Hypergraph Transversal Computation and Related Problems in Logic and AI

Thomas Eiter and Georg Gottlob

Institut für Informationssysteme, Technische Universität Wien
Favoritenstraße 9–11, A-1040 Wien, Austria
eiter@kr.tuwien.ac.at, gottlob@dbai.tuwien.ac.at

Abstract. Generating minimal transversals of a hypergraph is an important problem which has many applications in Computer Science. In the present paper, we address this problem and its decisional variant, i.e., the recognition of the transversal hypergraph for another hypergraph. We survey some results on problems which are known to be related to computing the transversal hypergraph, where we focus on problems in propositional Logic and AI. Some of the results have been established already some time ago, and were announced but their derivation was not widely disseminated. We then address recent developments on the computational complexity of computing resp. recognizing the transversal hypergraph. The precise complexity of these problems is not known to date, and is in fact open for more than 20 years now.

1 Introduction

A *hypergraph* $\mathcal{H} = (V, E)$ consists of a finite collection E of sets over a finite set V. The elements of E are called *hyperedges*, or simply *edges*. A *transversal* (or *hitting set*) of $\mathcal{H}$ is a set $T \subseteq V$ that meets every edge of E. A transversal is *minimal*, if it does not contain any other transversal as a subset. The set T of all minimal transversals of $\mathcal{H} = (V, E)$, constitutes together with V also a hypergraph $Tr(\mathcal{H}) = (V, T)$, which is called the *transversal hypergraph* of $\mathcal{H}$.

The famous Transversal Hypergraph Problem (TRANS-HYP) is then as follows:

Problem: Hypergraph Transversal (TRANS-HYP)
Instance: Two hypergraphs $\mathcal{G} = (V, E)$ and $\mathcal{H} = (V, F)$ on a finite set V.
Question: Does $\mathcal{G} = Tr(\mathcal{H})$ hold ?

Rephrased as a computation problem rather, the statement is as follows:

Problem: Transversal Enumeration (TRANS-ENUM)
Instance: A hypergraph $\mathcal{H} = (V, E)$ on a finite set V.
Output: The edges of the transversal hypergraph $Tr(\mathcal{H})$.

From the point of computability in polynomial time, the decisional and the computational variant of the transversal hypergraph problem are in fact equivalent: It is known that, for any class $\mathcal{C}$ of hypergraphs, TRANS-ENUM is solvable in *polynomial total time* (or *output-polynomial time*), i.e., in time polynomial in the combined size of $\mathcal{H}$ and $Tr(\mathcal{H})$, if and only if TRANS-HYP is in the class P for all pairs $(\mathcal{H}, \mathcal{G})$ such that $\mathcal{H} \in \mathcal{C}$ [3].

The problems TRANS-HYP and TRANS-ENUM have a large number of applications in many areas of Computer Science, including Distributed Systems, Databases, Boolean Circuits and Artificial Intelligence. There, they have important applications in Diagnosis, Machine Learning, Data Mining, and Explanation Finding, see e.g. [11, 13, 24, 28, 32, 33, 36] and the references therein.

Let us call a decision problem Π TRANS-HYP-*hard*, if problem TRANS-HYP can be reduced to it by a standard polynomial time transformation. Furthermore, Π is TRANS-HYP-*complete*, if Π is TRANS-HYP-hard and, moreover, Π can be polynomially transformed into TRANS-HYP; that is, Π and TRANS-HYP are equivalent modulo polynomial time transformations. We use analogous terminology of TRANS-ENUM-*hardness* and TRANS-ENUM-*completeness* for computations problems, i.e., problems with output. Here, Π reduces to Π', if there a polynomial functions f, g s.t. for any input I of Π, $f(I)$ is an input of Π', and if O is the output for $f(I)$, then $g(O)$ is the output of I, cf. [40]; we also request that O has size polynomial in the size of the output for I (if not, trivial reductions may exist).

The rest of this paper is organized as follows. In the next two sections, we illustrate some of the applications of TRANS-HYP and TRANS-ENUM in Logic and in Artificial Intelligence. Some of the results have been established already some time ago [10, 11], and were announced in [11] but remained yet unpublished. After that, Section 4 is dedicated to a review of recent developments on complexity of TRANS-HYP, and a new result is contributed (Theorem 11). The final Section 5 presents some open issues.

We close this section with some terminology. For any decision problem Π, we denote by co-Π the complementary problem, which has yes- and no-instances exchanged. A hypergraph $\mathcal{H} = (V, E)$ is *simple* (or *Sperner*, *clutter-free*), if no edge e of $\mathcal{H}$ strictly contains any other edge e' of $\mathcal{H}$, i.e., $\forall e, e' \, inE : e \subseteq e \Rightarrow e = e'$. We denote by $\min(\mathcal{H})$ the hypergraph on V whose edges are the ones of $\mathcal{H}$ which are minimal with respect to set inclusion, i.e., $\min(\mathcal{H}) = (V, \{e \in E \mid \forall e' \in \mathcal{E}\, e' \not\subset e\})$. We refer to $\overline{\mathcal{H}} = (V, \{V \setminus e \mid e \in E\})$ as the *complemented hypergraph of* $\mathcal{H}$. If $\mathcal{H} = \overline{\mathcal{H}}$, then $\mathcal{H}$ is called *self-complemented*.

2 Applications in Logic

2.1 Satisfiability Checking

The satisfiability problem of propositional formulas, in particular of sets of clauses, has special importance in complexity theory. There is a strong interconnection between satisfiability of a set of clauses and 2-colorability of hypergraphs, and these problems are decidable by similar resolution methods, cf. [34].

Recall that the satisfiability problem (SAT) is to decide whether a set of propositional clauses $\mathcal{C} = \{C_1, \ldots, C_m\}$, where each C_i is a set of literals, on atoms $X = \{x_1, \ldots, x_n\}$ is satisfied by some truth assignment to X. Its restriction MSAT, where every clause of C must be either positive or negative, i.e., must not contain any negative or positive literal, respectively, is still NP-hard.

In what follows, let us denote $Y^{\neg} = \{\neg y \mid y \in Y\}$ for any set of atoms Y, and for any instance C of MSAT on atoms X, let $\mathcal{C}^+(C)$ and $\mathcal{C}^-(C)$ denote the hypergraphs of the families of atom sets in the positive and the negative clauses of C, respectively, i.e., $\mathcal{C}^+(C) = (X, C)$ and $\mathcal{C}^-(C) = (X, \{Y \subseteq X \mid Y^{\neg} \in C\})$. If C is clear, we simply write $\mathcal{C}^+$ and $\mathcal{C}^-$ for $\mathcal{C}^+(C)$ and $\mathcal{C}^-(C)$, respectively.

Example 1. Let $X = \{x_1, \ldots, x_4\}$, and $C = \{\{x_1\}, \{x_2, x_4\}, \{\neg x_2, \neg x_4\}, \{\neg x_3\}\}$. Then $\mathcal{C}^+ = (X, \{\{x_1\}, \{x_2, x_4\}\})$ and $\mathcal{C}^- = (X, \{\{x_2, x_4\}, \{x_3\}\})$.

The following lemma, which can be found in various forms in the literature, links satisfiability to hypergraph transversals. Let, for hypergraphs $\mathcal{H} = (V, E)$ and $\mathcal{H}' = (V, E')$, denote $\mathcal{H} \succeq \mathcal{H}'$ that every edge of $\mathcal{H}$ contains some edge of $\mathcal{H}'$, i.e., $\forall e \in E \;\; \exists e' \in E' : e' \subseteq e$ holds.

Lemma 1. *An instance C of* MSAT *is unsatisfiable iff $Tr(\mathcal{C}^+) \succeq \mathcal{C}^-$.*

Proof. We may identify every truth value assignment ϕ to X with the set $X_\phi \subseteq X$ of atoms which are assigned the value true. Let $\mathcal{X}^+_\phi$ denote the set of all truth assignments that satisfy the positive clauses in C. Clearly, every $\tau \in \mathcal{X}^+_\phi$ must be a transversal of $\mathcal{C}^+$, i.e. $(X, \mathcal{X}^+_\phi) \succeq Tr(\mathcal{C}^+)$ holds. Now assume $Tr(\mathcal{C}^+) \succeq \mathcal{C}^-$ is true. Since $\succeq$ is transitive, $(X, \mathcal{X}^+_\phi) \succeq \mathcal{C}^-$ holds and each $\tau \in \mathcal{X}^+_\phi$ assigns all literals in some negative clause of C false, hence C is unsatisfiable. Conversely, if $\mathcal{C}^-$ is unsatisfiable, then $(X, \mathcal{X}^+_\phi) \succeq \mathcal{C}^-$ holds. Since for each $\tau \in Tr(\mathcal{C}^+)$, clearly $\tau \in \mathcal{X}^+_\phi$, we have $Tr(\mathcal{C}^+) \succeq (X, \mathcal{X}^+_\phi)$ and thus $Tr(\mathcal{C}^+) \succeq \mathcal{C}^-$. $\square$

The first subcase of MSAT which we consider is made up of instances with the property that every pair of a positive clause and a negative clause resolves, i.e. there is an atoms which occurs unnegated in the positive and negated in the negative clause. This subcase of MSAT, which we refer to turns out to be computationally equivalent to TRANS-HYP.

More formally, a set of clauses C (on atoms X) is an instance of INTERSECTING MSAT (IMSAT), if C is an instance of MSAT and, moreover, C satisfies $\forall e \in \mathcal{C}^+ \, \forall e' \in \mathcal{C}^- : e \cap e' \neq \emptyset$.

We have the following characterization of satisfiable IMSAT clause sets.

Theorem 1. *An instance C of* IMSAT *is satisfiable iff $Tr(\min(\mathcal{C}^+)) = \min(\mathcal{C}^-)$.*

Proof. Since C is an IMSAT instance, $\min(\mathcal{C}^-) \succeq Tr(\min(\mathcal{C}^+))$ holds. Furthermore, as easily checked, $\succeq$ is a partial order on simple hypergraphs, i.e., for any simple hypergraphs $\mathcal{G}$ and $\mathcal{H}$, $\mathcal{H} \succeq \mathcal{G}$ and $\mathcal{G} \succeq \mathcal{H}$ is equivalent to $\mathcal{H} = \mathcal{G}$. Thus, $Tr(\min(\mathcal{C}^+)) = \min(\mathcal{C}^-)$ is equivalent to $Tr(\min(\mathcal{C}^+)) \succeq \min(\mathcal{C}^-)$. By Lemma 1, C is satisfiable iff $Tr(\mathcal{C}^+) \not\succeq \mathcal{C}^-$, which is equivalent to $Tr(\min(\mathcal{C}^+)) \not\succeq \min(\mathcal{C}^-)$. Hence, the result follows. $\square$

Corollary 1. *Problem co-*IMSAT *is* TRANS-HYP*-complete.*

Proof. By Theorem 1, unsatisfiability testing for any IMSAT instance C is clearly polynomial transformable into TRANS-HYP. Conversely, any instance of co-TRANS-HYP is also easily transformable into an equivalent instance of co-IMSAT by Theorem 1. □

We remark that for the more general class of SAT instances where *every* pair of (distinct) clauses resolves, SAT is decidable in polynomial time. This follows from the observation of D.A. Plaisted (1991, personal communication) that such a clause set $C = \{C_1, \ldots, C_m\}$ is satisfiable if and only if $\sum_{i=1}^{m} 2^{-|C_i|} \neq 1$, provided that no clause is subsumed by some other clause and that no variable occurs both negated and unnegated in the same clause. (Note that since all numbers $|C_i|$ are given in *unary* notation, computing this sum is easy.)

In comparison to this subclass of SAT, the intersection restriction on the clause sets in IMSAT is weaker in a sense, since positive and negative clauses, respectively, are not interrelated among each other, and hence the whole clause set "splits" into two groups. However, imposing the intersection condition also on the positive and negative clauses makes IMSAT no easier.

Let SYMMETRIC IMSAT (SIMSAT) be the 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.)

Corollary 2. *Problem co-*SIMSAT *is* TRANS-HYP*-complete.*

Proof. By Corollary 1, any SIMSAT instance C can be polynomially transformed into a TRANS-HYP instance $(\mathcal{H}, \mathcal{G})$ such that C is unsatisfiable iff $(\mathcal{H}, \mathcal{G})$ is a yes-instance of TRANS-HYP. On the other hand, as shown in [11], instances $(\mathcal{H}, \mathcal{G})$ of TRANS-HYP such that $\mathcal{H} = \mathcal{G}$ are TRANS-HYP-hard. We may, without loss of generality, assume in addition that $\mathcal{H} = (V, E)$ is simple, that $e \cap e' \neq \emptyset$, for all $e, e' \in E$, and that $|E| > 0$. By Theorem 1, the clause set $C = E \cup \{e^\neg \mid e \in E\}$ on atoms V is satisfiable iff $\mathcal{H} \neq Tr(\mathcal{H})$; since C is efficiently constructible from $\mathcal{H}$, the result follows. □

Yet other subcases of SAT can be found whose unsatisfiability is equivalent to TRANS-HYP; one are MSAT instances where the positive clauses are a collection of partitions of the variable set X, i.e. if $X' \subseteq X$ appears, then also $X - X'$ appears, such that no positive clause subsumes any other, and the negative clauses are the clauses which result if, for each clause C_i, a positive atom from $X \setminus C_i$ is added to C_i in all ways, and then all atoms are negated.

There is an interesting observation regarding the clause sizes of the instances. Problem MSAT remains NP-complete even if each clause contains at most three literals (M3SAT, [20]); if an instance C of M3SAT is also an instance of IM3SAT, however, deciding satisfiability is polynomial: We may select any positive clause C_1 and negative clause C_2 from C, and for every of the at most $2^5 = 32$ truth value assignments to the literals in $C_1 \cup C_2$, the clause set is reducible to an instance of 2SAT, since each such assignment leaves in every clause at most two literals unassigned; 2SAT is well-known polynomial [20].

By Theorem 1 and results in [11], polynomiality of satisfiability testing generalizes from IM3SAT to IMkSAT, the restriction of IMSAT in which the clause size is bounded by a constant k, and even more general, also to the subclass where the size of either only the positive clauses or the negative clauses is bounded by k. Note that for $k > 3$, checking whether C is satisfiable for the at most 2^{k-1} many truth assignments to a positive clause C_1 and a negative clause C_2 is not promising for an efficient procedure, since the reduced clause sets are not evidently contained in a tractable subclass of SAT.

2.2 Dualization

There is a well-known and close connection of TRANS-ENUM to the well-known dualization problem of Boolean Functions:

Problem: DUALIZATION
Instance: A CNF φ of a Boolean function $f = f(x_1, \ldots, x_n)$
Output: A prime CNF ψ of its dual f^d, i.e., the function which has value 1 in input vector $b = (b_1, \ldots, b_n)$ iff f has value 0 on the input vector $\overline{b} = (b_1 \oplus 1, \ldots, b_n \oplus 1)$.

Recall that a CNF φ is prime, if it consists of prime clauses, i.e., no literal may be removed from any clause γ of φ without violating logical equivalence.

Special algorithms for the dualization problem can be tracked down at least to the 60's of the past century, cf. [2]. It is not hard to see that this problem is intractable; in fact, its decisional variant:

Problem: DUAL
Instance: CNFs φ and ψ of Boolean functions f and g, respectively.
Output: Do φ and ψ represent a pair (f, g) of dual Boolean functions?

is co-NP-complete, where hardness holds even if ψ is asserted to be a prime CNF of g. In case of monotone Boolean functions, the dualization problem is equivalent to determining the transversal hypergraph. Let MONOTONE DUALIZATION be the subcase of DUALIZATION where f is a monotone Boolean function, and similarly MONOTONE DUAL the subcase of DUAL where f is a monotone Boolean function. (Notice that in general, deciding whether a given CNF represents a monotone function is intractable, and thus strictly speaking these cases are promise problems [29], since valid input instances are not recognized in polynomial time. Under suitable syntactic restrictions, such as that the CNFs for f are negation-free, this can be ensured.)

The following result summarizes well-known results that are part of the folklore.

Theorem 2. MONOTONE DUALIZATION *is* TRANS-ENUM-*complete, and* MONOTONE DUAL *is* TRANS-HYP-*complete.*

Proof. Let φ be a CNF expression for a monotone Boolean function f on variables $X = \{x_1, \ldots, x_n\}$. Then, the CNF φ' which results from φ by removing all negative literals from φ is logically equivalent to φ. Indeed, clearly φ' logically implies φ. On the other hand, each clause γ in φ must be subsumed by same prime implicate $\gamma^\star$ of f. As well known, a Boolean function f is monotone iff every prime implicate of f is positive, i.e., contains no negative literal. Thus, γ is replaced in φ' by some clause γ' such that $\gamma^\star \subseteq \gamma'$; hence, φ logically implies φ'. The clauses of the (unique) prime CNF ψ of the dual function $g = f^d$ are then given by the edges of $Tr(\mathcal{C}^+(C))$, where $\mathcal{C}^+(C)$ is the hypergraph associated with the set C of clauses in φ'.

Thus, Monotone Dualization and reduces in polynomial time to TRANS-ENUM. Similarly, Monotone Dual reduces in polynomial time to TRANS-HYP, since f^d is monotone if f is monotone, and if a CNF ψ representing a monotone Boolean function g is $\neg$-free, then the unique prime CNF of g is described by the edges of $\min(\mathcal{C}^+)$. On the other hand, by the correspondence $\mathcal{C}^+(C)$ between a monotone clause set C and a hypergraph, TRANS-ENUM and TRANS-HYP trivially reduce to Monotone Dualization and Monotone Dual, respectively, in polynomial time. □

3 Applications in Artificial Intelligence

3.1 Theory Revision

Dealing with *knowledge in flux*, i.e. with knowledge bases that are subject to change in the light of new information, is an important problem of artificial intelligence. The subject has been studied in the database research community as well (e.g., [16, 45]) and has also attracted attention of philosophers of reasoning (e.g., [1, 19]). A number of different change approaches have been proposed (see [45, 39] for overviews); many of them adhere to the principle of *Minimal Change*, according to which the body of knowledge should change as little as possible if new information is incorporated.

A well-known knowledge change approach is the Set-Of-Theories approach, a formula-based change method first defined in [16] in the context of databases and considered for AI applications e.g. in [22, 38]. This method works as follows. Assume that the sentence p must be incorporated into the knowledge base T, which is a finite set of sentences. If T is consistent with p, then p is simply added to T; otherwise, $\neg p$ is provable from T. In this case, a set R of sentences is removed from T such that $\neg p$ is no longer provable from $T - R$, and then p is added; by the principle of Minimal Change, this set R is selected as small as possible under set inclusion.

Observe that R must pick a sentence from every subset $T' \subseteq T$ of sentences from which $\neg p$ is provable. Denote the collection of all such subsets $T' \subseteq T$ by $\mathcal{P}(T; \neg p)$. Thus, R must be a transversal of the hypergraph $(T, \mathcal{P}(T; \neg p))$; since R is selected as small as possible, R must even be minimal.

In general, R will not be unique; the result of the update, $\mathcal{U}(T; p)$, is thus defined as the set of all KBs obtained by removing some possible R and adding p. The possible R's constitute, as easily seen, the edges of $Tr(T, \mathcal{P}(T; \neg P))$.

Formally, let $T = \{f_1, \ldots, f_n\}$ be a finite set of satisfiable sentences and let p be a satisfiable sentence in a suitable logic. Then, define

$$\begin{aligned}
\mathcal{P}(T; \neg p) &= \{T' \subseteq T \mid T' \models \neg p\}, \\
\mathcal{W}(T; p) &= \{T - R \mid \mathit{Tr}(T, \mathcal{P}(T; \neg p)) = (T, E),\ R \in E\}, \\
\mathcal{U}(T; p) &= \{W \cup \{p\} \mid W \in \mathcal{W}(T; p)\}.
\end{aligned}$$

$\mathcal{W}(T; p)$ denotes the "possible worlds" [22], which are with respect to set inclusion maximal subsets of T in which the sentence p may be true. It is easily verified that $\mathcal{W}(T; p)$ is the collection of all *maximal independent sets* of the hypergraph $\mathcal{P}(T; \neg p) = (T, E)$, i.e., the maximal sets (under set inclusion) $W \subseteq T$ such that $S \not\subseteq W$ for every $S \in E$, and that $\mathcal{W}(T; p) = \{T' \subseteq T \mid T' \not\models \neg p \wedge \forall U : T' \subset U \subseteq T \Rightarrow U \models \neg p\}$. A sentence f is implied by $\mathcal{U}(T; p)$ (denoted $\mathcal{U}(T; p) \models f$), if $T' \models f$ for each T' from $\mathcal{U}(T; p)$, i.e., if f follows from the logical disjunction of the knowledge bases in $\mathcal{U}(T; p)$.

Example 2. Let $T = \{x_1,\ x_2,\ x_1 \wedge x_2 \Rightarrow x_3\}$ and $p = \neg x_3$ in propositional logic, where x_1, x_2, x_3 are propositional atoms. Then

$$\begin{aligned}
\mathcal{P}(T; \neg p) &= \{\{x_1, x_2, x_1 \wedge x_2 \Rightarrow x_3\}\}, \\
\mathit{Tr}(T, \mathcal{P}(T; \neg p)) &= \{\{x_1\}, \{x_2\}, \{x_1 \wedge x_2 \Rightarrow x_3\}\}, \\
\mathcal{W}(T; p) &= \{\{x_2, x_1 \wedge x_2 \Rightarrow x_3\}, \{x_1, x_1 \wedge x_2 \Rightarrow x_3\}, \{x_1, x_2\}\}, \\
\mathcal{U}(T; p) &= \{\{x_2, x_1 \wedge x_2 \Rightarrow x_3, \neg x_3\}, \{x_1, x_1 \wedge x_2 \Rightarrow x_3, \neg x_3\}, \{x_1, x_2, \neg x_3\}\}.
\end{aligned}$$

We have $\mathcal{U}(T; p) \models x_1 \vee x_2$, but neither $\mathcal{U}(T; p) \models x_1$ nor $\mathcal{U}(T; p) \models x_2$.

Procedures to compute $\mathcal{U}(T; p)$ are described in [22, 23]. They basically proceed in two steps: In the first step, the set $\mathcal{MP}(T; \neg p)$ of all minimal proofs of $\neg p$ from T, i.e. $(T, \mathcal{MP}(T; \neg p)) = \min(T, \mathcal{P}(T; \neg p)))$, is computed by a theorem prover. Then, $\mathcal{U}(T; p)$ is computed from $\mathcal{MP}(T; \neg p)$ collecting all sets $S = (T \cup \{p\}) - R$, where R is a transversal of $\mathcal{MP}(T; \neg p)$, such that S is maximal under set inclusion; [22] suggests to compute $\mathcal{U}(T; p)$ incrementally, i.e. "world by world". The essential task in step 2 of this method is to determine $\mathcal{W}(T; p)$. Consider the following decision problem:

Problem: Additional World
Instance: A finite satisfiable set T of first-order sentences, a satsifiable first-order sentence p, the collection $\mathcal{MP}(T; \neg p)$, and a collection $\mathcal{W} \subseteq \mathcal{W}(T; p)$.
Question: Does there exist $T' \in \mathcal{W}(T; p) - \mathcal{W}$?

If this problem is intractable, then no output-polynomial algorithm to compute the possible worlds $\mathcal{W}(T; p)$ from $\mathcal{MP}(T; \neg p)$ is likely to exist. We note the following result, which appears in [10]:

Theorem 3. *Problem co-*Additional World *is* TRANS-HYP*-complete.*

Proof. It is easily seen that $\mathcal{W}(T; p)$ are the edges of $\overline{\mathit{Tr}(T, \mathcal{MP}(T; \neg p))}$. Since $\mathcal{W} \subseteq \mathcal{W}(T; p)$, this problem is thus easily reduced in polynomial time transformable to the complement of TRANS-HYP. On the other hand, let $(\mathcal{H}; \mathcal{G})$ be a

TRANS-HYP instance, such that $\mathcal{H}$ and $\mathcal{G}$ are simple, $\mathcal{H} \subseteq Tr(\mathcal{G})$, and $\mathcal{G} = (V, E)$ has no empty edge.

Take V as propositional atoms, and denote by $f(e_i)$ the formula $\neg v_1 \vee \cdots \vee \neg v_k$, where $e_i = \{v_1, \ldots, v_k\}$ and $E = \{e_1, \ldots, e_m\}$, for $1 \leq i \leq m$. Then, define $T = V$ and $p = f(e_1) \wedge \cdots \wedge f(e_m)$. It is immediately verified that T as well as p is satisfiable and that $\mathcal{MP}(T; \neg p) = E$. Therefore, $\overline{\mathcal{W}(T;p)} = \overline{Tr(T, \mathcal{MP}(T; \neg p))} = \overline{Tr(\mathcal{G})}$. Since $\mathcal{H} \subseteq Tr(\mathcal{G})$, we have that $\overline{\mathcal{H}} \subseteq \overline{Tr(\mathcal{G})} = \mathcal{W}(T;p)$. Therefore, $\mathcal{H} = Tr(\mathcal{G})$ holds iff there exists no additional world for p, i.e. $\mathcal{W}(T;p) = \overline{\mathcal{H}}$. Hence, co-ADDITIONAL WORLD is TRANS-HYP-hard. $\square$

Note that [24] showed (independently) TRANS-ENUM-hardness of computing $\mathcal{U}(T;p)$ from T and p if all formulas in T and p are Horn; this result is implicit in the proof above.

By results in [11], computing $\mathcal{W}(T;p)$ from $\mathcal{MP}(T; \neg p)$ is polynomial in the input size if $\neg p$ has few minimal proofs from T, and is output-polynomial if the minimal proofs for $\neg p$ only refer to at most constantly many sentences of T.

3.2 Machine Learning and Data Mining

Im machine learning, the hypergraph transversal problem is related to problems in learning Boolean functions. In a simple model, an adversary fixes a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, which models a concept in the world. The learner should find out f, or an approximation of it, given access to an oracle which discloses partial information about f. In particular, a *membership query oracle* $MQ(f)$ allows the learner to ask the value of f at a given vector $b \in \{0,1\}^n$. The hypothesis on f produced by the learner is a Boolean formula, which is typically a CNF or a DNF expression.

The following result is due to [28]. Let, for any monotone Boolean function f, denote $CNF(f)$ and $DNF(f)$ its unique prime CNF and prime DNF, respectively.

Theorem 4. *If* TRANS-ENUM *can be solved in output-polynomial time, then there exists a learning algorithm for monotone Boolean functions f with membership queries, which produces both a CNF and a DNF representation of the function. The number of membership queries is bounded by $|CNF(f)| \cdot (|DNF(f)| = n^2)$ and the running time of the algorithm is polynomial in n, $|CNF(f)|$, and $|DNF(f)|$.*

There is also a relation to hypergraph transversals in the other direction, which has also been pointed out in [28].

Theorem 5. *If there is a learning algorithm for monotone Boolean functions f with membership queries, which produces both a DNF representation of the function and whose running time and number of queries to $MQ(f)$ are bounded by a function $T(n + |DNF(f)| + |CNF(f)|)$, then* TRANS-ENUM *can be solved in time $T(n + |DNF(f)| + |CNF(f)|)$.*

Thus, if $T()$ is a polynomial, then TRANS-ENUM can be solved in output-polynomial time. Note that as shown in [43], for almost all monotone Boolean functions a polynomial $T()$ exists if n tends to infinity.

These results are closely related to results on generating frequent sets in data mining. Given a $0/1$ $m \times n$ matrix A and an integral threshold t, associate with each subset $C \subseteq \{1, \ldots, n\}$ of column indices the subset $R(C)$ of all rows $r \in \{1, \ldots, m\}$ in A such that $A(r, j) = 1$ for every $j \in C$. Then C is called *frequent*, if $|R(C)| \geq t$, and C is called *infrequent*, if $|R(C)| < t$. Let us denote by $F_t(A)$ and $\hat{F}_t(A)$ the sets of all frequent and infrequent column sets C in A, respectively.

The generation of frequent and infrequent sets in A is a key problem in knowledge discovery and data mining, which occurs in mining association rules, correlations, and other tasks. Of particular interest are the maximal frequent sets $M_t \subseteq F_t$ and the minimal infrequent sets $I_t \subseteq \hat{F}_t$, since they mark the boundary of frequent sets (both maximal and minimal under set inclusion). The following result has been recently proved in [6].

Theorem 6. *The problem of computing, given a 0/1 matrix A and a threshold t, the sets M_t and I_t is* TRANS-ENUM*-complete.*

A related but different task in data mining is dependency inference. Here the problem is, given a (not necessarily $0/1$) $m \times n$ matrix A, to find the dependencies $C_{i_1} C_{i_2} \cdots C_{i_k} \rightarrow C_{i_0}$ which hold on A (denoted $A \models C_{i_1} C_{i_2} \cdots C_{i_k} \rightarrow C_{i_0}$), i.e., for each pair of rows $r, r' \in \{1, \ldots, m\}$, either $A(r, i_j) \neq A(r', i_j)$ for some $j \in \{1, \ldots, k\}$ or $A(r, i_0) = A(r', i_0)$. Such implications are known as *functional dependencies* (FDs) in databases. The set $Dep(A) = \{f \mid A \models f\}$ of all dependencies which hold on A, is usually represented by a subset $C \subseteq Dep(A)$, called a *cover*, such that $\{f \mid C \models f\} = Dep(A)$, where $C \models f$ is logical implication. Consider thus the following problem:

Problem: FD-RELATION EQUIVALENCE (FD-EQ)
Instance: A $m \times n$ matrix A, a set D of dependencies.
Question: Is D a cover for $Dep(A)$?

This problem, as shown in [27], is TRANS-HYP-hard; however, it is not known whether FD-EQ is reducible to TRANS-HYP in polynomial time; as shown in the extended version of [11], this is possible if D is in *MAK*-form, i.e., $D = \{X \rightarrow C_{i_0} \in Dep(A) \mid X = C_{i_1} \cdots C_{i_k}$, and $X \setminus C_{i_j} \rightarrow C_{i_0} \notin Dep(A)$, for all $j \in \{1, \ldots, k\}\}$. (Informally, X is a minimal key or prime implicant for attribute C_{i_0}.)

Theorem 7. *Problem* FD-EQ *for instances (A, D) where D is in MAK-form is* TRANS-HYP*-complete.*

Some polynomial cases of FD-EQ are given in [27]. As shown in [33], FD-EQ is related to similar problems involving charcteristic models and Horn CNFs. For these and further results about problems in data mining equivalent to TRANS-ENUM and TRANS-HYP, see [33].

3.3 Model-Based Diagnosis

Different from the heuristic approach, model-based diagnosis [41, 8] takes a logical description of the technical system and the behavior of its components as a basis for diagnosing faults by means of consistency. Since model-based diagnosis offers several advantages, much efforts were spent on it in the last 15 years.

Briefly, a *system* is a pair $(SD, COMP)$, where SD, the *system description*, is a set of sentences from an underlying decidable fragment of first-order logic (e.g., propositional logic) and $COMP$ is a set of constants which model the system *components*. SD is used together with a set OBS of sentences, which are particular *observations* on the system behavior, to diagnose faults. For that, SD makes use of a distinguished predicate $AB(c)$ which reads "component c operates in abnormal mode". Now a *diagnosis* for $(SD, COMP, OBS)$ is a minimal (under inclusion) set $\Delta \subseteq COMP$ of components such that

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

is satisfiable.If there is no fault, $\Delta = \emptyset$ must be a diagnosis, otherwise the system description SD is not sound.

In [41] 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 unsatifiable; C is minimal, if no proper subset of C is a conflict set. In terms of hypergraphs, the fundamental theorem on conflict sets and diagnoses in [41] is as follows. Let $CS(DP)$ denote the collection of all all minimal conflict sets of $DP = (SD, COMP, OBS)$.

Theorem 8. *$\Delta \subseteq COMP$ is a diagnosis for $DP = (SD, COMP, OBS)$ iff $\Delta \in Tr(COMP, CS(DP))$.*

Therefore, we obtain the following result for computing all dignoses.

Corollary 3. *The problem of computing, given $CS(DP)$ of s diagosis problem $DP = (SD, COMP, OBS)$, all diagnoses of DP is* TRANS-ENUM*-complete.*

Proof. By Theorem 8, it remains to show TRANS-ENUM-hardness. Let $\mathcal{H} = (V, E)$ be simple and define $DP = (SD, COMP, OBS)$ by $SD = \{AB(v_1) \vee \cdots \vee AB(v_k) \vee a \mid \{v_1, \ldots, v_k\} \in E\}$, $OBS = \{\neg a\}$, and $COMP = V \cup \{a\}$, where a is a fresh symbol. As easily checked, $CS(DP) = \mathcal{H}$, and thus the edges of $Tr(\mathcal{H})$ are given by the diagnoses of DP. □

We remark that modulo the costs of satisfiability checking, a diagnosis for $DP = (SD, COMP, OBS)$ can be found in polynomial time: Set $C = COMP$, and test if C is a superset of some diagnosis. Subsequently remove any component c from C such that $C - \{c\}$ is also superset of a diagnosis. Repeating this elimination step until no longer possible, the procedure stops with C as a diagnosis. However, given a set D of diagnoses for DP, deciding whether there is another diagnosis not in D was proved NP-complete in [18] for propositional Horn theories, where consistency checking is polynomial.

3.4 Horn Envelope

Recall that a logical theory Σ is *Horn*, if it is a set of Horn clauses, i.e., disjunctions $l_1 \vee \cdots \vee l_m$ of literals l_i such that at most one of them is positive. Semantically, Horn theories are characterized by the property that their set of models, $mod(\Sigma)$, is closed under intersection, i.e., $M, M' \in mod(\Sigma)$ implies $M \cap M' \in mod(\Sigma)$; here, $M \cap M'$ is the model M'' which results by atomwise logical conjunction of M and M', i.e., $M'' \models a$ iff $M \models a$ and $M' \models a$, for every atom a.

Any theory Σ has a unique *Horn envelope*, which is the strongest (w.r.t. implication) Horn theory Σ' such that $\Sigma \models \Sigma'$. The Horn envelope might be represented by different Horn theories, but there is a unique representation, which we denote by $HEnv(\Sigma)$, which consists of all prime clauses of Σ'. The following result was established in [33], where the TRANS-ENUM hardness part was proved in [32].

Theorem 9. *The problem of computing, given the models $mod(\Sigma)$ of a propositional theory Σ, the Horn envelope $HEnv(\Sigma)$ is* TRANS-ENUM-*complete.*

3.5 Abductive Explanations

Abduction is a fundamental mode of reasoning, which has been recognized as an important principle of common-sense reasoning which is in particular used for finding explanations. Given a Horn theory Σ, called the background theory, a formula q (called query), and a set of literals $A \subseteq Lit$, an *explanation of q w.r.t. A* is a minimal set of literals E over A such that (i) $\Sigma \cup E \models q$, and (ii) $\Sigma \cup E$ is satisfiable.

As shown by Kautz *et al.* [31], it is possible to generate an explanation for a query q that is an atom, w.r.t. a given set $A = S \cup \{\neg p \mid p \in S\}$ of literals over a given subset S of the atoms, in polynomial time, if Σ is semantically represented by the set of its characteristic models, $char(\Sigma)$; under syntax-based representation, the problem is intractable [42]. A model m of Σ is *characteristic*, if $m \notin Cl_\wedge(mod(\Sigma) \setminus \{v\})$, where $Cl_\wedge(S)$ denotes for any set of models S its closure under intersection, i.e., the least set of models $S' \supseteq s$ such that $M, M' \in S'$ implies $M \cap M' \in S'$.

The polynomial abduction algorithm in [31] implicitly involves the computation of a minimal transversal of a hypergraph, which is the bulk of the computation effort. This result has been extended in [13] to the following result.

Theorem 10. *Given the characteristic set $char(\Sigma)$ of a Horn theory Σ, a query literal q, and a subset A of all literals, computing the set of all explanations for q from Σ w.r.t. A is* TRANS-ENUM-*complete.*

Thus, the generation of explanations and of the transversal hypergraph are intimately related in the model-based representation framework.

4 Recent Developments on Complexity of the Problems

4.1 Structural complexity

Let us first turn to issues of *structural* complexity. In a landmark paper, Fredman and Khachiyan [17] proved that TRANS-HYP can be solved in time $n^{o(\log n)}$, and thus in quasi-polynomial time. This shows that the problem is most likely not co-NP-complete, since no co-NP-complete problem is known which is solvable in quasi-polynomial time; if any such problem exists, then all problems in NP and co-NP can be solved in quasi-polynomial time.

A natural question is whether TRANS-HYP lies in some lower complexity class based on other resources than just runtime. In a recent paper [12], it was shown that the complement of this problem is solvable in polynomial time with *limited nondeterminism*, i.e, by a nondeterministic polynomial-time algorithm that makes only a poly-logarithmic number of guesses in the size of the input. For a survey on complexity classes with limited nondeterminism, and for several references see [25]. More precisely, [12] shows that non-duality of a pair $\mathcal{G}, \mathcal{H})$ can be proved in polynomial time with $O(\chi(n) \cdot \log n)$ suitably guessed bits, where $\chi(n)$ is given by $\chi(n)^{\chi(n)} = n$; note that $\chi(n) = o(\log n)$.

This result is surprising, because most researchers dealing with the complexity of the transversal hypergraph thought so far that these problems are completely unrelated to limited nondeterminism.

4.2 Tractable cases

A large number of tractable cases of TRANS-HYP and TRANS-ENUM are known in the literature, e.g. [7, 5, 4, 9, 11, 14, 12, 21, 35, 37], and references therein.

Examples of tractable classes are instances $(\mathcal{H}, \mathcal{G})$ where $\mathcal{H}$ has the edge sizes bounded by a constant, or where $\mathcal{H}$ is *acyclic*. Various "degrees" of hypergraph acyclicity have been defined in the literature [15]. The most general notion of hypergraph acyclicity (applying to the largest class of hypergraphs) is α-acyclicity; less general notions are (in descending order of generality) β-, γ-, and Berge-acyclicity (see [15]). In [11], it was shown that Hypergraph transversal instances with β-acyclic $\mathcal{H}$ are tractable. In [12], this tractability result has been recently improved to instances where $\mathcal{H}$ is α-acyclic and simple. This result is a corollary to a more general tractability result for hypergraphs whose *degeneracy* is bounded by a constant; simple, α-acyclic hypergraphs have degeneracy 1.

Furthermore, [12] shows that instances $(\mathcal{H}, \mathcal{G})$ of TRANS-HYP where the vertex-hyperedge incidence graphs of $\mathcal{H}$ (or of $\mathcal{G}$) have *bounded treewidth* are solvable in polynomial time. Note that this class of hypergraphs does not generalize α-acyclic hypergraphs. In [26] a concept of *hypertree width* of a hypergraph was defined, which generalizes α-acyclicity properly, as follows.[1]

A *hypertree for a hypergraph* $\mathcal{H} = (V, E)$ is a triple $\langle T, \chi, \lambda \rangle$ where T is a rooted tree (N, A) and $\chi : N \to 2^V$ and $\lambda :\to 2^E$ are labeling functions which

[1] Actually, this notion was defined for conjunctive queries rather than hypergraphs, but is obvious from the natural correspondence between queries and hypergraphs.

associate with each node $n \in T$ a set of vertices $\chi(n) \subseteq V$ and a set of edges $\lambda(n) \subseteq E$. For any subtree $T' = (N', A')$ of T, we denote $\chi(T') = \bigcup_{n \in N'} \chi(n)$.

Definition 1. *A hypertree $\langle T, \chi, \lambda \rangle$ for a hypergraph $\mathcal{H} = (V, E)$ is a* hypertree decomposition *of $\mathcal{H}$, if it satisfies the following conditions:*

1. *for each $e \in E$, there exists some $n \in N$ such that $e \subseteq \chi(n)$, i.e., each edge of E is covered by some vertex label of T;*
2. *for each vertex $v \in V$, the set $\{n \in N \mid v \in \lambda(n)\}$ induces a connected subtree in T;*
3. *for each node $n \in N$, $\chi(n) \subseteq \bigcup \lambda(n)$ holds, i.e., each vertex in the vertex label occurs in some edge of the edge label; and*
4. *for each node $n \in N$, $\bigcup \lambda(n) \cap \chi(T_n) \subseteq \chi(n)$, where T_n denotes the subtree of T rooted at n; that is, each vertex v that occurs in some edge of the edge label and in the vertex label of n or some node below, must already occur in the vertex label of n.*

The width *of hypertree decomposition $\langle T, \chi, \lambda \rangle$ is given by $\max_{n \in N} |\lambda(n)|$, i.e., the largest size of some edge label.*

Then, the *hypertree width* of a hypergraph $\mathcal{H}$ is the minimum width over all hypertree decompositions for $\mathcal{H}$.

A (nonempty) hypergraph is α-acyclic, if and only if its hypertree width is equal to one. It was shown that several NP-hard decision problems whose underlying structure can be described by a hypergraph become tractable if its hypertree width is bounded by a constant. In particular, this turned out to be the case for answering Boolean conjunctive queries and for finite-domain constraint satisfaction problems. However, as the following result shows, even a bound on the hypertree-with as low as two does not imply the tractability of TRANS-HYP (unless P=NP). Denote by HT_k the subcase of TRANS-HYP where the hypertree width of $\mathcal{H}$ in instances $(\mathcal{H}, \mathcal{G})$ is bounded by k.

Theorem 11. HT_2 *for instances $(\mathcal{H}, \mathcal{G})$ where $\mathcal{H}$ is simple is* TRANS-HYP*-hard.*

Proof. It obviously suffices to logspace-reduce a TRANS-HYP instance $(\mathcal{H}, \mathcal{G})$ to an instance $(\mathcal{H}^*, \mathcal{G}^*)$ of HT_2, where $\mathcal{H} = (V, E)$ is simple and $|E| > 1$. We define $\mathcal{H}^* = (V^*, E^*)$ as follows. Let $V^* = V \cup \{a, b\}$, where a and b are new distinct vertices, and let E^* consist of the following edges:

– $e_{ab} := \{a, b\}$,
– $e_{aV} := \{a\} \cup V$,
– $e_b := e \cup \{b\}$, for each $e \in E$.

It is not hard to see that $\mathcal{H}^*$ is simple and has hypertree width bounded by 2. In fact, a hypertree decomposition of width 2 is given by the hypertree decomposition $\langle T, \chi, \lambda \rangle$ where the root of T groups together the two edges e_{ab} and e_{aV}, and where each edge e_b is appended as a singleton child to the root. Moreover, all variables at each node n of the decomposition are "relevant", i.e., in terms of [26], $\chi(n) = \bigcup \lambda(n)$. On the other hand, there is no hypertree decomposition $\langle T, \chi, \lambda \rangle$ for $\mathcal{H}$ which has width 1.

Now observe that the minimal transversals of $\mathcal{H}^*$ are the following subsets of V^*: $\{a, b\}$, $\{a\} \cup \tau$, where $\tau \in Tr(\mathcal{H})$, and $\{b, x\}$, where $x \in V$.

Let $\mathcal{G}^* = (V^*, F^*)$ where F^* consists of the edges $\{a, b\}$, $\{a\} \cup \tau$, where $\tau \in \mathcal{G}$, and $\{b, x\}$, where $x \in V$. It is clear that $Tr(\mathcal{H}) = \mathcal{G}$ iff $Tr(\mathcal{H}^*) = \mathcal{G}^*$. Moreover, the reduction is clearly feasible in polynomial time (in fact, even in logspace).

Note that this reduction can also be used to transform TRANS-ENUM to the subcase of HT_2 instances in polynomial time. □

Thus, different from hypergraph degeneracy, bounded hypertree-width alone is not a criterion which lets us establish tractability of TRANS-HYP.

5 Conclusion

Computing some or all minimal transversals of hypergraph is an important problem, which has applications to many areas of computer science, and in particular to logic and AI. While many algorithms or solving this problem exist, cf. e.g. [30, 10, 12], unfortunately the complexity of this problem is not fully understood today, and it is open whether the problem is tractable (in the sense of permitting output-polynomial algorithms [30]) or not. Recent progress on the status of this problem, and in particular solvability by limited nondeterminism, suggests however that this problem is more likely to be expected tractable than intractable.

Several open issues remain for further work. One is whether the amount of nondeterminism for solving TRANS-HYP and TRANS-ENUM can be further significantly decreased, e.g., to $O(\log \log v \cdot \log v)$ many bits. Another issue is, of course, to find new tractable cases; an interesting question is whether the tractability result for TRANS-HYP where either $\mathcal{G}$ or $\mathcal{H}$ has edge size bounded by a constant can be generalized to a non-constant bound, in particular to bound $O(\log n)$. Finally, it remains to formally assess that known algorithms for solving TRANS-HYP (see, e.g., [10]) are non-polynomial; only most recently, it was shown that a simple, well-known algorithm which uses additional information is not polynomial [44].

Acknowledgments. This work was supported by the Austrian Science Fund (FWF) Project Z29-INF.

References

1. C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Logic*, 50:510–530, 1985.
2. C. Benzaken. Algorithme de dualisation d'une fonction booléenne. *Revue Francaise de Traitment de l'Information – Chiffres*, 9(2):119–128, 1966.
3. C. Bioch and T. Ibaraki. Complexity of identification and dualization of positive Boolean functions. *Information and Computation*, 123:50–63, 1995.

4. E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan. An efficient incremental algorithm for generating all maximal independent sets in hypergraphs of bounded dimension. *Parallel Processing Letters*, 10(4):253–266, 2000.
5. E. Boros, V. Gurvich, and P. L. Hammer. Dual subimplicants of positive Boolean functions. *Optimization Methods and Software*, 10:147–156, 1998.
6. E. Boros, V. Gurvich, L. Khachiyan, and K. Makino. On the complexity of generating maximal frequent and minimal infrequent sets. In *Proc. STACS-02*, LNCS 2285, pp. 133–141, 2002.
7. E. Boros, P. Hammer, T. Ibaraki, and K. Kawakami. Polynomial time recognition of 2-monotonic positive Boolean functions given by an oracle. *SIAM J. Comput.*, 26(1):93–109, 1997.
8. J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.
9. C. Domingo, N. Mishra, and L. Pitt. Efficient read-restricted monotone CNF/DNF dualization by learning with membership queries. *Machine Learning*, 37:89–110, 1999.
10. T. Eiter. *On Transversal Hypergraph Computation and Deciding Hypergraph Saturation*. PhD thesis, Institut für Informationssysteme, TU Wien, Austria, 1991.
11. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.
12. T. Eiter, G. Gottlob, and K. Makino. New results on monotone dualization and generating hypergraph transversals. In *Proc. ACM STOC-2002*, pp. 14–22, 2002. Full paper Tech. Rep. INFSYS RR-1843-02-05, TU Wien. Available as *Computer Science Repository Report (CoRR) nr. cs.DS/0204009* via URL: http://arxiv.org/abs/cs/0204009.
13. T. Eiter and K. Makino. On computing all abductive explanations. In *Proc. 18th National Conference on Artificial Intelligence (AAAI '02)*. AAAI Press, 2002.
14. T. Eiter, K. Makino, and T. Ibaraki. Decision lists and related Boolean functions. *Theoretical Computer Science*, 270(1-2):493–524, 2002.
15. R. Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *Journal of the ACM*, 30:514–550, 1983.
16. R. Fagin, J. D. Ullman, and M. Y. Vardi. On the semantics of updates in databases. In *Proc. PODS-83*, pp. 352–365, 1983.
17. M. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.
18. G. Friedrich, G. Gottlob, and W. Nejdl. Physical negation instead of fault models. In *Proc. AAAI-91*, July 1990.
19. P. Gärdenfors. *Knowledge in Flux*. Bradford Books, MIT Press, 1988.
20. M. Garey and D. S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.
21. D. Gaur and R. Krishnamurti. Self-duality of bounded monotone Boolean functions and related problems. In *Proc. 11th International Conference on Algorithmic Learning Theory (ALT)*, LNCS 1968, pp. 209–223. Springer, 2000.
22. M. L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30:35–79, 1986.
23. M. L. Ginsberg and D. E. Smith. Reasoning about action I: A possible worlds approach. *Artificial Intelligence*, 35:165–195, 1988.
24. G. Gogic, C. Papadimitriou, and M. Sideri. Incremental recompilation of knowledge. *J. Artificial Intelligence Research*, 8:23–37, 1998.
25. J. Goldsmith, M. Levy, and M. Mundhenk. Limited nondeterminism. *SIGACT News*, 27(2):20–29, 1978.

26. G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. In *Proc. 18th ACM Symp. on Principles of Database Systems (PODS-99)*, pp. 21–32, 1999. Full paper to appear in *Journal of Computer and System Sciences*.
27. G. Gottlob and L. Libkin. Investigations on Armstrong relations, dependency inference, and excluded functional dependencies. *Acta Cybernetica*, 9(4):385–402, 1990.
28. D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen. Data mining, hypergraph transversals, and machine learning. In *Proc. 16th ACM Symp. on Principles of Database Systems (PODS-97)*, pp. 209–216, 1997.
29. D. S. Johnson. A Catalog of Complexity Classes. In J. van Leeuwen, ed., *Handbook of Theoretical Computer Science*, A, chapter 2. Elsevier, 1990.
30. D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27:119–123, 1988.
31. H. Kautz, M. Kearns, and B. Selman. Reasoning with characteristic models. In *Proc. AAAI-93*, pp. 34–39, 1993.
32. D. Kavvadias, C. Papadimitriou, and M. Sideri. On Horn envelopes and hypergraph transversals. In W. Ng, editor, *Proc. 4th International Symposium on Algorithms and Computation (ISAAC-93)*, LNCS 762, pp. 399–405, 1993.
33. R. Khardon. Translating between Horn representations and their characteristic models. *J. Artificial Intelligence Research*, 3:349–372, 1995.
34. N. Linial and M. Tarsi. Deciding hypergraph 2-colorability by H-resolution. *Theoretical Computer Science*, 38:343–347, 1985.
35. K. Makino and T. Ibaraki. A fast and simple algorithm for identifying 2-monotonic positive Boolean functions. *Journal of Algorithms*, 26:291–302, 1998.
36. H. Mannila and K.-J. Räihä. Design by Example: An application of Armstrong relations. *Journal of Computer and System Sciences*, 22(2):126–141, 1986.
37. N. Mishra and L. Pitt. Generating all maximal independent sets of bounded-degree hypergraphs. In *Proc. Tenth Annual Conference on Computational Learning Theory (COLT-97)*, pp. 211–217, 1997.
38. B. Nebel. A knowledge level analysis of belief revision. In *Proc. 1st Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR-89)*, pp. 301–311, 1989.
39. B. Nebel. How Hard is it to Revise a Belief Base ? In D. Gabbay and Ph.Smets, eds, *Handbook on Defeasible Reasoning and Uncertainty Management Systems*, volume III: Belief Change, pp. 77–145. Kluwer Academic, 1998.
40. C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
41. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.
42. B. Selman and H. J. Levesque. Abductive and default reasoning: A computational core. In *Proc. AAAI-90*, pp. 343–348, 1990.
43. I. Shmulevich, A. Korshunov, and J. Astola. Almost all monotone boolean functions are polynomially learnable using membership queries. *Information Processing Letters*, 79:211–213, 2001.
44. K. Takata. On the sequential method for listing minimal hitting sets. In *Proc. Workshop on Discrete Mathematics and Data Mining, 2nd SIAM International Conference on Data Mining, April 11-13, Arlington, Virginia, USA*, 2002.
45. M. Winslett. *Updating Logical Databases*. Cambridge University Press, 1990.