ON THE MAXIMAL MINIMAL CUBE LENGTHS IN DISTINCT DNF TAUTOLOGIES

MANUEL KAUERS *, MARTINA SEIDL **, AND DORON ZEILBERGER

ABSTRACT. Inspired by a recent article by Anthony Zaleski and Doron Zeilberger, we investigate the question of determining the largest k for which there exist Boolean formulas in disjunctive normal form (DNF) with n variables, which are tautologies, whose conjunctions have distinct sets of variables, and such that all the conjunctions have at least k literals. Using a SAT solver, we answer the question for some sizes which Zaleski and Zeilberger left open. We also determine the corresponding numbers for DNFs obeying certain symmetries.

1. Problem Statement

We consider Boolean formulas with n variables $x_1, \ldots, x_n$. A *literal* is a variable or a negated variable, e.g., x_3 or $\bar{x}_7$. A *cube* is a conjunction of literals, e.g., $x_3 \wedge \bar{x}_7$. The length of a cube is the number of distinct literals appearing in it. A formula in *disjunctive normal form* (DNF) is a disjunction of cubes, e.g., $(x_3 \wedge \bar{x}_7) \vee (x_5 \wedge \bar{x}_6 \wedge x_7)$. Such a DNF is called a *tautology* if it evaluates to true for all assignments of the variables. For example $x_3 \vee x_5 \vee (\bar{x}_3 \wedge \bar{x}_5)$ is a tautology. This formula consists of two cubes of length 1 and one cube of length 2.

Work of Erdös [5] on covering systems for integers has recently led Zaleski and Zeilberger [9] to consider DNFs in which all cubes have distinct supports. In a sense, these are natural Boolean analogs of the covering systems studied by Erdös. The support of a cube is the set of variables occurring in it. For example the support of the cube x_3 is the singleton set $\{x_3\}$, the support of the cube $\bar{x}_5$ is the singleton set $\{x_5\}$, while the support of the cube $\bar{x}_3 \wedge x_5$ is the set $\{x_3, x_5\}$. This implies that the DNF $x_3 \vee \bar{x}_5 \vee (\bar{x}_3 \wedge x_5)$ has distinct supports. On the other hand the Hamlet question $x_1 \vee \bar{x}_1$ does not have distinct supports.

Zaleski and Zeilberger call the formulas with distinct supports *distinct DNFs.* They want to know, for any given n, what is the largest k such that there is a distinct DNF tautology with n variables only consisting of cubes of length at least k. Using a greedy algorithm, they searched for distinct DNF tautologies with a prescribed number of variables and a prescribed minimal cube length. The largest minimal cube length for which they found formulas are as follows:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
k	0	1	1	2	3	4	4	5	6	$6_?$	7	8	9	$9_?$

These numbers are only lower bounds for the optimal values of k. However, Boole's inequality from probability theory implies that the optimal k must satisfy the inequality $\sum_{i=k}^{n} \binom{n}{i} 2^{-i} \geq 1$, which gives rise to upper bounds [9]. The numbers given in the table above turn out to match the upper bounds except for $n = 10$ and $n = 14$ (indicated by question marks), where they are off by one.

As a variant of the problem, Zaleski and Zeilberger also wanted to know, for any given n, what is the largest k such that there is a distinct DNF tautology with n variables only consisting of cubes of length exactly k. In this case, it follows from Boole's inequality that such a k must satisfy $\binom{n}{k} 2^{-k} \geq 1$, which again gives an upper bound. With their greedy approach, they determined the following lower bounds. Again, mismatches with the upper bound are indicated by a question mark.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
k	0	0	$0_?$	2	$2_?$	3	4	5	$5_?$	6	7	8	$8_?$	9

It is clear that there is no solution for $n = 3$ and $k = 1$, so in this case the upper bound is too pessimistic and $k = 0$ is the right value.

For $n = 5$ and $n = 9$ the computations reported in the present paper imply that the values 2 and 5 are also correct. We were not able to confirm the entry for $n = 10$ in the first table with about one year of

* Supported by the Austrian FWF grants F5004 and P31571-N32.
** Supported by the Austrian FWF grant S11408-N23.

computation time (more precisely: about 15 days of parallel computation time on 24 processors). We did not attempt to confirm the entries for $n = 14$ in the first or $n = 13$ in the second table.

We add two refinements to the problem. First, we introduce an additional parameter u which binds the lengths of the cubes from above. For any particular choice n, u, we want to know the largest k such that there is a distinct DNF tautology with n variables only consisting of cubes of length at least k and at most u. The special case $u = n$ corresponds to the first variant of Zaleski and Zeilberger and the special case $u = k$ corresponds to the second variant. Studying also the intermediate cases gives a broader picture of the situation.

Our second refinement concerns symmetries. Letting permutations act on the indices of the variables, we say that a DNF is *invariant* under a certain subgroup G of S_n if every $g \in G$ maps the DNF to itself. For example, the DNF $(x_1 \wedge \bar{x}_2 \wedge x_3) \vee (x_2 \wedge \bar{x}_3 \wedge x_4) \vee (x_3 \wedge \bar{x}_4 \wedge x_1) \vee (x_4 \wedge \bar{x}_1 \wedge x_3)$ is invariant under the cyclic group C_4. For the cyclic group C_n, the dihedral group D_n, the alternating group A_n, and the full symmetric group S_n (see [1] for definitions of those groups), and for various choices of n and u, we have determined the largest k such that there is a distinct DNF tautology with n variables consisting of cubes of lengths at least k and at most u which are invariant under the given group.

2. SAT Encoding

Our results were obtained with the help of a SAT solver [3, 7], using a rather straightforward encoding of the problem. For each cube, we introduced one Boolean variable that indicates whether or not this cube is going to be a part of the DNF we are looking for. Note that this creates $\sum_{i=k}^{u} \binom{n}{i} 2^i$ variables, a quantity that grows quickly when n or $u - k$ increase. For example, in the case $n = u = 10$ and $k = 7$, where we were unable to complete the computation, we were dealing with 33024 variables.

In order to enforce that the DNF is a tautology, we specify for every assignment a clause saying that at least one of the cubes that becomes true under this assignment must be selected. In order to enforce that the DNF is distinct, we have to specify clauses which encode the requirement that for every possible support, at most one of the cubes having this support can be selected. There are many ways to encode a constraint of the form “at most one”, and their pros and cons are discussed extensively in the literature [4, 6]. For our purpose, the so-called binary encoding seemed to work well.

For example, for $n = 4$ and $k = u = 2$, we have the 24 cubes $\bar{x}_1 \wedge \bar{x}_2$, $\bar{x}_1 \wedge x_2$, $\dots$, $x_3 \wedge \bar{x}_4$, $x_3 \wedge x_4$. Denote the corresponding Boolean variables by $c_1, \dots, c_{24}$, respectively. Then the clauses

$$
\begin{aligned}
&(\bar{c}_1 \vee \bar{t}_1) \wedge (\bar{c}_1 \vee \bar{t}_2) \wedge (\bar{c}_2 \vee \bar{t}_1) \wedge (\bar{c}_2 \vee t_2) \wedge (\bar{c}_3 \vee t_1) \wedge (\bar{c}_3 \vee \bar{t}_2) \wedge (\bar{c}_4 \vee t_1) \wedge (\bar{c}_4 \vee t_2) \wedge \\
&(\bar{c}_5 \vee \bar{t}_3) \wedge (\bar{c}_5 \vee \bar{t}_4) \wedge (\bar{c}_6 \vee \bar{t}_3) \wedge (\bar{c}_6 \vee t_4) \wedge (\bar{c}_7 \vee t_3) \wedge (\bar{c}_7 \vee \bar{t}_4) \wedge (\bar{c}_8 \vee t_3) \wedge (\bar{c}_8 \vee t_4) \wedge \\
&\cdots\cdots\cdots \\
&(\bar{c}_{21} \vee \bar{t}_{11}) \wedge (\bar{c}_{21} \vee \bar{t}_{12}) \wedge (\bar{c}_{22} \vee \bar{t}_{11}) \wedge (\bar{c}_{22} \vee t_{12}) \wedge (\bar{c}_{23} \vee t_{11}) \wedge (\bar{c}_{23} \vee \bar{t}_{12}) \wedge (\bar{c}_{24} \vee t_{11}) \wedge (\bar{c}_{24} \vee t_{12})
\end{aligned}
$$

with the additional auxiliary variables $t_1, \dots, t_{12}$ encode the condition that we must select at most one from $\bar{x}_1 \wedge \bar{x}_2$, $\bar{x}_1 \wedge x_2$, $x_1 \wedge \bar{x}_2$, $x_1 \wedge x_2$, at most one from $\bar{x}_1 \wedge \bar{x}_3$, $\bar{x}_1 \wedge x_3$, $x_1 \wedge \bar{x}_3$, $x_1 \wedge x_3$, and so forth. Next, the 16 clauses

$$
(c_1 \vee c_5 \vee c_9 \vee c_{13} \vee c_{17} \vee c_{21}) \wedge (c_1 \vee c_5 \vee c_{10} \vee c_{13} \vee c_{18} \vee c_{22}) \wedge \cdots\cdots \wedge (c_4 \vee c_8 \vee c_{12} \vee c_{16} \vee c_{20} \vee c_{24})
$$

enforce that for each of the 16 assignments $\sigma\colon \{x_1, \dots, x_4\} \to \{0, 1\}$ we select at least one cube it makes true, so that the selected cubes form a tautology.

In order to enforce invariance under a certain group, we chose a set of generators and added for each cube c and each generator g a clause that says “if c is selected, then also $g(c)$”. For example, the clause

$$
(\bar{c}_1 \vee c_{13})
$$

enforces that when the cube $\bar{x}_1 \wedge \bar{x}_2$ is selected, then the cube $\bar{x}_2 \wedge \bar{x}_3$ is selected as well.

The encoding as described so far is sufficient for proving existence or non-existence of a distinct DNF tautology for any prescribed n, u, k, and any prescribed group. In order to speed up the computations in practice, we may add some further constraints. One idea is to add clauses which forbid to select two cubes where one is strictly contained in the other. This is clearly a valid restriction, because when there is a solution that has two cubes that are contained in one another, we can discard the smaller one from it and obtain another solution. However, it turns out that this particular idea floods the formula with too many additional clauses and slows down the computation rather than speeding it up.

It is more efficient to break the symmetry of the problem, a standard technique in the context of SAT solving [8]. Clearly, when there is a distinct DNF for certain n, u, k and a certain group, then permuting all the variables $x_1, \dots, x_n$ in some way will yield another solution. Also replacing a certain variable x_i by its negation $\bar{x}_i$ (and canceling double negation introduced by that) turns a solution into a new solution. Since we dropped the idea to forbid cubes that are contained in other cubes, we can restrict the search to a solution containing a cube of length k, and because we are free to permute and negate variables, we may assume this cube to be $x_1 \wedge x_2 \wedge \cdots \wedge x_k$. Continuing the example above, this means that we can add the clause

$$c_4$$

consisting of the variable corresponding to this cube. Adding this clause to the formula allows for an appreciable amount of simplification (called unit propagation [3] in SAT jargon). We are left with the freedom to permute the variables $x_1, \dots, x_k$ and the variables $x_{k+1}, \dots, x_n$. By the first, it is fair to enforce an assumption that the variables are indexed in such a way that when a cube with support $x_1, \dots, x_{k-1}, x_{k+1}$ is selected, there is some i such that $x_1, \dots, x_i$ appear negated in it and the remaining variables do not. This assumption may still leave some degrees of freedom, which can be used to make a similar restriction as to which cubes with support $x_1, \dots, x_{k-1}, x_{k+2}$ may be selected. The freedom to permute the variables $x_{k+1}, \dots, x_n$ is exploited by restricting the search to DNFs such that for every $i = k+1, \dots, n-1$, the cube $x_1 \wedge \cdots \wedge x_{k-1} \wedge x_{i+1}$ is only selected when $x_1 \wedge \cdots \wedge x_{k-1} \wedge x_i$ is also selected. The corresponding clauses in the running example are

$$\bar{c}_5 \wedge \bar{c}_7 \wedge \bar{c}_9 \wedge \bar{c}_{11} \wedge (c_8 \vee \bar{c}_{12}).$$

3. Results

We have written a Python script that produces the SAT instances described in the previous section, and we have used Biere's award-winning SAT solver Treengeling [2] to solve them. Our script is available on the website of the first author. We chose Treengeling because it is currently one of the best SAT solvers that support parallel computation. We used the out-of-the-box settings of Treengeling, and we did not try other SAT solvers.

The results are summarized in the following tables, in which n appears increased towards the right and u grows downwards. Entries with $u > n$ are left blank because they are equivalent to $u = n$.

By Boole's inequality, the maximal k for a particular choice of n and u must satisfy the inequality $\sum_{i=k}^{u} \binom{n}{i} 2^{-i} \geq 1$. An entry in the table is boxed when it does not match this bound. For the entries marked with a question mark, we have not been able to prove that the k we found is really optimal, but the long and unsuccessful search is at least some indication that the bound is not reached in these cases. For $(n, u) \in \{(5,3), (9,6), (10,8)\}$, the SAT solver is able to show that distinct DNF tautologies with $k = 3$, $k = 6$, $k = 7$, respectively, do not exist, although their existence would not be in conflict with the bound.

	2	3	4	5	6	7	8	9	10
2	1	1	2	2	2	2	2	2	2
3		1	2	$\boxed{2}$	3	3	3	3	3
4			2	3	3	4	4	4	4
5				3	4	4	5	5	5
6					4	4	5	$\boxed{5}$	6
7						4	5	6	6
8							5	6	$\boxed{6}$
9								6	$\boxed{6}$?
10									$\boxed{6}$?

The next two tables contain our results about distinct DNF tautologies invariant under certain groups. We have investigated the cyclic group C_n, the dihedral group D_n, the alternating group A_n, and the full symmetric group S_n. The table on the left lists the numbers for C_n and D_n, which turn out to be identical, apparently because D_n is only slightly larger than C_n. Boxed entries highlight the differences to the previous table. The question marks refer to the search for C_n, which for three entries did not terminate in a reasonable amount of time. Interestingly, it follows from the previous table that the entry for $(n, u) = (10, 8)$ is correct, but while the SAT solver was able to prove this in the (seemingly harder) case without invariant constraints, it did not succeed with the constraints for C_n. The computations for all entries terminated in presence of the constraints for D_n.

	2	3	4	5	6	7	8	9	10
2	1	1	1	1	1	1	1	1	1
3		1	2	2	2	3	3	3	3
4			2	$\boxed{2}$	3	$\boxed{3}$	4	4	4
5				$\boxed{2}$	$\boxed{3}$	4	$\boxed{4}$	5	5
6					$\boxed{3}$	4	5	5	6
7						4	5	6	6
8							5	6	$6_?$
9								6	$6_?$
10									$6_?$

	2	3	4	5	6	7	8	9	10	11	12	13	14
2	1	1	1	1	1	1	1	1	1	1	1	1	1
3		1	2	2	2	2	2	2	2	2	2	2	2
4			2	2	3	3	3	3	3	3	3	3	3
5				2	3	3	4	4	4	4	4	4	4
6					3	3	4	4	5	5	5	5	5
7						3	4	4	5	5	6	6	6
8							4	4	5	5	6	6	7
9								4	5	5	6	6	7
10									5	5	6	6	7
11										5	6	6	7
12											6	6	7
13												6	7
14													7

The table on the right lists the numbers for A_n and S_n, which also turn out to be the same, apparently because A_n is only slightly smaller than S_n. For these groups, the invariant constraints make the problem easier, so that we were able to cover slightly larger values of n and u. All given numbers have been proved to be optimal. Note that a regular pattern emerges: we seem to have the formula $k = \min(u-1, \lfloor n/2 \rfloor)$.

References

[1] Michael Artin. *Algebra.* Prentice Hall, 1991.

[2] Armin Biere. CaDiCaL, Lingeling, Plingeling, Treengeling, YalSAT Entering the SAT Competition 2017. *Proc. Proceedings of SAT Competition 2017 - Solver and Benchmark Descriptions,* 2017.

[3] Armin Biere, Marijn Heule, and Hans van Maaren, eds. *Handbook of Satisfiability.* Vol. 185. IOS press, 2009.

[4] Jiangchao Chen. A new SAT encoding of the at-most-one constraint. *Proc. Constraint Modelling and Reformulation,* Toni Mancini, Justin K. Pearson (eds.), 2010.

[5] Paul Erdös. On integers of the form $2^k + p$ and some related problems. *Summa Brasil. Math.,* 1950.

[6] Alan M. Frisch and Paul A. Giannaros. SAT encodings of the at-most-k constraint: some old, some new, some fast, some slow. *Proc. of the Tenth Int. Workshop of Constraint Modelling and Reformulation,* 2010.

[7] Donald E. Knuth. Satisfiability. *The Art of Computer Programming,* Volume 4, Fascicle 6, Addison-Wesley, 2015.

[8] Karem A. Sakallah. Symmetry and Satisfiability. *Handbook of Satisfiability,* Armin Biere, Marijn Heule, Hans van Maaren (eds.), 2009.

[9] Anthony Zaleski and Doron Zeilberger. Boolean Function Analogs of Covering Systems. To appear in Mathematics Magazine. Also `https://arxiv.org/abs/1801.05097` (Combinatorics), 2018.

Manuel Kauers, Institute for Algebra, J. Kepler University Linz, Austria

E-mail address: `manuel.kauers@jku.at`

Martina Seidl, Institute for Formal Models and Verification, J. Kepler University Linz, Austria

E-mail address: `martina.seidl@jku.at`

Doron Zeilberger, Department of Mathematics, Rutgers University, USA

E-mail address: `DoronZeil@gmail.com`