On Resolution Like Proofs of Monotone Self-Dual Functions

Daya Ram Gaur
Department of Mathematics and Computer Science
University of Lethbridge, Lethbridge, AB, Canada, T1K 3M4
gaur@cs.uleth.ca

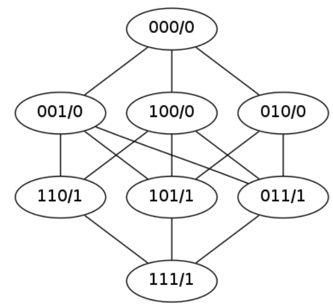
Abstract

We examine the time complexity of resolution like proofs for monotone self-dual functions. For a class of monotone boolean functions that strictly contain the class of self-dual functions we give a new characterization of self-duality. We show that an obvious implementation of a resolution like proof system has an exponential lower bound on the number of step in resolution proofs for monotone self-dual functions. We pose some questions, resolution of which would further the program on self-dual monotone boolean functions. General lower bounds on the length of resolution like proofs for self-dual monotone boolean functions is an interesting open question.

Introduction

We consider boolean functions in disjunctive normal form. A clause is defined as a conjunction of literals, and a disjunction of the clauses is a boolean function in disjunctive normal form. For two vectors $x, y \in \{0,1\}^n$ we say $x \leq y$ if and only if $x_i \leq y_i$ for all $i \in [1..n]$. A boolean function f is said to be monotone if $f(x) \leq f(y)$ for all $x \leq y$. Monotone boolean functions can be expressed in disjunctive normal form without any negative literals in the clauses. For sake of convenience we will represent the variable x_i in the function using the integer i. Following is a monotone boolean function M_3 over three variables $(1 \wedge 2) \vee (1 \wedge 3) \vee (2 \wedge 3)$. A boolean function f is said to be self-dual if and only if $f(x) = \overline{f(\bar{x})}$ for every input x. Central to our inquiry is the question whether a given monotone function f (in disjunctive normal form) is self-dual.

Lattice on the right shows the value of the function M_3 on each input x. Each node in the lattice is labeled $x/f(x)$ where x is the input vector and $f(x)$ is the value of the function on the vector x. It can be verified that for all inputs x, the value of the function M_3 is the opposite of the value on $\bar{x}$. Therefore the majority function on three variables M_3 is self-dual.



Dual of a function f can be obtained by exchanging $\wedge$ and $\vee$ and the constants T and F. Dual of M_3^d is $(1 \vee 2) \wedge (1 \vee 3) \wedge (2 \vee 3)$. M_3 is self-dual if and only if the propositional statement $S = M_3 \equiv M_3^d$ is a tautology. Self-dual monotone boolean functions have interesting applications spanning seemingly diverse areas in computer science, areas such as database theory, machine learning, distributed systems, game theory (please see (Eiter and Gottlob 2002; Eiter, Makino, and Gottlob 2008; Hagen 2008)). The complexity of determining whether a given monotone boolean function is self-dual is unlikely to be NP-complete as Fredman and Khachiyan (Fredman and Khachiyan 1996) describe an $O(n^{o(\log n)})$ algorithm for the problem. The self-duality problem is known be in co-NP, as there exists a short certificate in case the function is not self-dual. The certificate is a vector x such that $f(x) = f(\bar{x})$. It is known that monotone self-duality can be solved by polynomial time algorithms that use poly logarithmic many non deterministic steps (Eiter, Gottlob, and Makino 2003; Kavvadias and Stavropoulos 2003). Using a decomposition due to Boros and Makino (Boros and Makino 2009), recently it was established that monotone self-duality can be decided in $O(\log^2 n)$ space (Gottlob 2013). The decomposition in (Gaur and Krishnamurti 2004) also yields a logarithmic depth search tree, and could have been used to obtain the quadratic log space bound (Gottlob 2013). However the exact complexity of determining whether an arbitrary monotone function is self-dual is still open. For several interesting special classes of monotone boolean functions, self-duality can be determined in polynomial time (Makino 2003; Gaur and Krishnamurti 2008). Boros et. al. (Boros, Elbassioni, and Makino 2010) show that many sub-classes of monotone formulas such as bounded size, bounded degree, bounded intersection, bounded con-formality, read once formula can be dualized in polynomial or quasi-polynomial time using Berge multiplication of clauses and simplification using the absorption law. The concept of self-duality has recently been used to construct nano-circuits (Cui and Lieber 2001). Altun and Riedel (Altun and Riedel 2012) use the concept of self-duality to give efficient algorithms for constructing boolean circuits using lattice of four-terminal switches. Lattices of four-terminals can be constructed using nano wires (Cui and Lieber 2001) this facilitates the implementation of logic functions at the nano level in principle. The literature on the problem is vast and our apologies to works, references to which have been omitted in this brief.

Resolution

Let S be a propositional statement that f a monotone boolean function is not self-dual. As any propositional statement can be converted into conjunctive normal form a resolution proof that S is false can be provided if f is self-dual (Davis and Putnam 1960). In general resolution proofs have exponential size (Haken 1985). This motivates the question whether a "resolution like" proof can be provided to establish that a monotone function is self-dual. As we will see in a moment; a proof that shows that f is not self-dual is quite short and simple. However no such short proof is known for establishing that a monotone f is self-dual.

We will exploit the connection between hypergraph coloring and monotone self-duality, and results of Linial and Tarsi (Linial and Tarsi 1985) and Seymour (Seymour 1974) to study the complexity of resolution like proofs for proving self-duality. First we begin with some definitions. An hypergraph (V,E) is a collection E of subsets of V. Elements of V are referred to as the vertices and the elements of E are referred to as the edges of the hypergraph. Given a monotone function f in disjunctive normal form, we associate with f an hypergraph H_f as follows: the variables in f are the vertices in H_f, and each disjunct in f is an edge in H_f. A hypergraph is said to be a sperner if for all edges E, E' $E \not\subseteq E'$ (no edge contains any other edge). A hypergraph is said to be intersecting if every pair of edges have a vertex in common. It is well known that the class of hypergraphs that correspond to self-dual monotone boolean functions are sperner and intersecting.

The hypergraph H_{M_3} associated with function M_3 is $(V = \{1,2,3\}, E = \{\{1,2\},\{1,3\},\{2,3\}\})$. For typographical reasons at times we will use $[,]$ instead of $\{,\}$ to enclose the elements in a set. A 2-coloring of an hypergraph is function $\chi : V \to \{0,1\}$ such that for every edge there are two vertices u,v with $\chi(v) \neq \chi(u)$. In other words a 2-coloring of an hypergraph is an assignment of colors $0,1$ to each vertex such that no edge contains vertices of a single color. We will consider monotone functions of atleast two variables.

Following was noted by several authors and is well known.

Theorem 1 (Eg. (Berge 1984), (Benzaken 1980) ,(Garcia-Molina and Barbara 1985),(Ibaraki and Kameda 1993),(Eiter and Gottlob 1995)), (Crama and Hammer 2011) *A monotone boolean function f such that H_f is intersecting and sperner is self-dual if and only if H_f is not 2-colorable.*

For results regarding the fractional version of the Theorem 1 please see (Gaur and Makino 2009). The hypergraph $[[1,2],[1,3],[2,3]]$ is clearly not 2-colorable hence from the above theorem we conclude that M_3 is self-dual. Given a hypergraph H, a subset S is called *independent* if it does not contain any hyperedge of H. A maximal independent set which does not intersect all the edges is called a *resolvent*.

Example: $[1],[2],[3]$ are all resolvents for $[[1,2],[1,3],[2,3]]$ and no subset of size two or more is a resolvent.

Following is essentially a restatement of the Theorem due to Linial and Tarsi (Linial and Tarsi 1985). For the sake of completeness we provide an independent proof of the statement.

Theorem 2 ((Linial and Tarsi 1985)) *Given a hypergraph $H = (V,E)$ and a resolvent W of H, H is not 2-colorable if and only if $H' = (V, E' = E \cup \{W\})$ is not 2-colorable.*

Proof: Forward direction is easy. H' contains all the edges of H and the resolvent W. If H is not 2-colorable, H' cannot be 2-colorable, as a 2-coloring of H' is also a 2-coloring of H.

To establish the reverse direction assume that H' is not 2-colorable but H is 2-colorable. Let (R,G) be a 2-coloring of H. As H' is not 2-colorable, let us say that edge W is contained in the set R ($W \subseteq R$) without loss of generality. $B \in H, B \cap W = \emptyset$ implies, by the maximality of W that for each $b \in B$ there exists a hyperedge $H_b \in H$ such that $H_b \setminus W = \{b\}$. Note that such a B exists because W is a resolvent. Since H is 2-colored by (R,G) and since $W \subseteq R$, it must be the case that $b \in G$ for each $b \in B$, that is $B \subseteq G$. This contradicts the 2-coloring of H. □

The theorem of Linial and Tarsi gives us the following algorithm to determine whether an arbitrary hypergraph H is 2-colorable. To describe the algorithm we need a function `maximal` that returns the lexicographically smallest maximal subset of the vertices $[1..n]$ that does not contain any edge of the hypergraph H. A maximal W can be constructed using the greedy procedure described below. Starting with empty W, consider the vertices in H in the lexicographic order. If the $W \cup \{v\}$ does not contain any edge of H then add v to W.

```
function C = maximal(H)
W= []
for x in [1..n],
        if  {W union x} does not contain
                              any edge of H
                      W = W union x;
        endif
endfor
endfunction
```

Function `resolve` below adds a maximal W to H until either a resolvent W of size one is found or a maximal W that intersects every edge of H (but does not contain any edge of H) is found. It returns TRUE if the input hypergraph is not 2-colorable FALSE otherwise.

```
function R = resolve(H),
W = maximal(H);
if (W intersects all the edges of H),
        return FALSE; # H is 2-colorable
endif
if (size(W) == 1),
        return TRUE; # H is not 2-colorable
endif
R = resolve(H union W);
endfunction
```

Theorem 3 *Procedure* `resolve` *correctly determines the 2-colorabilty of* H.

Proof: If the function exits on line 4, then H is 2-colorable, $(W, V \setminus W)$ is a 2-coloring. Otherwise H is not 2-colorable and resolvent W is added, and by Theorem 2. the claim holds. □

Example: Consider the hypergraph $[[1,2],[1,3],[2,3]$ associated with function M_3. $W = [1]$ is resolvent that does not intersect edge $B = [2,3]$. By the previous theorem M_3 is self-dual.

Example: Consider the following hypergraph that arises from the Fano plane:

$$[1,2,3],[3,4,5],[1,5,6],[1,4,7],[3,6,7],[2,5,7],[2,4,6]$$

If the function `maximal` constructs W in lexicographic order then following 23 resolvents are added to H.

1 : [1, 2, 4, 5]	13 : [1, 3, 7]
2 : [1, 2, 4]	14 : [1, 3]
3 : [1, 2, 5]	15 : [1, 4, 5]
4 : [1, 2, 6, 7]	16 : [1, 4, 6]
5 : [1, 2, 6]	17 : [1, 4]
6 : [1, 2, 7]	18 : [1, 5, 7]
7 : [1, 2]	19 : [1, 5]
8 : [1, 3, 4, 6]	20 : [1, 6, 7]
9 : [1, 3, 4]	21 : [1, 6]
10 : [1, 3, 5, 7]	22 : [1, 7]
11 : [1, 3, 5]	23 : [1]
12 : [1, 3, 6]	

A list of resolvents added iteratively, until a resolvent of size 1 is added, is referred to as a resolution "like" proof or simply a resolution proof.

Proof complexity

Proof complexity for a propositional formula f in conjunctive normal form is the number of resolution steps in the smallest resolution proof for f. Haken (Haken 1985) was first to establish that the proof complexity of boolean formulas (specifically piegon formulas from (Cook and Reckhow 1979)) is exponential. Here we ask a similar question. For a resolution proof of not 2-colorability, the size of the resolution like proof (or simply resolution proof) is defined as the number of resolvents added. For a fixed monotone self-dual f, the proof complexity of f is the size of the smallest resolution proof that H_f is not 2-colorable.

Example: Consider F_5 below:

$$[[1,2,4],[2,3,4],[1,3,4],[1,2,5],[2,3,5],[1,3,5],[4,5]]$$

The shortest resolution proof below, discovered using breadth first search contains 4 resolvents. The resolvents were added in the order listed. Note that exhaustive search is prohibitive even for the example arising out of the Fano plane due to the high branching factor in the breadth first search.

1: [1,4]	3: [3,4]
2: [2,4]	4: [4]

We refer to the procedure above as the resolution procedure for 2-coloring of hypergraphs. It has been established that for every hypergraph H over n vertices there exists a boolean formula f (not monotone) over n variables such that the f is satisfiable if and only if H 2-colorable (Linial and Tarsi 1985). Furthermore if the resolution for H takes k steps then the corresponding f can be resolved (using propositional resolution) in $2kn$ steps (Linial and Tarsi 1985).

In this section we will study the time complexity of the above resolution procedure `resolve`. We exhibit an infinite family of hypergraphs arising from self-dual monotone boolean formulas for which the resolution using procedure `resolve` takes $O(2^n)$ steps where n is the number of vertices. As the number of distinct resolvents is less than 2^n the procedure `resolve` takes no more than $O(2^n)$ time.

Wheel hypergraphs are defined as follows:

$$W_n = \{\{i,n\} \mid i \in [1,2,\ldots,n-1] \cup \{1,2,\ldots,n-1\}\}$$

$W_3 = M_3 = [[1,3],[2,3],[1,2]]$ and $W_4 = [[1,4],[2,4],[3,4],[1,2,3]]$. Algorithm `resolve` on input W_4 will add the resolvents $[1,2],[1,3],[1]$ and conclude that W_4 is not 2-colorable (hence the corresponding function is self-dual). If we consider the following permutation $[[1,2],[1,3],[1,4],[2,3,4]]$ of W_4 then the `resolve` function will add a single resolvent $[1]$ and conclude that the hypergraph is not 2-colorable. We will show that for W_n the function `resolve` will add $2^{n-2}-1$ resolvents before it concludes that the hypergraph is not 2-colorable. It is well known that family W_n corresponds to a class of monotone boolean functions that are self-dual, also known as wheel coteries.

Theorem 4 *There exists a class of self-dual monotone functions W_n for which there exists a resolution proof of length at least $2^{n-2}-1$.*

Proof: Using induction. Omitted here. □

The length of the resolution proof produced by procedure `resolve` is sensitive to the order of the vertices, as considered by procedure `maximal`. As noted above if the vertices are considered in the order $[n,...]$ then the single resolvent $[n]$ is sufficient to conclude that W_n is not 2-colorable.

Note that the characterization in Theorem 2 is valid for 2-colorability of all hypergraphs.

Remark: Given an intersecting and sperner hypergraph H and a resolvent W, $H \cup \{W\}$ can be made sperner by removal of edges of H that contain the newly added edge W. However the intersection property is lost. The resulting hypergraph is not intersecting anymore, and this loss in the structure is a stumbling stone for any inductive argument (or a recursive algorithm). Ideally we want a characterization such that the resulting problem obeys both the structural properties.

Next we strengthen the characterization in Theorem 2 to critically not 2-colorable hypergraphs. A hypergraph H is said to be critically not 2-colorable if H is not 2-colorable but every proper subset of edges of H is 2-colorable. Hypergraphs that correspond to self-dual monotone boolean functions are critically not 2-colorable. We will refer to critically not 2-colorable hypergraphs as critical hypergraphs at times. The class of critical hypergraphs contains all hypergraphs arising from self-dual monotone boolean functions and more. For instance odd cycles are also critical hypergraphs.

Next we describe an extended resolution procedure for the family of critical hypergraphs. For a hypergraph $H=(V,E)$ the restriction of H to $Z \subseteq V$ (written $H|Z$) is

$$H|Z = \{B \cap Z \mid B \in H, B \cap Z \neq \emptyset\}.$$

Given a hypergraph H, A a subset of vertices is said to have *support* S in hypergraph H (or simply *supported*) if there exist edge $B=(b_1,b_2,\ldots,b_k) \in H$ and $B_i \in H, i=1,\ldots,k$ such that $B_i \setminus A = \{b_i\}$ for all $i=1,..,k$.

Example: Consider R_5 below:

$$[[1,2,4],[2,3,4],[1,3,4],[1,2,5],[2,3,5],[1,3,5],[4,5]]$$

Let A be $[1,2,3]$. A has support in H. Edge $B=[4,5]$ is disjoint from A. $B_4=[1,2,4], B_5=[1,2,5]$ are edges in H with the desired property.

A set Z is said to be *well supported* (in H) if every edge A in the hypergraph $H|Z$ has a support in H.

Example: Give R_5, let us consider $Z=[1,2,3]$. $H|[1,2,3]$ contains edges $[[1,2],[1,3],[2,3]]$. $B=[4,5]$ does not intersect Z. For edge $[i,j]$ in $H|[1,2,3]$, $B_4=[i,j,4], B_5=[i,j,5]$ are the supporting edges in H. Therefore Z is well supported (in H).

Theorem 5 *Let H be a critical hypergraph and W a well supported set in H (H possibly corresponds to a self-dual monotone function f). H is not 2-colorable if and only if $H|W$ is not 2-colorable.*

Proof:

Forward direction follows from a result of Seymour (Seymour 1974).

Reverse direction: We use our definition of well-supported sets and the idea due to Linial and Tarsi (Linial and Tarsi 1985). To derive a contradiction assume that $H|W$ is not 2-colorable but H is 2-colorable. Let (R,G) be the 2-coloring of H. The reason why (R,G) is not a 2-coloring of $H|W$ is because, say R contains some edge A of $H|W$. Note that A is a subset of W. Given that W has support in H, consider the edge $B=(b_1,\ldots,b_k) \in H$ and edge $B_i \in H$ for all $i=1,..,k$, such that $B_i \setminus A = \{b_i\}$ for all $i=1,..,k$. As $A \subseteq R$, each $b_i \in G$. Therefore $B \subseteq G$. This contradicts the fact that (R,G) is a 2-coloring of H. □

```
function R = extendedResolution(H),
W = support(H); # W is well-supported in H
# Each edge in H|W is supported in H
if (W intersects all edges of H),
        return FALSE; # H is 2-colorable
endif
if (size(W) == 1),
        return TRUE; # H is not 2-colorable
endif
# check for H restricted to W
R = extendedResolution(H|W);
endfunction
```

Remark: The resolution procedure above works for all hypergraphs that are critically not 2-colorable and for which a well supported set W can be easily identified in every recursive step. Note that if H does not have well-supported set and H is not 2-colorable then the algorithm does not handle the input instance. The extended resolution procedure described above can provide a two step resolution proof for each formula in the family W_n described above. Each R_{2k+1} for $k \geq 1$ is reduced to R_3 using the procedure `extendedResolution`. A resolution proof for R_3 is computed using the `resolve` procedure.

Recall that that last edge in W_n is $[1,2,\ldots,n-1]$. Consider $W=[n]$ (this W is maximal and lexicographically the largest). It can be observed that every edge in $H|W=[n]$ set W has support in W_n. $B=[1,2,\ldots,n-1]$ is disjoint from W. $B_i=[i,n]$ is an edge in H for all i. Since all edges in $H|W$ have support in H, we check for 2-colorability of $H|W$. $H|W$ has edge of size 1, therefore we can infer that W_n in not 2-colorable. This resolution proof takes only two steps for all n. We were able to correctly guess W for the example above. In general it might not be easy to identify a set of vertices W

(such that each edge in $H|W$ has support in H) over which to resolve H. A super polynomial lower bound on the number of steps in the minimum length resolution like proofs is not known for critical hypergraphs. Hence the next question.

Question: What is the complexity of resolution proofs for critical hypergraphs?

Next we show that if the hypergraph H belongs to the family R described below, then the resolvent W in the Theorem above can be identified efficiently in each recursive step. Hence the constant size resolution proofs can be provided using the `extendedResolution` procedure on R_3. We do not account for the $O(n)$ steps needed to reduce the number of variables to 3.

Let R_n be a monotone self-dual function, defined recursively as follows. R_n is obtained from R_{n-2}. R_n is defined only for odd $n \geq 3$. For every edge E in R_{n-2} introduce two new edges $[E, n-1], [E, n]$. Also add one edge $[n-1, n]$. The base case is $R_3 = [[1,2],[1,3],[2,3]]$. $R_5 = [[1,2,4],[2,3,4],[1,3,4],[1,2,5],[2,3,5],[1,3,5],[4,5]]$. R_n is intersecting and sperner. The hypergraph obtained by restricting R_n to vertices in $[1..n-2]$, $R_n|[1..n-2]$ is the same as the hypergraph R_{n-2} and is intersecting and sperner, by construction.

Lemma 1 *Given R_n, a) The set $W = V \setminus [n, n-1]$ is well supported in R_n . b) $R_{n-2} = R_n|[1..n-2]$ is intersecting and sperner.*

Proof: a) Recall that W is well supported in H if every edge $e \in H|W$ is supported in H. Consider an edge $e \in R_n|W$, edge e does not contain either vertex n or $n-1$ therefore edge $B = [n-1, n]$ is disjoint from e. By construction there are edges $[n-1, W_1]$ and $[n, W_2]$ in R_n such that $W_1, W_2 \subseteq W$.

b) Let $E_n(E_{n-1})$ be the edges in $E \setminus [n, n-1]$ of R_n that contain vertices $n(n-1)$ respectively. By construction $E_n|W == E_{n-1}|W$, i. e. every edge in $E_n|W$ is also in $E_{n-1}|W$ and vice-versa. These are precisely the edges in R_{n-2}. By definition R_{n-2} is intersecting and sperner. □

The previous Lemma is critical as it will help us reduce an intersecting sperner hypergraph R_n to a hypergraph R_{n-2} that are intersecting and sperner such that the original hypergraph is not 2-colorable if and only if resulting hypergraph is not 2-colorable. Theorem 5 reduces the problems of not 2-colorability of R_n to a proof of not 2-colorability of R_{n-2}. Thus a constant size resolution proof of R_3 shows that R_n is not 2-colorable.

There are self-dual functions for which no subset is well supported, Fano plane is one such example. That is we cannot use Theorem 5 to reduce the size of the input instance. The family of hypergraphs with well-supported sets seems to be sparse. This motivates the following definition. Let H be defined strongly critical if H is critical, and every subset of vertices W is not well-supported. Following question is now natural.

Question: What is the complexity of resolution proofs for strongly critical hypergraphs?

Acknowledgements

The author would like to thank the reviewers for detailed comments.

Conclusions

Two important questions remain open. It is not known whether shortest resolution style proofs for 2-colorability for monotone self-dual functions are exponential in size. A relationship between the size of the proof and the width of a resolution proof for propositional formulas in conjunctive normal form was established in (Ben-Sasson and Wigderson 2001). This relationship reduces the task of lower bounding the length of the resolution proofs, to the task of placing upper bounds on the width of a proof. It would be interesting to see if such a size-width relationship also exists for resolution like proofs for monotone self-dual functions.

References

Altun, M., and Riedel, M. D. 2012. Logic synthesis for switching lattices. *Computers, IEEE Transactions on* 61(11):1588–1600.

Ben-Sasson, E., and Wigderson, A. 2001. Short proofs are narrow-resolution made simple. *Association for Computing Machinery. Journal of the Association for Computing Machinery* 48(2):149.

Benzaken, C. 1980. Critical hypergraphs for the weak chromatic number. *Journal of Combinatorial Theory, Series B* 29(3):328–338.

Berge, C. 1984. *Hypergraphs: combinatorics of finite sets*. North Holland.

Boros, E., and Makino, K. 2009. A fast and simple parallel algorithm for the monotone duality problem. In *ICALP (1)*, 183–194.

Boros, E.; Elbassioni, K.; and Makino, K. 2010. Left-to-right multiplication for monotone boolean dualization. *SIAM Journal on Computing* 39(7):3424–3439.

Cook, S. A., and Reckhow, R. A. 1979. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic* 44(1):36–50.

Crama, Y., and Hammer, P. L. 2011. *Boolean functions: theory, algorithms, and applications*. Cambridge University Press.

Cui, Y., and Lieber, C. M. 2001. Functional nanoscale electronic devices assembled using silicon nanowire building blocks. *Science* 291(5505):851–853.

Davis, M., and Putnam, H. 1960. A computing procedure for quantification theory. *Journal of the ACM (JACM)* 7(3):201–215.

Eiter, T., and Gottlob, G. 1995. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing* 24(6):1278–1304.

Eiter, T., and Gottlob, G. 2002. Hypergraph transversal computation and related problems in Logic and AI. In Flesca, S.;

Greco, S.; Ianni, G.; and Leone, N., eds., *Logics in Artificial Intelligence*, volume 2424 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 549–564.

Eiter, T.; Gottlob, G.; and Makino, K. 2003. New results on monotone dualization and generating hypergraph transversals. *SIAM J. Comput.* 32(2):514–537.

Eiter, T.; Makino, K.; and Gottlob, G. 2008. Computational aspects of monotone dualization: A brief survey. *Discrete Applied Mathematics* 156(11):2035–2049.

Fredman, M. L., and Khachiyan, L. 1996. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms* 21(3):618–628.

Garcia-Molina, H., and Barbara, D. 1985. How to assign votes in a distributed system. *Journal of the ACM (JACM)* 32(4):841–860.

Gaur, D. R., and Krishnamurti, R. 2004. Average case self-duality of monotone boolean functions. In *Canadian Conference on AI*, 322–338.

Gaur, D. R., and Krishnamurti, R. 2008. Self-duality of bounded monotone boolean functions and related problems. *Discrete Applied Mathematics* 156(10):1598–1605.

Gaur, D. R., and Makino, K. 2009. On the fractional chromatic number of monotone self-dual boolean functions. *Discrete Mathematics* 309(4):867–877.

Gottlob, G. 2013. Deciding monotone duality and identifying frequent itemsets in quadratic logspace. In *PODS*, 25–36.

Hagen, M. 2008. *" Algorithmic and Computational Complexity Issues of MONET*. Cuvillier Verlag.

Haken, A. 1985. The intractability of resolution. *Theoretical Computer Science* 39:297–308.

Ibaraki, T., and Kameda, T. 1993. A theory of coteries: Mutual exclusion in distributed systems. *Parallel and Distributed Systems, IEEE Transactions on* 4(7):779–794.

Kavvadias, D. J., and Stavropoulos, E. C. 2003. Monotone boolean dualization is in co-NP $[\log^2 n]$. *Information Processing Letters* 85(1):1–6.

Linial, N., and Tarsi, M. 1985. Deciding hypergraph 2-colourability by H-resolution. *Theoretical Computer Science* 38:343–347.

Makino, K. 2003. Efficient dualization of $O(\log n)$-term monotone disjunctive normal forms. *Discrete Applied Mathematics* 126(2-3):305–312.

Seymour, P. D. 1974. On the two-colouring of hypergraphs. *The Quarterly Journal of Mathematics* 25(1):303–311.