J. Symbolic Computation (1994) **17**, 215–225

Exact Transversal Hypergraphs and Application to Boolean μ-Functions

THOMAS EITER

Christian Doppler Laboratory for Expert Systems, Information Systems Department, Technical University of Vienna[†]

(Received 18 September 1992)

Call an hypergraph, that is a family of subsets (edges) from a finite vertex set, an exact transversal hypergraph iff each of its minimal transversals, i.e., minimal vertex subsets that intersect each edge, meets each edge in a singleton. We show that such hypergraphs are recognizable in polynomial time and that their minimal transversals as well as their maximal independent sets can be generated in lexicographic order with polynomial delay between subsequent outputs, which is impossible in the general case unless **P**= **NP**. The results obtained are applied to monotone Boolean μ-functions, that are Boolean functions defined by a monotone Boolean expression (that is, built with $\wedge, \vee$ only) in which no variable occurs repeatedly. We also show that recognizing such functions from monotone Boolean expressions is co-**NP**-hard, thus complementing Mundici's result that this problem is in co-**NP**.

1. Introduction

A hypergraph, i.e., a family of subsets (called edges) of a finite vertex set, is a natural generalization of the concept of a graph to attack combinatorial problems beyond graphs (Berge, 1989). A well-known problem of generating all solutions to a problem on hypergraphs is computing all minimal transversals (hitting sets) of a hypergraph $\mathcal{H}$, cf. (Reiter, 1987; Mannila and Räihä, 1991; Eiter and Gottlob, 1991). A minimal transversal is a minimal vertex subset that has a nonempty intersection with each edge; $Tr(\mathcal{H})$ is the hypergraph whose edges are all minimal transversals of $\mathcal{H}$. The minimal transversals correspond 1-1 to the maximal independent sets of $\mathcal{H}$, which are the maximal vertex subsets that contain no edge of $\mathcal{H}$. The standard notion of a polynomial time algorithm for generating all minimal transversals refers to input-polynomiality, which is as follows.

Input-polynomial time: An algorithm meets this criterion if it runs in time polynomial in the input size of $\mathcal{H}$.

It is well-known and easy to see that $\mathcal{H}$ can have exponentially many minimal transversals resp. maximal independent sets in the number of edges of $\mathcal{H}$, and hence they cannot be computed in input-polynomial time. It remains to see whether this is possible in

[†] Mailing address: Paniglgasse 16, A-1040 Wien, Austria. eiter@vexpert.dbai.tuwien.ac.at

'output-polynomial time', i.e., if the output size is accounted in measuring the 'complexity' of such an algorithm; two possible criteria for output-polynomial computation are the following (see Johnson, Yannakakis and Papadimitriou (1988) for a discussion and further criteria):

Total output-polynomial time: An algorithm meets this criterion if it works in time bounded by a polynomial in the input size of $\mathcal{H}$ and the number of minimal transversals (maximal independent sets).

Output with polynomial delay: An algorithm meeting this criterion must output the minimal transversals (maximal independent sets) of $\mathcal{H}$ in some order such that the time until the first output and the time between subsequent outputs (and the time after the last output) is always bounded by a polynomial in the size of $\mathcal{H}$.

Note that the latter criterion strengthens the former. It is open whether a total output-polynomial algorithm for computing $Tr(\mathcal{H})$ resp. all maximal independent sets of a hypergraph exists, cf. (Lawler, Lenstra and Rinnooy Kan 1980; Johnson, Yannakakis and Papadimitriou, 1988; Mannila and Räihä, 1991; Ibaraki and Kameda, 1990; Eiter and Gottlob, 1991).

We describe an polynomial-delay algorithm for lexicographic output of the minimal transversals of xt-hypergraphs, i.e., hypergraphs whose minimal transversals meet each edge in a singleton; for general hypergraphs, this is not possible unless $\mathbf{P} = \mathbf{NP}$. We also show that the maximal independent sets of such a hypergraph can be output in lexicographic order with polynomial delay. Moreover, we show that xt-hypergraphs can be recognized in polynomial time. Notice that deciding whether a hypergraph has any minimal transversal that meets each edge in exactly one vertex is **NP**-complete.

Our results on hypergraphs have an interesting application to the class of Boolean μ-functions, which are definable by Boolean expressions in which no variable occurs repeatedly (called μ-formulas in (Valiant, 1984; Mundici, 1989b)); such formulas have applications in machine learning, for instance (Valiant, 1984; Pitt and Valiant, 1988). The properties of μ-formulas, especially if the Boolean expression is monotone, i.e., built with $\wedge, \vee$ only, have recently been studied in (Hunt III and Stearns, 1986,1990; Mundici 1989a,1989b); monotone μ-formulas correspond to series-parallel networks (cf. Mundici, 1991). Mundici showed that deciding whether a monotone Boolean expression E is equivalent to some monotone μ-formula is in co-**NP**. We show that this problem is polynomial if E is in CNF or DNF, and that the problem is co-**NP**-hard in the general case. We also show that the dualization problem, i.e., computing from the prime implicants of a Boolean function the prime implicants of the dual function, as well as a variant thereof can be solved in case of monotone Boolean μ-functions by polynomial delay algorithms.

The rest of the paper is organized as follows. Section 2 introduces preliminaries and notation. Section 3 treats recognition of xt-hypergraphs, while Section 4 considers minimal transversals and maximal independent sets computation. The results of Sections 3,4 are applied in Section 5 to Boolean μ-functions. Section 6 concludes the paper.

2. Preliminaries and Notation

DEFINITION 2.1. *A hypergraph is a pair* $(V, \mathcal{E})$ *of a finite vertex set* $V = \{v_1, \ldots, v_n\}$ *and a family* $\mathcal{E} = \{E_1, \ldots, E_m\}$ *of subsets of* V*, called hyperedges or simply edges.*

We will identify a hypergraph $\mathcal{H} = (V, \mathcal{E})$ with $\mathcal{E}$ if V is understood, and we will write

$H \in \mathcal{H}$ for $H \in \mathcal{E}$, $\mathcal{H}_1 \cup \mathcal{H}_2$ for $(V_1 \cup V_2, \mathcal{E}_1 \cup \mathcal{E}_2)$ where $\mathcal{H}_i = (V_i, \mathcal{E}_i)$, $i = 1, 2$ if there is no danger of ambiguity.

We assume that sets (and edges) are stored as bit vectors, and hence the problem size $S = S(\mathcal{H})$ of a hypergraph $\mathcal{H}$ on V is of order $\Theta(|V||\mathcal{E}|)$.

For every hypergraph $\mathcal{H}$, let $V_e(\mathcal{H}) = \{v : \exists H \in \mathcal{H} \text{ s.t. } v \in H\}$ denote the *essential vertices* of $\mathcal{H}$. A hypergraph $\mathcal{H}$ is called *simple* iff $H \subseteq H' \Rightarrow H = H'$ for each $H, H' \in \mathcal{H}$. In particular, $\emptyset$ is simple.

Let for each hypergraph $\mathcal{H}$ on V denote $\min(\mathcal{H})$ the hypergraph on V with edge set $\{H \in \mathcal{H} : \not\exists H' \in \mathcal{H} \text{ s.t. } H' \subset H\}$. Clearly, $\min(\mathcal{H})$ is simple.

The *star* $St(v, \mathcal{H})$ of vertex v in the hypergraph $\mathcal{H}$ on V is the hypergraph $(V, \{H \in \mathcal{H} : v \in H\})$.

DEFINITION 2.2. *A set $T \subseteq V$ is called a transversal* (Berge, 1989) *(or hitting set) of hypergraph $\mathcal{H}$ on V iff $\forall H \in \mathcal{H} : T \cap H \neq \emptyset$, and T is called minimal iff for all $T' \subset T$, T' is not a transversal of $\mathcal{H}$. The transversal hypergraph $Tr(\mathcal{H})$ of $\mathcal{H}$ is the hypergraph on V with edge set $\{T : T$ is a minimal transversal of $\mathcal{H}\}$. In particular, $Tr(\emptyset) = \{\emptyset\}$ and $Tr(\{\emptyset\}) = \emptyset$.*

The following properties of $Tr(\mathcal{H})$ are well-known (cf. Berge, 1989).

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

PROPOSITION 2.2. *Let $\mathcal{G}$,$\mathcal{H}$ be simple hypergraphs. Then $Tr(Tr(\mathcal{H})) = \mathcal{H}$ and $Tr(\mathcal{G}) = Tr(\mathcal{H})$ iff $\mathcal{G} = \mathcal{H}$.*

COROLLARY 2.3. *Let $\mathcal{H}$ be a simple hypergraph on V, and let $v \in V$. Then there exists $T \in Tr(\mathcal{H})$ such that $v \in T$ iff $v \in V_e(\mathcal{H})$.*

DEFINITION 2.3. *A set $I \subseteq V$ is an independent set of a hypergraph $\mathcal{H}$ on V iff $\forall H \in \mathcal{H} : H \not\subseteq I$, and I is called maximal iff no $I' \supset I$ is an independent set of $\mathcal{H}$.*

Clearly, I is a maximal independent set of $\mathcal{H}$ iff $V - I \in Tr(\mathcal{H})$ holds.

Given a linear order $<$ on $V = \{v_1, \ldots, v_n\}$, $v_1 < v_2 < \cdots < v_n$, the *lexicographic order* $<_l$ on the powerset $\mathcal{P}(V)$ is defined as follows: $V_1 <_l V_2$ iff the least element in $(V_1 - V_2) \cup (V_2 - V_1)$ is in V_1 (Johnson, Yannakakis and Papadimitriou, 1988). Johnson *et al.* showed the following.

PROPOSITION 2.4. *Finding the lexicographically last maximal independent set of a hypergraph (even graph) $\mathcal{H}$ is* **NP***-hard.*

It is easily checked that $V_1 <_l V_2$ iff $V - V_2 <_l V - V_1$ holds. Hence

COROLLARY 2.5. *Finding the lexicographically first minimal transversal of a hypergraph (even graph) $\mathcal{H}$ is* **NP***-hard.*

Thus, unless $\mathbf{P} = \mathbf{NP}$, there is no polynomial delay algorithm that outputs the minimal transversals of a hypergraph (even a graph) in lexicographic order. (For graphs, (Johnson, Yannakakis and Papadimitriou, 1988) entails a clever polynomial delay algorithm for inverse lexicographic ordering.)

Call a transversal T of hypergraph $\mathcal{H}$ *exact* iff $|T \cap H| = 1$, for each $H \in \mathcal{H}$.

DEFINITION 2.4. *A hypergraph $\mathcal{H}$ is an exact transversal (xt-)hypergraph iff each $T \in Tr(\mathcal{H})$ is an exact transversal of $\mathcal{H}$. In particular, $\emptyset$ and $\{\emptyset\}$ are xt-hypergraphs.*

Notice that deciding if *any* $T \in Tr(\mathcal{H})$ is exact is **NP**-complete, since deciding if $\mathcal{H}$ has an exact transversal is well-known **NP**-complete (Karp, 1972). The similar xt-hypergraph recognition problem is as follows. Given a hypergraph $\mathcal{H}$, decide whether $\mathcal{H}$ is an xt-hypergraph. This problem is in a sense dual to deciding if some minimal exact transversal exists, and turns out to be polynomial.

We conclude this section with additional notation. For any hypergraph $\mathcal{H}$ on V and $V' \subseteq V$, $\mathcal{H}[V']$ denotes the hypergraph $(V, \{H \cap V' : H \in \mathcal{H}\})$, and for $v \in V$, $R(\mathcal{H}; v; V')$ denotes the hypergraph $(V, \{H - V' : H \in \mathcal{H}, v \notin H\})$. That is, $\mathcal{H}[V']$ contains all edges of $\mathcal{H}$ projected to V', and $R(\mathcal{H}; v; V')$ contains all edges of $\mathcal{H}$ that do not contain v, projected to $V - V'$.

3. Recognizing Exact Transversal Hypergraphs

We show in this section that recognizing an xt-hypergraph is polynomial. We note the following properties of xt-hypergraphs.

LEMMA 3.1. *Let $\mathcal{H}$ be a hypergraph on vertices V, and let $v \in V$, $H \in \mathcal{H}$ such that $v \in H$, and let $\mathcal{H}' = R(\mathcal{H}; v; H)$. Then $Tr(\mathcal{H}') = \{T - \{v\} : T \in Tr(\mathcal{H}), T \cap H = \{v\}\}$.*

PROOF. It is easily checked that this holds if $\mathcal{H}' = \{\emptyset\}$, which is the case iff $H' \subseteq H - \{v\}$ exists such that $H' \in \mathcal{H}$. For the rest, we exclude this case.

Let $T \in Tr(\mathcal{H}')$. Clearly, $T \cup \{v\}$ is a transversal of $\mathcal{H}$. Furthermore, $T \cup \{v\} \in Tr(\mathcal{H})$ holds; for if not, since $T \cap H = \emptyset$ clearly holds, $T' \subset T$ must exist such that $T' \cup \{v\} \in Tr(\mathcal{H})$; since T' is a transversal of $\mathcal{H}'$, this contradicts $T \in Tr(\mathcal{H}')$. On the other hand, if $T \in Tr(\mathcal{H})$ such that $T \cap H = \{v\}$, then $T - \{v\}$ must be a transversal of $\mathcal{H}'$. Assume $T - \{v\} \notin Tr(\mathcal{H}')$ holds. Then $T' \in Tr(\mathcal{H}')$ with $T' \subset T - \{v\}$ exists, and since $T' \cup \{v\}$ is a transversal of $\mathcal{H}$, this contradicts $T \in Tr(\mathcal{H})$. Hence $Tr(\mathcal{H}') = \{T - \{v\} : T \in Tr(\mathcal{H}), T \cap H = \{v\}\}$. $\Box$

COROLLARY 3.2. *Let $\mathcal{H}$ be an xt-hypergraph on vertices V, and let $v \in V$, $H \in \mathcal{H}$ such that $v \in H$, and let $\mathcal{H}' = R(\mathcal{H}; v; H)$. Then $Tr(\mathcal{H}') = \{T - \{v\} : T \in Tr(\mathcal{H}), v \in T\}$ and $\mathcal{H}'$ is an xt-hypergraph.*

PROOF. Follows easily from Lemma 3.1 and the definition of xt-hypergraphs. $\Box$

Consider the algorithm *XTREC* in Figure 1.

THEOREM 3.3. *On input hypergraph $\mathcal{H}$ on vertices V, XTREC correctly outputs whether $\mathcal{H}$ is an xt-hypergraph in time $O(mS^2)$, where $m = |\mathcal{H}|$.*

PROOF. From Lemma 3.1, Proposition 2.1, and Corollary 2.3 it is easily verified that

```
Algorithm XTREC(H);
    input:     a hypergraph H.
    output:    true if H is an xt-hypergraph, otherwise false.
    begin
        for v ∈ V_e(H) do
            F := ⋃_{H ∈ St(v,H)} H;
            for H ∈ St(v,H) do
                H' = min(R(H; v; H));
                if F ∩ V_e(H') ≠ ∅ then output(false);
                    stop;
                fi;
            endfor;
        endfor;
        output(true);
    end.
```

Figure 1. Algorithm *XTREC*.

XTREC outputs *false* iff there exist $v, v' \in V$, $H, H' \in \mathcal{H}$ and $T \in Tr(\mathcal{H})$ such that $v' \notin H$, $T \cap H = \{v\}$, $\{v, v'\} \subseteq H'$, and $\{v, v'\} \subseteq T$ holds. However, this is clearly the case if and only if T is not an xt-hypergraph.

Let $n = |V|$. Note that $V_e(\mathcal{H})$, F are computable in $O(nm)$ time (assume that set operations on vertex sets take linear time.) Clearly, $\mathcal{H}' = \min(R(\mathcal{H}; v; H))$ is computable in time $O(nm^2)$, and $V_e(\mathcal{H}')$ is computable in $O(nm)$ time. The inner loop is executed no more than m times, and the outer no more than n. Thus it is not hard to see that the runtime of *XTREC* is bounded by $O(n^2m^3) = O(mS^2)$. $\Box$

4. Generating all Minimal Transversals and all Maximal Independent Sets

In this section, we show that the minimal transversals as well the maximal independent sets of an xt-hypergraph $\mathcal{H}$ can be output in lexicographic order with polynomial delay. Notice that $\mathcal{H}_n = \{\{v_{2i-1}, v_{2i}\} : 1 \leq i \leq n\}$ on $V = \{v_1, \ldots, v_{2n}\}$, $n \geq 0$, is an xt-hypergraph and that $Tr(\mathcal{H}_n) = \{\{w_1, \ldots, w_n\} : w_i \in \{v_{2i-1}, v_{2i}\}\}$, $|Tr(\mathcal{H}_n)| = 2^n$, thus no input-polynomial algorithms for these problems exist.

Lemma 4.1. *Let $\mathcal{H}$ be an xt-hypergraph on vertices V, and let $v \in V$. Then $\mathcal{H}' = \mathcal{H}[V - \{v\}]$ is an xt-hypergraph and $Tr(\mathcal{H}') = \{T \in Tr(\mathcal{H}) : T \subseteq V - \{v\}\}$.*

Proof. It is easily checked that this holds if $\mathcal{H}' = \{\emptyset\}$. For the rest, we exclude this case. Clearly, every $T \in Tr(\mathcal{H})$ such that $T \subseteq V - \{v\}$ is a transversal of $\mathcal{H}'$; furthermore $T \in Tr(\mathcal{H}')$ holds: for if $T' \subset T$ is a transversal of $\mathcal{H}'$, then T' is a transversal of $\mathcal{H}$ which contradicts $T \in Tr(\mathcal{H})$. On the other hand, every $T \in Tr(\mathcal{H}')$ fulfills $T \subseteq V - \{v\}$ and is a transversal of $\mathcal{H}$. Clearly, $T \in Tr(\mathcal{H})$ holds, for if $T' \subset T$ with $T' \in Tr(\mathcal{H})$ exists, $T \notin Tr(\mathcal{H}')$ follows. Thus $Tr(\mathcal{H}') = \{T \in Tr(\mathcal{H}) : T \subseteq V - \{v\}\}$. Since $\mathcal{H}$ is an xt-hypergraph, it is not hard to see that $\mathcal{H}'$ is an xt-hypergraph. $\Box$

Theorem 4.2. *Let $\mathcal{H}$ be an xt-hypergraph on vertices V, and let $v \in V$, $H \in \mathcal{H}$ such*

```
Algorithm XTR(U, H);
    input:   a finite set U and a hypergraph H, U ∩ V_e(H) = ∅.
    output:  {U ∪ T : T ∈ Tr(H)} in lexicographic order.
    begin
        if (H = ∅) then output(U);
        else
            if (∅ ∉ H) then
                select least v ∈ V_e(H) and H ∈ St(v, H);
                XTR(U ∪ {v}, R(H; v; H));
                XTR(U, H[V − {v}]);
            fi;
        fi;
    end.
```

Figure 2. Algorithm *XTR*.

that $v \in H$. Then, $Tr(\mathcal{H}) = \mathcal{H}_v^+ \cup \mathcal{H}_v^-$, where $\mathcal{H}_v^+ = \{T \cup \{v\} : T \in Tr(R(\mathcal{H}; v; H))\}$ and $\mathcal{H}_v^- = Tr(\mathcal{H}[V - \{v\}])$.

PROOF. Follows immediately from Corollary 3.2 and Lemma 4.1. □

Now consider the algorithm *XTR* in Figure 2.

Recall that finding the lexicographically first minimal transversal of an arbitrary hypergraph (even graph) is **NP**-hard.

THEOREM 4.3. *Let $\mathcal{H}$ be an xt-hypergraph on V. Then, $XTR(\emptyset, \mathcal{H})$ outputs the minimal transversals of $\mathcal{H}$ in lexicographic order with $O(nS)$ output delay, where $n = |V|$ and S is the input size.*

PROOF. Let $n_e(\mathcal{H}) = |V_e(\mathcal{H})|$. We show by induction on $n_e(\mathcal{H})$ that if $\mathcal{H}$ is an xt-hypergraph and $U \cap V_e(\mathcal{H}) = \emptyset$, then $XTR(U, \mathcal{H})$ outputs the elements of $\{U \cup T : T \in Tr(\mathcal{H})\}$ in lexicographic order. For the base $n_e = 0$, this is obvious. The inductive step $n_e > 0$ is verified as follows. We have $\mathcal{H} \neq \emptyset$. If $\emptyset \in \mathcal{H}$, the induction statement clearly holds; if $\emptyset \notin \mathcal{H}$, then by Corollary 3.2 and Lemma 4.1 and the induction hypothesis, the sets in $\{U \cup \{v\} \cup T : T \in Tr(R(\mathcal{H}; v; H))\}$ and $\{U \cup T : T \in Tr(\mathcal{H}[V - \{v\}])\}$ respectively, are output in lexicographic order; since v is the least vertex in $V_e(\mathcal{H})$, by Theorem 4.2 $\{U \cup T : T \in Tr(\mathcal{H})\}$ is output in lexicographic order, and the induction statement holds. Thus $XTR(\emptyset, \mathcal{H})$ outputs the minimal transversals of $\mathcal{H}$ in lexicographic order.

Concerning the polynomial delay property, let n_f, n_b, n_l denote the number of recursive calls of *XTR* before the first output, between outputs and after the last output upon call of $XTR(U, \mathcal{H})$, respectively. (n_f, n_b, n_l depend only on $\mathcal{H}$.) It can be shown by induction on n_e again that $n_f(\mathcal{H}) \leq n_e(\mathcal{H})$, $n_b(\mathcal{H}) \leq 2n_e(\mathcal{H})$, and $n_l(\mathcal{H}) \leq n_e(\mathcal{H})$ holds (smaller bounds exist though). The base $n_e = 0$ is trivial. For the inductive step $n_e > 0$, if $\emptyset \in \mathcal{H}$, then the induction statement holds. If $\emptyset \notin \mathcal{H}$, then by the induction hypothesis (let $\mathcal{H}_1 = R(\mathcal{H}; v; H)$, $\mathcal{H}_2 = \mathcal{H}[V - \{v\}]$)

$$n_f(\mathcal{H}) \quad = \quad \max\{1 + n_f(\mathcal{H}_1),\ 1 + n_f(\mathcal{H}_2)\} \leq (n_e(\mathcal{H}) - 1) + 1 = n_e(\mathcal{H})$$

$$
\begin{aligned}
n_b(\mathcal{H}) &\leq \max\{n_b(\mathcal{H}_1),\ n_b(\mathcal{H}_2),\ n_l(\mathcal{H}_1)+n_f(\mathcal{H}_2)+1\} \leq 2(n_e(\mathcal{H})-1)+1 \\
&< 2n_e(\mathcal{H}) \\
n_l(\mathcal{H}) &= \max\{n_l(\mathcal{H}_1)+1,\ n_l(\mathcal{H}_2)\} \leq (n_e(\mathcal{H})-1)+1 = n_e(\mathcal{H}),
\end{aligned}
$$

hence the induction statement holds. It is not hard to see that the runtime of the body of *XTR* is bounded by $O(n|\mathcal{H}|)$. Since the number of edges of the hypergraphs in recursive calls monotonically decreases, the $O(nS)$ output delay property of *XTR* clearly follows. □

It is easy to see that if the order of the statements $XTR(U \cup \{v\}, R(\mathcal{H}; v; H))$ and $XTR(U, \mathcal{H}[V-\{v\}])$ in *XTR* is changed, then the resulting algorithm outputs the minimal transversals of $\mathcal{H}$ in inverse lexicographic order, and it is not difficult to show that the same bound for output delay holds. Thus,

THEOREM 4.4. *The minimal transversals of an xt-hypergraph can be output in inverse lexicographic order with $O(nS)$ delay.*

COROLLARY 4.5. *The maximal independent sets of an xt-hypergraph can be output in lexicographic order with $O(nS)$ delay.*

5. Application to Boolean μ-Functions

In this section, we apply the results of the previous sections to Boolean μ-functions. The reader is assumed familiar with Boolean functions (BFs); for a background, see Wegener (1987).

Boolean expressions (BEs) are built using the connectives $\neg, \wedge, \vee, \rightarrow$ from Boolean variables $x_1, \ldots, x_n$ and constants $0, 1$. The BF represented by the Boolean expression E is denoted by f_E. A BE is monotone iff it is built only with $\wedge, \vee$, and a BF is monotone iff it is represented by some monotone BE. We adopt the terminology of Mundici (1989b).

DEFINITION 5.1. *A BE E is a μ-formula iff each variable occurs in E at most once, and a BE F is called μ-equivalent iff F is equivalent to some μ-formula E.*

Call a BF f a *μ-function* iff there is some μ-formula E that represents f. It is easy to see that a monotone μ-function f is represented by a monotone μ-formula (Mundici, 1989b). Hunt III and Stearns (1990) showed that deciding whether a monotone BE E defines a *particular* μ-function is co-**NP**-complete. For deciding whether E defines *any* μ-function, Mundici obtained the following upper bound.

PROPOSITION 5.1. Mundici (1989b). *Given a monotone BE E, deciding if f_E is a μ-function is in co-**NP**.*

This bound is tight, as we will show. We first show that this problem is polynomial if E is in CNF or in DNF. Recall that a BE is in CNF (DNF) iff it is a conjunction of clauses, i.e., disjunctions of literals (a disjunction of monomials, i.e., conjunctions of literals), where a literal is a BE of type x or $\neg x$.

We can think of a prime implicant (prime clause) of a monotone BF $f(x_1, \ldots, x_n)$ as a minimal set $\{y_1, \ldots, y_k\} \subseteq \{x_1, \ldots, x_n\}$ such that $y_1 \wedge \cdots \wedge y_k \rightarrow E$ ($E \rightarrow y_1 \vee \cdots \vee y_k$) is a tautology (in case of the empty set, $1 \to E$ and $E \to 0$, respectively) (cf. Mundici, 1989b). Let $\mathcal{PI}(f)$ and $\mathcal{PC}(f)$ denote the hypergraphs on vertices $\{x_1, \ldots, x_n\}$ with respective edge sets of all prime implicants and prime clauses of f.

Clearly, $\mathcal{PI}(f)$ and $\mathcal{PC}(f)$ are simple hypergraphs, which are well-known related as follows.

LEMMA 5.2. *Let f be a monotone BF. Then, $\mathcal{PI}(f) = Tr(\mathcal{PC}(f))$, $\mathcal{PC}(f) = Tr(\mathcal{PI}(f))$.*

Monotone μ-functions can be characterized by xt-hypergraphs due to a deep structural result by Mundici (1989a) as follows.

PROPOSITION 5.3. *A monotone BF f is a μ-function if and only if $\mathcal{PI}(f)$ is an xt-hypergraph.*

Thus we get the following result.

LEMMA 5.4. *Given $\mathcal{PI}(f)$ or $\mathcal{PC}(f)$ for a monotone BF f, deciding if f is a μ-function is possible in $O(mS^2)$ time.*

PROOF. If $\mathcal{H}$ is a simple hypergraph, then $\mathcal{H}$ is xt iff $Tr(\mathcal{H})$ is xt. Hence the lemma follows immediately from Proposition 5.3, Lemma 5.2, and Theorem 3.3. $\Box$

THEOREM 5.5. *Let E be a monotone BE on variables $x_1, \ldots, x_n$ which is in CNF (DNF). Deciding if f_E is μ-equivalent is possible in time $O(n^2m^3)$, where m is the number of clauses (monoms) of E.*

PROOF. If $E = \bigvee_{i=1}^{m}(x_{i_1} \wedge \cdots \wedge x_{i_{k_i}})$, then $\mathcal{PI}(f_E) = \min(\mathcal{H})$ where $\mathcal{H}$ is the hypergraph on $\{x_1, \ldots, x_n\}$ with edge set $\{\{x_{i_1}, \ldots, x_{i_{k_i}}\} : 1 \le i \le m\}$. (The case of CNF is similar.) Since $\min(\mathcal{H})$ is computable in time $O(nm^2)$, the result follows by Lemma 5.4. $\Box$

The complexity of recognizing μ-functions from monotone BEs not necessarily in DNF or CNF is co-**NP**-complete, however. co-**NP**-hardness results already if DNF and CNF are mixed. We refer in the proof of this result to the following lemma.

LEMMA 5.6. *Let $\mathcal{H}$ be a simple hypergraph on vertices $V = \{v_1, \ldots, v_{2n}\}$, $n \ge 1$, such that (i) $\{v_i, v_{n+i}\} \in \mathcal{H}$, $1 \le i \le n$, and (ii) $\forall H \in \mathcal{H} : |H| \le 2 \Rightarrow H = \{v_i, v_{n+i}\}$ for some $i \in \{1, \ldots, n\}$. Then, $\mathcal{H}$ is an xt-hypergraph if and only if $\mathcal{H} = \{\{v_i, v_{n+i}\} : 1 \le i \le n\}$.*

PROOF. The *if* direction is trivial. The *only-if* direction is shown by contradiction. Assume that $\mathcal{H}$ is an xt-hypergraph but $\mathcal{H} \ne \{\{v_i, v_{n+i}\} : 1 \le i \le n\}$. Hence, $H \in \mathcal{H}$ exists such that $|H| = k > 2$. Without loss of generality we may assume that $H = \{v_1, \ldots, v_k\}$. Indeed, since $\mathcal{H}$ is simple, $\{v_i, v_{n+i}\} \not\subseteq H'$ for $1 \le i \le n$ and all $H' \in \mathcal{H}$, thus $H = \{v_1, \ldots, v_k\}$ can be reached by a vertex renaming. Since $v_1 \in V_e(\min(\mathcal{H}))$, by Proposition 2.1 and Corollary 2.3, there exists a $T \in Tr(\mathcal{H})$ such that $T \cap (H - \{v_1\}) = \emptyset$; it follows that $\{v_{n+2}, \ldots, v_{n+k}\} \subseteq T$. By the xt-property of $\mathcal{H}$, we have that no edge of $\mathcal{H}$ contains two vertices from $\{v_{n+2}, \ldots, v_{n+k}\}$, and hence each edge of $\mathcal{H}$ must contain a vertex from $T' = V - \{v_{n+2}, \ldots, v_{n+k}\}$. Consequently, T' is a transversal of $\mathcal{H}$. Since $T' - \{v_i\}$ is not a transversal of $\mathcal{H}$ for $2 \le i \le k$, it follows that $T'' \in Tr(\mathcal{H})$ exists with

$\{v_2, \ldots, v_k\} \subseteq T'' \subseteq T'$. But $\{v_2, \ldots, v_k\} \subseteq H \cap T''$ and $k > 2$ means that $\mathcal{H}$ is not an xt-hypergraph, a contradiction. □

Notice that without (ii) the lemma no longer holds. E.g., the hypergraph $(V, \{\{v_1, v_3\}, \{v_2, v_4\}, \{v_1, v_2\}, \{v_3, v_4\}\})$ with $V = \{v_1, v_2, v_3, v_4\}$ is an xt-hypergraph.

THEOREM 5.7. *Let E be a monotone BE. Deciding if f_E is a μ-function is co-**NP**-complete. This holds even if $E = E_1 \vee E_2$, where E_1 is in CNF and E_2 is in DNF.*

PROOF. Membership of this problem in co-**NP** is given by Proposition 5.1.

We prove co-**NP**-hardness by a polynomial transformation from deciding if a BE E on distinct variables $x_1, \ldots, x_n$, $n \geq 1$, which is in CNF is a contradiction. Let $y_1, \ldots, y_n$ be new distinct Boolean variables, and define the monotone BE E' as follows.

$$E' = E[y] \vee (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n),$$

where $E[y]$ denotes the BE obtained if in E each occurrence of $\neg x_i$ is replaced by y_i, for $1 \leq i \leq n$. (We remark that this construction is similar to the one used in the proof of Theorem 3.3 from (Hunt III and Stearns, 1990).) Without loss of generality we may assume that there exists no $I \in \mathcal{PI}(f_E)$ such that $|I| \leq 2$. Indeed, this can be decided in polynomial time, hence deciding if E is a contradiction is easily transformable to this subcase in polynomial time.

Under this restriction, $\mathcal{I} = \{\{x_i, y_i\} : 1 \leq i \leq n\} \subseteq \mathcal{PI}(f_{E'})$ holds, and $\mathcal{I} = \mathcal{PI}(f_{E'})$ if and only if E is a contradiction. Since each $I \in \mathcal{PI}(f_{E'}) - \mathcal{I}$ fulfills $|I| > 2$, by Lemma 5.6 and Proposition 5.3 it follows that E' is μ-equivalent if and only if E is a contradiction. Since E' is polynomially constructible, we have the result. □

Another immediate result is obtained for the dualization problem of monotone BFs, which is to compute from $\mathcal{PI}(f)$ the prime implicants of the dual function f^d (cf. Wegener (1987) for dual functions). By well-known properties of duality, $\mathcal{PI}(f^d) = \mathcal{PC}(f)$ holds. Hence by Theorem 4.3,

THEOREM 5.8. *Given the prime implicants of a n-ary monotone μ-function f, the prime implicants of the dual function f^d can be generated in lexicographic order with $O(nS)$ delay.*

A variant of the dualization problem is formulated by Crama (1987) as follows. Let $\leq$ partially order the Boolean n-vectors by $(a_1, \ldots, a_n) \leq (b_1, \ldots, b_n)$ iff $a_i = 1 \Rightarrow b_i = 1$, for all i, and let $f(x_1, \ldots, x_n)$ be a monotone BF. Given the minimal vectors a such that $f(a) = 1$ (the minimal true points, MTPs, of f), find all maximal vectors b such that $f(b) = 0$ (the maximal false points, MFPs, of f). Let for $a = (a_1, \ldots, a_n)$ denote $\sigma(a) = \{x_i : a_i = 1, 1 \leq i \leq n\}$. It is easy to see that a is a MTP of f iff $\sigma(a) \in \mathcal{PI}(f)$ holds, and since f is monotone iff $a \leq b, f(a) = 1 \Rightarrow f(b) = 1$ for all a, b, cf. Wegener (1987), it is not hard to see that a is a MFP of f iff $\sigma(a)$ is a maximal independent set of $\mathcal{PI}(f)$. Define the linear order $<_t$ on Boolean vectors by $(a_1, \ldots, a_n) <_t (b_1, \ldots, b_n)$ iff $a_i = 1$ for the least i with $a_i \neq b_i$. Thus if $x_1 < \cdots < x_n$, $a <_t b$ iff $\sigma(a) <_l \sigma(b)$. Hence by Corollary 4.5, THEOREM 5.9. *Given the MTPs of a n-ary monotone Boolean μ-function f, the MFPs of f can be generated in $<_t$ order with $O(nS)$ delay, where S is the input size.*

6. Conclusion

We have shown that hypergraphs whose minimal transversals are their exact transversals, i.e., those vertex subsets that meet each edge in a singleton, are efficiently recognizable, and that the minimal transversals as well as the maximal independent sets of such hypergraphs can be output in lexicographic order with polynomial delay. Lexicographically sorted output of the minimal transversals is not possible for general hypergraphs unless $\mathbf{P} = \mathbf{NP}$; it is open whether any output total-polynomial algorithm for computing all minimal transversals (equivalently, all maximal independent sets) exists.

We have further applied the results on hypergraphs to monotone Boolean μ-functions, i.e., Boolean functions represented by a Boolean expression with $\wedge, \vee, 0$, 1 in which no variable occurs repeatedly (Hunt III and Stearns, 1986,1990; Mundici, 1989a,1989b). We obtained that such functions are efficiently recognizable from the prime implicants resp. prime clauses. However, the recognition problem from general monotone expressions was shown co-**NP**-hard. This complements the upper bound of membership in co-**NP** obtained by Mundici (1989b), and shows that Mundici's bound is tight.

Finally, we considered the dualization problem for monotone Boolean μ-functions. Our results imply a polynomial delay algorithm for this problem, thus showing that the problem can be solved efficiently with respect to the combined size of the input and the output. This result extends the classes of Boolean functions for which such an algorithm is known, cf. Crama (1987).

Acknowledgement

The author is grateful to Georg Gottlob and an anonymous referee for their comments on this paper.

References

Berge, C. (1989). *Hypergraphs*. Elsevier Science Publishers B.V. (North-Holland).

Crama, Y. (1987). Dualization of Regular Boolean Functions. *Discrete Applied Mathematics*, 16:79–85.

Eiter,T., Gottlob, G. (1991). Identifying the Minimal Transversals of a Hypergraph and Related Problems. Technical Report CD-TR 91/16, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, January 1991.

Hunt III, H., Stearns, R., (1986). Monotone Boolean Functions, Distributive Lattices, and the Complexities of Logics, Algebraic Structures, and Computation Structures (Preliminary Report). In *Proceedings STACS-86*, Springer LNCS 210, pages 277–287.

Hunt III, H., Stearns, R., (1990). The Complexity of Very Simple Boolean Formulas With Applications. *SIAM J. Comp.*, 19(1):44–70.

Ibaraki, T., Kameda, T. (1990). Theory of Coteries. Technical 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.

Johnson, D. S., Yannakakis, M., Papadimitriou, Ch. H. (1988). On Generating All Maximal Independent Sets. *Information Processing Letters*, 27:119–123.

Karp, R. (1972). Reducibility Among Combinatorial Problems. In R. Miller and J. W. Thatcher, editors, *Complexity Of Computer Computations*, pages 85–103, New York: Plenum Press.

Lawler, E., Lenstra, J., Rinnooy Kan, A. H. G. (1980). Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM J. Comp.*, 9:558–565.

Mannila, H., Räihä, K.-J. (1991). Algorithms for Inferring Functional Dependencies. Technical Report C-1991-41, University of Helsinki, Dept. of Comp. Sc., Series of Publ. C, April 1991.

Mundici, D. (1989a). Functions Computed by Monotone Boolean Formulas With No Repeated Variables. *Theoretical Computer Science*, 66:113–114.

Mundici, D. (1989b). Reducibility of Monotone Formulas to μ-Formulas. In *3rd Workshop Computer Science Logic (CSL)*, Springer LNCS 440, pages 267–270.

Mundici, D. (1991). Solution of Rota's Problem on the Order of Series-Parallel Networks. *Advances in Applied Mathematics*, 12:455–463.

Pitt, L., Valiant, L. (1988). Computational Limitations on Learning from Examples. *Journal of the ACM*, 35:965–984.

Reiter, R. (1987). A Theory of Diagnosis From First Principles. *Artificial Intelligence*, 32:57–95.

Valiant, L. (1984). A Theory of the Learnable. *Commun. ACM*, 27(11):1134–1142.

Wegener, I. (1987). *The Complexity of Boolean Functions.* Wiley - Teubner series in computer science. Stuttgart: Teubner and Wiley & Sons Ltd.