2 + 2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H}_2)| + 2 \\ &= 2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H}_1)| + 2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H}_2)|. \end{aligned}$$

Now, since $|Tr(\mathcal{H})| = |Tr(\mathcal{H}_1)| + |Tr(\mathcal{H}_2)|$, it follows that $calls(\mathcal{H}) \leq 2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H})|$, i.e., the claimed statement holds.

The body of $\mathbf{BetaTr}$ can be clearly executed in polynomial time. Thus, the total runtime of $\mathbf{BetaTr}(\mathcal{H})$ is bounded by $p(m, n) \cdot (calls(\mathcal{H}) + 1)$ for some polynomial p in m and n (i.e., the number of edges and vertices of $\mathcal{H}$). Since

$$p(m, n) \cdot (calls(\mathcal{H}) + 1) \leq p(m, n) \cdot (2v_e(\mathcal{H}) \cdot |Tr(\mathcal{H})| + 1),$$

it follows that the runtime of $\mathbf{BetaTr}(\mathcal{H})$ is bounded by $q(m, n) \cdot (|Tr(\mathcal{H})| + 1)$ for some polynomial q. Thus, $\mathbf{BetaTr}(\mathcal{H})$ P-enumerates the minimal transversals of $\mathcal{H}$. The result follows. □

We remark that it is not difficult to modify **BetaTr** such that the minimal transversals are output with polynomial-time delay; we leave this to the reader.

From Theorem 5.10, we obtain together with Theorems 4.5 and 4.12 immediately the following polynomial cases of the transversal problem and the saturation problem.

COROLLARY 5.11. TRANS-HYP *and* SIMPLE-H-SAT *are polynomial for* β*-acyclic hypergraphs.*

6. Overview of related problems and applications. In this section we give a short overview of some related problems and applications in different fields of computer science. For space reasons the exposition is rather succinct. In particular, we omit several formal definitions and all proofs. A full discussion, exact definitions, all proofs, and more material can be found in the extended report [15]; other interesting problems related to TRANS-HYP have recently been studied in [32].

6.1. Clause satisfiability. We have identified the following restrictions of the well-known SATISFIABILITY problem which are $\leq_m^p$-equivalent to TRANS-HYP (resp. co-TRANS-HYP).

Problem: INTERSECTING MONOTONE SAT (IMSAT)
Instance: A set $\mathcal{C}$ of clauses such that each clause is either positive (i.e., consists entirely of positive literals) or negative (i.e., consists entirely of negative literals) and for each positive clause C_1 and negative clause C_2 of $\mathcal{C}$, there exists an atom u such that $u \in C_1$ and $\neg u \in C_2$.
Question: Is $\mathcal{C}$ satisfiable ?

Problem: SYMMETRIC INTERSECTING MONOTONE SAT (SIMSAT)
Instance: Restriction of IMSAT to instances $\mathcal{C}$ where the negative clauses are precisely all clauses C^- such that $C^- = \{\neg u : u \in C^+\}$ for some positive clause $C^+ \in \mathcal{C}$. (By this restriction, nonempty positive clauses of $\mathcal{C}$ are mutually intersecting.)
Question: Is $\mathcal{C}$ satisfiable ?

Both problems are $\leq_m^p$-equivalent to co-SIMPLE-H-SAT. Interestingly, unlike SATISFIABILITY or most known other restrictions of SATISFIABILITY, these problems become polynomial-time decidable as soon as the cardinality of the clauses in the problem instances is bounded by a constant.

6.2. Design of relational databases. For the basic concepts of relational database theory consult [39] or [53]. If F is a set of functional dependencies (FDs) on a set of attributes U, then F^+ denotes the closure of F, i.e., all those dependencies on U that follow from F. Let $RS = \langle U, F\rangle$ be a relation schema, where U is a finite set of attributes and F is a set of functional dependencies. A relation instance R over U is an *Armstrong relation* for RS iff the functional dependencies that hold in R are precisely F^+. It was recently advocated that Armstrong relations can be used as a very profitable tool in database design [44, 40]. In this context it is important to compute an Armstrong relation from a given set of functional dependencies and vice-versa. The following related decision problem is $\leq_m^p$-equivalent to SIMPLE-H-SAT:

Problem: FD-RELATION EQUIVALENCE
Instance: A relation instance R and a set F of FDs in BCNF, both on a set of attributes U.
Question: Is R an Armstrong relation for F ?

A relation scheme $RS = \langle U, F\rangle$ is in *Boyce-Codd Normal Form* (BCNF) if for every $X \to Y \in F^+$ such that $Y \not\subseteq X$, X is a superkey for RS, i.e., $X \to U$.

By using Corollary 5.3 it can be shown that the problem FD-RELATION EQUIVALENCE becomes polynomial if F is in BCNF and for each $X \to Y$ of F, X contains fewer than k attributes for some constant k. Moreover, we can show that, under the same restrictions, the problem of *generating* an Armstrong relation for F can be done in output-polynomial time.

A relation instance R on U is in BCNF if the schema $\langle U, F_R\rangle$ is in BCNF, where F_R is the set of all FDs that hold on R. The following problem can be shown to be $\leq_m^p$-equivalent to co-SIMPLE-H-SAT:

Problem: ADDITIONAL KEY for relation instances
Instance: A relation instance R on attributes U, a set $\mathcal{K}$ of minimal keys for R.
Question: Is there a minimal key for the scheme $RS = \langle U, F_R\rangle$ not contained in $\mathcal{K}$?

Note that the additional key problem for relation *schemes* is polynomial-time solvable by an algorithm of Lucchesi and Osborn [37].

6.3. Updates in distributed databases. In distributed databases, mutual exclusion of groups of sites, which is necessary for executing critical operations, can be realized by defining a priori a set of groups (quorums) that intersect each other [33]. A group of sites can perform the critical operation only if it contains a quorum of this set. Clearly, it is natural to consider only minimal quorums, i.e., no quorum should properly contain any other quorum. In terms of hypergraph theory, this means that the specified quorums should constitute a simple, intersecting hypergraph on the set of sites, termed a *coterie* in [20]. For example, the hypergraph $\mathcal{C} = \{ \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\} \}$ is a coterie on vertices $\{a, b, c, d\}$. Coteries appear to be a fruitful concept for access control in distributed databases, cf. [20, 25, 19].

A coterie $\mathcal{A}$ *dominates* a coterie $\mathcal{B}$ if and only if $\mathcal{A} \neq \mathcal{B}$ and for each $B \in \mathcal{B}$ there exists an $A \in \mathcal{A}$ such that $B \supseteq A$. A coterie $\mathcal{A}$ is *nondominated* (*ND*) if there is no coterie $\mathcal{B}$ that dominates $\mathcal{B}$. In practice, ND coteries are desired for decision agreement groups with respect to reliability considerations. Unfortunately, no efficient test for nondomination of a coterie is known; in [20] merely an algorithm which has worst case run time exponential in the number of vertices is described. Interestingly, this problem turns out to be an instance of the SELFTRANSVERSALITY problem.

THEOREM 6.1. *A coterie $\mathcal{C}$ is ND if and only if $\mathcal{C} = Tr(\mathcal{C})$.*

Since every selftransversal hypergraph is a coterie, we get from Theorems 6.1 and 4.13 the following.

COROLLARY 6.2. *Checking if a coterie is ND* (ND COTERIE) *is $\leq_m^p$-equivalent to* SIMPLE-H-SAT.

Note that ND COTERIE is by Theorem 6.1 and Corollary 5.3 polynomial for small quorums.

In the extended report, we also show that for a recent generalization of this approach to bi- and semicoteries, which model read and write operations by read quorums and write quorums [19, 25], the complexity of testing nondomination is not increased and remains $\leq_m^p$-equivalent to TRANS-HYP.

6.4. Boolean switching theory. We assume that the reader is familiar with the basic concepts of switching circuits and Boolean functions (BFs); for details we refer to the standard literature (e.g., [56]). The design of efficient algorithms for the computation of prime implicants from a function table or a logical expression has been topic of research over decades.

It is well-known that every monotone Boolean function has a unique *minimal conjunctive normal form* consisting of a conjunction of disjunctions of atoms where no conjunct subsumes any other.

In [15] we show the following results. Given a monotone Boolean expression E and a set P of prime implicants of E, it is **NP**-complete to determine whether there exists an additional prime implicant of E. However, if E is in minimal conjunctive normal form, then the same problem is $\leq_m^p$-equivalent to co-SIMPLE-H-SAT. In particular, in the latter case, the prime implicants of E are precisely the minimal transversals of the conjuncts of E. Thus, the polynomial subcases for computing all minimal hypergraph transversals described in Section 5 carry over to the problem of computing the prime implicants of a monotone Boolean function in conjunctive normal form. This problem is also investigated in [9], which gives some new results.

6.5. Model-based diagnosis. Basic techniques for model-based diagnosis have been developed within AI by de Kleer and Williams [13], and by Reiter [45]. To introduce the necessary concepts briefly, a *system* is a pair $(SD, COMP)$, where SD, the *system description*, is a set of usually first-order sentences and $COMP$ is a set of constants which model the

system *components*. This general system description is used together with a set OBS of first-order sentences, which are particular *observations* on the system behavior, to diagnose faults. The system description SD makes use of a distinguished predicate $AB(c)$ which interprets "component c operates in abnormal mode". Now a *diagnosis* for $(SD, COMP, OBS)$ is a minimal set $\Delta \subseteq COMP$ of components such that

$$\mathcal{T} = SD \cup OBS \cup \{AB(c) \mid c \in \Delta\} \cup \{\neg AB(c) \mid c \in COMP - \Delta\}$$

is consistent.[4] Of course, if there is no fault, $\Delta = \emptyset$ must be a diagnosis, otherwise SD is not a sound system description.

In [45] a characterization of diagnoses in terms of so called conflict sets is given. A *conflict set* is a set $C \subseteq COMP$ such that $SD \cup OBS \cup \{\neg AB(c) \mid c \in C\}$ is inconsistent. A conflict set C is minimal if and only if no proper subset of C is a conflict set. The fundamental theorem on conflict sets and diagnoses in [45] put into hypergraph terminology is

THEOREM 6.3 ([45]). *$\Delta \subseteq COMP$ is a* diagnosis *for $(SD, COMP, OBS)$ if and only if $\Delta \in Tr(\mathcal{C})$, where $\mathcal{C}$ denotes the set of all minimal conflict sets of $(SD, COMP, OBS)$.*

From Theorems 6.3 and 4.12 we immediately have the following theorem:

THEOREM 6.4. *Let $\mathcal{C}$ be the set of minimal conflict sets of $(SD, COMP, OBS)$, and $\mathcal{D}$ a set of diagnoses for $(SD, COMP, OBS)$. Given $\mathcal{C}$ and $\mathcal{D}$ for input, deciding if there is an additional diagnosis not contained in $\mathcal{D}$ is $\leq_m^p$-equivalent to co-*SIMPLE-H-SAT.

The determination of diagnoses from the given minimal conflict sets is essential in popular algorithms for model-based diagnosis [45, 13]. Deciding if an already computed set of diagnoses is complete with respect to a given set of minimal conflict sets, i.e., consists of all diagnoses, is an important subproblem if diagnoses are computed incrementally. Therefore, the complexity of the additional diagnosis problem is of crucial interest.

7. Conclusion. We have studied two computational problems on hypergraphs in this paper, namely deciding saturation of a hypergraph (H-SAT) and the recognition of the transversal hypergraph (TRANS-HYP). The latter problem is closely related to computing all minimal transversals of a hypergraph.

The complexity of computing (resp. recognizing) all minimal transversals of a hypergraph is an open problem [20, 41, 14, 29]. We showed that checking saturation for a simple hypergraph (SIMPLE-H-SAT) is computationally equivalent to TRANS-HYP; without the restriction to simple hypergraphs, H-SAT was proved co-**NP**-complete in its general version as well as for various subcases.

In the study of SIMPLE-H-SAT and TRANS-HYP, we investigated the relationships of these problems to well-studied problems such as satisfiability and hypergraph 2-colorability. Several $\leq_m^p$-equivalent subproblems were exhibited, the most important among them SELF-TRANSVERSALITY. Moreover, narrowing the open problems "frontier", we showed that some generalizations of these two problems are intractable and that several important subcases are polynomial. Some of those results use algorithms that compute under certain restrictions all minimal transversals of a hypergraph in output-polynomial total time; the probably most important restriction is a constant upper bound on the edge size.

Our results apply to various problems in database theory, switching theory, logic, and artificial intelligence which are all closely related to SIMPLE-H-SAT and TRANS-HYP.

For future research, we present the following open questions:

1. What is the complexity of SIMPLE-H-SAT and TRANS-HYP ? In particular, are these problems co-**NP**-complete, or can they be solved in polynomial time (or, less restrictive, in

[4] It is presupposed that $\mathcal{T}$ is always in a decidable subclass of first-order predicate calculus.

nondeterministic polynomial time) ? In connection with this, are the minimal transversals of a hypergraph $\mathcal{H}$ computable in output-polynomial total time ?

2. How do SIMPLE-H-SAT and TRANS-HYP relate to other open problems in **NP**-completeness (cf. [28]) ? The most famous such problem, GRAPH ISOMORPHISM, has been studied extensively in the literature. It is known that if GRAPH ISOMORPHISM is **NP**-complete and $\mathbf{P} \neq \mathbf{NP}$, then the polynomial hierarchy collapses at level two [48], which is not expected by the experts [27]. However, there seems to be no trivial relation between GRAPH ISOMORPHISM on the one hand and SIMPLE-H-SAT and TRANS-HYP on the other.

3. What is the gain by using randomized algorithms, probabilistic algorithms, or interactive proof systems (cf. [26, 27]) for SIMPLE-H-SAT and TRANS-HYP, and what about weaker forms of reductions than polynomial transformability (especially, randomized reductions, cf. [1, 47, 55])? Schöning [47] defines within **NP** a Low Hierarchy $L_0, L_1, \ldots$, where $L_0 = \mathbf{P}$, $L_1 = \mathbf{NP} \cap \text{co-}\mathbf{NP}$, and a High Hierarchy $H_0, H_1, \ldots$, where $H_0 = \mathbf{NP}$-complete, and the other classes $H_i, i > 0$, correspond to weakened versions of **NP**-completeness such as γ-completeness [1] (fully contained in H_1) etc. which are believed still strong enough that no problem in **P** satisfies one of them. Classifying co-SIMPLE-H-SAT into the Low or High Hierarchy would give strong evidence that the problem is not **NP**-complete or intractable, respectively.

Acknowledgments. The authors thank L. Lovász and M. Yannakakis for interesting discussions of this work and valuable comments. They would further like to thank the referees of previous versions of this paper, as well as H. Mannila and D. Plaisted, for their comments and helpful suggestions.

Note added in proof. (1) In a recent paper, Fredman and Khachiyan showed that $Tr(\mathcal{H})$ can be recognized in time $O(n^{\log n})$ ("On the Complexity of Dualization of Monotone Disjunctive Normal Forms," Technical Report LCS-TR-225, Dept. of Computer Science, Rutgers University, 1994). This implies that all problems that are $\leq_m^p$-equivalent to TRANS-HYP can be recognized within the same time bound. Moreover, it strongly suggests that all these problems are not co-**NP**-complete and provides partial evidence to the thesis that TRANSHYP is close to the border between polynomiality and co-**NP**-hardness.

(2) As pointed out by V. Gurvich, $\leq_m^p$-equivalence between TRANSHYP and SELFTRANSVERALITY (Theorem 4.13) was earlier proved by P.D. Seymour in the context of Boolean functions in his masters thesis (see p. 309 of "On the two-coloring of hypergraphs," Quart. J. Math. Oxford (3) 25, (1974), pp. 303–312).

REFERENCES

[1] L. Adleman and K. Manders, *Reducibility, randomness, and intractability*, in Proceedings of the 9^{th} ACM Symposium on the Theory of Computing, 1977, pp. 151–163.

[2] C. Batini, A. D'Atri, and M. Moscarini, *Formal tools for top-down and bottom-up generation of acyclic relational database schemata*, in Proceedings 7^{th} Int. Conf. on Graph-Theoretic Concepts in Computer Science, Linz, Austria 1981.

[3] C. Beeri, M. Down, R. Fagin, and R. Statman, *On the structure of Armstrong relations for functional dependencies*, Journal of the ACM, 31 (1984), pp. 30–46.

[4] C. Beeri, R. Fagin, D. Maier, A. Mendelzon, J. Ullman, and M. Yannakakis, *Properties of acyclic database schemata*, in Proceedings 13^{th} ACM Symposium on the Theory of Computing, May 1981 1981, pp. 355–362.

[5] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis, *On the desirability of acyclic database schemes*, Journal of the ACM, 30 (1983), pp. 479–513.

[6] C. Benzaken, *Critical hypergraphs for the weak chromatic number*, Journal of Combinatorial Theory, Series B, 29 (1980), pp. 328–338.

[7] C. Berge, *Hypergraphs*, vol. 45 of North Holland Mathematical Library, Elsevier Science Publishers B.V. (North-Holland), 1989.

[8] P. A. Bernstein and D. W. Chiu, *Using semijoins to solve relational queries*, Journal of the ACM, 28 (1981), pp. 25–40.
[9] C. Bioch and T. Ibaraki, *Complexity of dualization and identification of positive boolean functions*, tech. report, Erasmus University Rotterdam, 1994.
[10] G. Burosch, J. Demetrovics, and G. O. H. Katona, *The poset of closures as a model of changing databases*, Order, 4 (1987), pp. 127–142.
[11] S. Cook, *A taxonomy of problems with fast parallel algorithms*, Information and Control, 64 (1985), pp. 2–22.
[12] D. E. Daykin, J. Godfrey, and A. J. W. Hilton, *Existence theorems for Sperner families*, Journal of Combinatorial Theory, Series A, 17 (1974), pp. 245–251.
[13] J. de Kleer and B. C. Williams, *Diagnosing multiple faults*, Artificial Intelligence, 32 (1987), pp. 97–130.
[14] J. Demetrovics and V. D. Thi, *Keys, antikeys and prime attributes*, Annales Univ. Sci. Budapest, Sect. Comp., 8 (1987), pp. 35–52.
[15] T. Eiter and G. Gottlob, *Identifying the minimal transversals of a hypergraph and related problems*, Tech. Report CD-TR 91/16, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, January 1991.
[16] R. Fagin, *Degrees of acyclicity for hypergraphs and relational database schemes*, Journal of the ACM, 30 (1983), pp. 514–550.
[17] R. Fagin, A. Mendelzon, and J. Ullman, *A simplified universal relation assumption and its properties*, ACM Trans. on Database Syst., 7 (1982), pp. 343–360.
[18] J.-C. Fournier and M. L. Vergnas, *A class of bichromatic hypergraphs*, Annals of Discrete Mathematics, 21 (1984), pp. 21–27.
[19] A. Fu, *Enhancing Concurrency and Availability for Database Systems*, PhD thesis, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6, 1990.
[20] H. Garcia-Molina and D. Barbara, *How to assign votes in a distributed system*, Journal of the ACM, 32 (1985), pp. 841–860.
[21] M. Garey and D. S. Johnson, *Computers and Intractability – A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
[22] M. R. Garey, D. S. Johnson, and L. Stockmeyer, *Some simplified NP-complete graph problems*, Theoretical Computer Science, 1 (1976), pp. 237–267.
[23] M. Graham, *On the universal relation*, tech. report, Univ. of Toronto, Toronto, Ont., Can., September 1979.
[24] R. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison Wesley, 1989.
[25] T. Ibaraki and T. Kameda, *Theory of coteries*, Tech. Report CSS/LCCR TR90–09, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6, August 1990. To appear in IEEE Transactions on Parallel and Distributed Systems.
[26] D. S. Johnson, *The NP-completeness column – an ongoing guide*, Journal of Algorithms, 5 (1984), pp. 147–160. Column 10: Updates, Updates, Updates.
[27] ———, *The NP-completeness column – an ongoing guide*, Journal of Algorithms, 9 (1988), pp. 426–444. Column 21: Interactive Proof Systems for Fun and Profit.
[28] ———, *The NP-completeness column – an ongoing guide*, Journal of Algorithms, 11 (1989), pp. 144–151. Column 22: The Story So Far.
[29] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, *On generating all maximal independent sets*, Information Processing Letters, 27 (1988), pp. 119–123.
[30] P. Kanellakis, *Elements of Relational Database Theory*, in Handbook of Theoretical Computer Science, J. van Leeuwen, ed., vol. B, Elsevier Science Publishers B.V. (North-Holland), 1990, ch. 17.
[31] R. Karp and A. Wigderson, *A fast parallel algorithm for the maximal independent set problem*, Journal of the ACM, 32 (1986), pp. 762–773.
[32] D. Kavvadias, C. Papadimitriou, and M. Sideri, *On Horn envelopes and hypergraph transversals*, in Proceedings 4th International Symposium on Algorithms and Computation (ISAAC-93), W. Ng, ed., LNCS 762, Hong Kong, December 1993, Springer.
[33] L. Lamport, *The implementation of reliable distributed multiprocess systems*, Comp. Networks, 2 (1978), pp. 95–114.
[34] E. Lawler, J. Lenstra, and A. Rinnooy Kan, *Generating all maximal independent sets: NP-hardness and polynomial-time algorithms*, SIAM J. Comp., 9 (1980), pp. 558–565.
[35] E. L. Lawler, *Covering problems: duality relations and a new method of solution*, SIAM J. Appl. Math., 14 (1966), pp. 1115–1132.
[36] L. Lovász, *Coverings and colorings of hypergraphs*, in Proceedings of the 4th Conference on Combinatorics, Graph Theory, and Computing, 1973, pp. 3–12.
[37] C. L. Lucchesi and S. Osborn, *Candidate keys for relations*, Journal of Computer and System Sciences, 17 (1978), pp. 270–279.
[38] K. Maghout, *Sur la détermination des nombres de stabilité et du nombre chromatique d'un graphe*, in Comptes Rendus des Séances de l'Académie des Sciences, vol. 248, Paris, Gauthier-Villars, 1959, pp. 3522–3523.
[39] D. Maier, *The Theory of Relational Databases*, Computer Science Press, Rockville, Md., 1983.

[40] H. Mannila and K.-J. Räihä, *Design by example: an application of Armstrong relations*, Journal of Computer and System Sciences, 22 (1986), pp. 126–141.
[41] ———, *Dependency inference*, in Proceedings of the 13^{th} VLDB, 1987, pp. 155–158.
[42] ———, *Algorithms for inferring functional dependencies*, Tech. Report A-1988-3, University of Tampere, Dept. of Comp. Sc., Series of Publ. A, April 1988.
[43] ———, *Generating Armstrong databases for sets of functional and inclusion dependencies*. manuscript, August 1988.
[44] ———, *Practical algorithms for finding prime attributes and testing normal forms*, in Proceedings of the 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 1989, pp. 128–133.
[45] R. Reiter, *A theory of diagnosis from first principles*, Artificial Intelligence, 32 (1987), pp. 57–95.
[46] B. Roy, *Algébre moderne et Thèorie des graphes*, vol. II, Dunod, Paris, 1970.
[47] U. Schöning, *A low and a high hierarchy within NP*, Journal of Computer and System Sciences, 27 (1983), pp. 14–28.
[48] ———, *Graph isomorphism is in the Low Hierarchy*, Journal of Computer and System Sciences, 37 (1988), pp. 312–323.
[49] E. Sperner, *Ein Satz über Untermengen einer endlichen Menge*, Mathematische Zeitschrift, XXVII (1928), pp. 544–548.
[50] R. E. Tarjan and M. Yannakakis, *Simple linear time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comp., 13 (1984), pp. 566–579.
[51] V. D. Thi, *Minimal keys and antikeys*, Acta Cybernetica, 7 (1986), pp. 361–371.
[52] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa, *A new algorithm for generating all maximal independent sets*, SIAM J. Comp., 6 (1977), pp. 505–517.
[53] J. D. Ullman, *Principles of Database and Knowledge Base Systems*, vol. 2, Computer Science Press, 1988.
[54] L. G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science, 8 (1979), pp. 189–201.
[55] L. G. Valiant and V. V. Vazirani, *NP is as easy as detecting unique solutions*, Theoretical Computer Science, 47 (1986), pp. 85–93.
[56] I. Wegener, *The Complexity of Boolean Functions*, Wiley - Teubner series in computer science, Wiley & Sons Ltd and Teubner, Stuttgart, 1987.
[57] M. Yannakakis, *Algorithms for acyclic database schemes*, in Proceedings 7^{th} Conf. on Very Large Database Systems, Cannes, France 1982, pp. 82–94.
[58] C. T. Yu and M. Ozsoyoglu, *An algorithm for tree-query membership of a distributed query*, in Proceedings IEEE COMPSAC, 1979, pp. 306–312.