Electronic Colloquium on Computational Complexity, Report No. 85 (2022)

A Note on Lower Bounds for Monotone Multilinear Boolean Circuits

Andrzej Lingas[1]

Department of Computer Science, Lund University
Andrzej.Lingas@cs.lth.se

Abstract. A monotone Boolean circuit is a restriction of a Boolean circuit allowing for the use of disjunctions, conjunctions, the Boolean constants, and the input variables. A monotone Boolean circuit is multilinear if for any AND gate the two input functions have no variable in common. We show that the known lower bounds on the size of monotone arithmetic circuits for multivariate polynomials that are sums of monomials consisting of k distinct variables yield the analogous lower bounds divided by $O(k^2)$ on the size of monotone multilinear Boolean circuits computing the Boolean functions represented by the corresponding multivariate Boolean polynomials.

Keywords: monotone Boolean circuit, monotone multilinear Boolean circuit, monotone arithmetic circuit, circuit size

1 Introduction

The derivation of superlinear lower bounds on the size of Boolean circuits for natural problems appeared extremely hard. Therefore, already at the end of of the 70s and the beginning of the 80s, several researches started to study the complexity of monotone arithmetic or monotone Boolean circuits for natural multivariate arithmetic polynomials and natural Boolean functions, respectively. The monotone arithmetic circuits are composed of addition gates, multiplication gates and input gates for variables and non-negative real constants. Similarly, monotone Boolean circuits are composed of OR gates, AND gates, and the input gates for variables and Boolean constants. In the case of monotone arithmetic circuits, one succeeded to derive even exponential lower bounds relatively easily [2, 9] while in the case of monotone Boolean circuits the derivation of exponential lower bounds for natural problems required much more effort [1, 7].

The problem of computing the permanent of an $n \times n$ matrix equivalent to counting the number of perfect matchings in a bipartite graph is an example of a problem for which the gap between lower bounds in the models of monotone arithmetic circuits and monotone Boolean circuits remains very large up to today. Namely, Jerrum and Snir established an exponential lower bound on the size of monotone arithmetic circuits for this problem [2] while the best known lower bound on the size of a monotone Boolean circuit computing the Boolean

ISSN 1433-8092

variant of the permanent shown by Razborov [8] is only superpolynomial. In order to tackle the gap, Ponnuswami and Venkateswaran introduced the concept of monotone multilinear Boolean circuits and showed an exponential lower bond on the size of the restricted monotone Boolean circuits for the Boolean permanent [4]. A Boolean circuit is multilinear if for any AND gate the two input functions have no variable in common. On the other hand, Raz and Widgerson showed that monotone Boolean circuits for the Boolean permanent require linear depth [6] and Raz proved that multilinear Boolean formulas for this problem have superpolynomial size [5].

In this note, we consider monotone Boolean circuits. They are a restriction of Boolean circuits allowing for disjunctions, conjunctions, the Boolean constants, and the input variables.

We use a simple argument to obtain a more general result than the lower bound of Ponnuswami and Venkateswaran in [4]. We show that the known lower bounds on the size of monotone arithmetic circuits for multivariate polynomials that are sums of monomials consisting of k distinct variables [2, 9] yield the analogous lower bounds divided by $O(k^2)$ on the size of monotone multilinear Boolean circuits computing the Boolean functions represented by the corresponding multivariate Boolean polynomials.

Since several of the aforementioned lower bonds are exponential, our results give further evidence that the model of monotone multilinear Boolean circuit is substantially more restricted than that of monotone Boolean circuit. In particular, the multilinearity condition makes impossible to use the full power of idempotency.

2 Monotone circuits, Boolean functions, and arithmetic polynomials

A *monotone Boolean circuit* is a finite directed acyclic graph with the following properties:

1. The indegree of each vertex (termed gate) is either 0 or 2.
2. The source vertices (i.e., vertices with indegree 0 called input gates) are labeled by variables or the Boolean constants 0, 1.
3. The vertices of indegree 2 are labeled by elements of the set $\{OR, AND\}$ and termed OR gates and AND gates, respectively.

A *monotone arithmetic circuit* is defined analogously by replacing OR and AND with addition and multiplication, respectively. As for constants one could allow for positive ones as in [9] or not use them at all as in [2]. For technical reasons, we shall just allow for the 0, 1 constants. Of course, they can be easily eliminated without increasing the number of additions or multiplications.

For convenience, we shall denote the function represented by the multivariate Boolean or arithmetic polynomial computed at a gate g of a monotone Boolean or arithmetic circuit also by g. The *size* of a monotone Boolean or an arithmetic circuit is the total number of its non-input gates.

A monotone Boolean circuit is *multilinear* if for any AND gate the two input Boolean functions have no variable in common.

With each gate g of a monotone Boolean circuit, we shall associate a set $T(g)$ of terms in a natural way. Thus, with each input gate, we associate the singleton set consisting of the corresponding variable or constant. Next, with an OR gate, we associate the union of the sets associated with its direct predecessors. Finally, with an AND gate g, we associate the set of concatenations t_1t_2 of all pairs of terms t_1, t_2, where $t_i \in T(g_i)$ and g_i stands for the i-th direct predecessor of g for $i = 1, 2$. The function computed at the gate g is the disjunction of the functions (called monoms) represented by the terms in $T(g)$. The monom $con(t)$ represented by a term t is obtained by replacing concatenations in t with conjunctions, respectively. A term in $T(g)$ is a *zero-term* if it contains the Boolean constant 0. Clearly, a zero-term represents the Boolean constant 0. By the definition of $T(g)$ and induction on the structure of the monotone Boolean circuit, $g = \bigvee_{t \in T(g)} con(t)$ holds. For a term $t \in T(g)$, the set of variables occurring in t is denoted by $Var(t)$.

A Boolean form is a finite set of Boolean 0-1 functions. An *implicant* of a Boolean form F is a conjunction of some variables and/or Boolean constants (monom) such that there is a function belonging to F which is true whenever the conjunction is true. If the conjunction includes the Boolean 0 then it is a *trivial implicant* of F.

A non-trivial implicant of F that is minimal with respect to included variables is a *prime implicant* of F. The set of prime implicants of F is denoted by $PI(F)$.

An arithmetic multivariate monomial is a product of a finite number of variables. An arithmetic multivariate polynomial is a linear combination of a finite number of (arithmetic multivariate) monomials possibly extended by a free constant coefficient. The set of monomials of the polynomial P is denoted by $Mon(P)$. If all coefficients at the monomials in $Mon(P)$ and the free constant coefficient (if any) are positive then the polynomial P is *monotone*. We shall say that two monotone arithmetic multivariate polynomials P and Q are *similar* if $Mon(P) = Mon(Q)$.

3 Lower bounds for monotone multilinear Boolean circuits

Our main idea is to transform a monotone multilinear Boolean circuit C computing a monotone Boolean function P, whose prime implicants consist of k distinct variables, into an $O(k^2)$ times larger monotone multilinear Boolean circuit C' that in particular includes a gate whose terms coincide with the non-zero terns of the output gate of C that contain k variable occurrences. By the multilinearity of C', the latter terms contain exactly k distinct variables and represent exactly all prime implicants of P. After replacing disjunctions and conjunctions by additions and multiplications in C', respectively, we obtain a monotone arithmetic circuit computing a multivariate polynomial having the same set of monomials as the arithmetic multivariate polynomial corresponding to the representation

of P as a disjunction of its prime mplicants. The following lemma is helpful in implementing this idea.

Lemma 1. *Let g be a gate of a monotone Boolean circuit without constant input gates for a monotone Boolean function f. For each prime implicant p of g there is a term $t \in T(g)$ representing p that does not contain two or more occurrences of the same variable. Consequently, if all prime implicants of g have the same length k then $T(g)$ restricted to terms of length k consists of (non-zero) terms having k distinct variable occurrences and representing solely all prime implicants of g.*

Proof. The first part is proved by an induction on the structure of the circuit in a bottom up manner. A term with two or more occurrences of the same variable can be formed only at an AND gate. Let $g_1,\ g_2$ be direct gate predecessors of g. Suppose first that g is an AND gate. Consider $p \in PI(g)$. It follows that there are $p_i \in PI(g_i)$, $i = 1, 2$, such that $P = p_1 \wedge p_2$. By the induction hypothesis, there are terms $t_i \in PI(g_i)$ reprsenting p_i, $i = 1, 2$, respectively, and having only distinct variable occurrences. We have $Var(t_1) \cap Var(t_2) = \emptyset$ by the multilinearity assumption since $p_1,\ p_2$ are prime implicants of $g_1,\ g_2$, respectively. Therefore, the term $t_1 t_2$ has only distinct variable occurrences. The case when is g is an OR gate follows immediately by the induction hypothesis since if $p \in PI(g)$ then $p \in PI(g_1) \cup PI(g_2)$. This completes the proof of the first part. The second part follows immediately from the fact that each term in $T(g)$ represents an implicant of g and if it does not represent a prime implicant of g then the term has to contain at least $k + 1$ variable occurrences. □

Theorem 1. *Let f_n be a sequence of monotone Boolean functions from $\{0,1\}^n$ to $\{0,1\}$ such that each prime implicant of f_n consists of $k(n)$ distinct variables. Let C be a monotone multilinear Boolean circuit for f_n with $\beta(n)$ OR gates and $\alpha(n)$ AND gates. There is a multivariate polynomial $\sum_{p \in PI(f_n)} a_p mon(p)$, where a_p are natural coefficients and $mon(p)$ is the monomial resulting from the replacement of conjunctions with multiplications in p, and a monotone arithmetic circuit with $O(k(n)^2\alpha(n) + k(n)\beta(n))$ addition gates and $O(k(n)^2\alpha(n))$ multiplication gates computing the polynomial.*

Proof. We may assume w.l.o.g. that the circuit C does not use input gates with the Boolean constants 0, 1, since otherwise, we can easily eliminate them preserving the properties of C, in particular the multilinearity, without increasing the number of OR and AND gates.

Now, the idea is to construct a monotone multilinear Boolean circuit C' on the basis of C, where for each gate g of C there are at most $k(n)$ corresponding gates $g_1, .., g_{k(n)}$. For $1 \leq \ell \leq k(n)$, the set of terms in $T(g_\ell)$ is supposed to consist of the (non-zero) terms in $T(g)$ having exactly ℓ variable occurrences. The construction of C' is straightforward. For an input gate g corresponding to a variable x, only $g_1 = x$ is defined. For a gate g of C with direct predecessor gates $g',\ g''$ and $1 \leq \ell \leq k(n)$, if g is an OR gate then $g_\ell = g'_\ell \vee g''_\ell$ provided that both g'_ℓ and g''_ℓ are defined. If only one of the latter gates is defined then it is substituted for g_ℓ. Finally, if none of the gates $g'_\ell,\ g''_\ell$ is defined then g_ℓ is not defined.

Thus, there are at most $k(n)\beta(n)$ OR gates in C' corresponding to an OR gate in C. If g is an AND gate then $g_\ell = \bigvee_{j=1}^{\ell-1} g'_j \wedge g''_{\ell-j}$, where the conjunction $g'_j \wedge g''_{\ell-j}$ takes place if both g'_j and $g''_{\ell-j}$ are defined for $j = 1, ..., \ell - 1$. If no conjunction takes place in the sum g_ℓ is not defined. Thus, in the case of the AND gate, a partial convolution of $(g'_1, ..., g'_{k(n)})$ and $(g''_1, ..., g''_{k(n)})$ needs to be computed. It requires $O(k(n)^2)$ AND and OR gates. Importantly, the AND gates in the monotone Boolean subcircuit computing the convolution satisfy the multilinearity condition since the original AND gate g does it. Thus, the resulting monotone Boolean circuit is multilinear and it has $O(k(n)^2\alpha(n)+k(n)\beta(n)))$ OR gates and $O(k(n)^2\alpha(n))$ AND gates. The correctness of the construction of C' , i.e., the fulfillment of the supposed relationship between gates in C and the corresponding gates in C' follows by an induction on the structure of C in a bottom-up manner By the second part of Lemma 1, $g = \bigvee_{p \in PI(g)} p = \bigvee_{t \in T(g_{k(n)})} con(t) = g_{k(n)}$ holds and each term in $T(g_{k(n)}$ represents a prime implicant in $PI(g)$ and has $k(n)$ different variable occurrences. Hence, if we transform C' to a monotone arithmetic circuit by replacing OR gates by addition gates and AND gates by multiplication gates, the multivariate polynomial computed at the gate in the transformed circuit corresponding to $g_{k(n)}$ will have the form stated in the theorem. □

Corollary 1. *Let P_n be a sequence of monotone arithmetic n-variable polynomials whose each monomial is a product of $k(n)$ distinct variables. Suppose that any monotone arithmetic circuit computing P_n or a polynomial similar to P_n requires $a(n)$ additions and $m(n)$ multiplications. Let P'_n be the corresponding monotone Boolean polynomial resulting from the replacement of additions and multiplications by disjunctions and conjunctions, respectively, and discarding the positive coefficients at the monomials. Any monotone multilinear Boolean circuit computing the Boolean function represented by P'_n has $\Omega((a(n) + m(n))/k(n)^2)$ size. In particular, it requires $\Omega(m(n)/k(n)^2)$ AND gates.*

Proof. Observe that the monoms of P'_n corresponding to the monomials of P_n form the set of prime implicants of the Boolean function represented by P'_n. Clearly, the monotone arithmetic multivariate polynomial resulting from the replacement of disjunctions and conjunctions by additions and multiplications, respectively, in P'_n is similar to P_n. Hence, if for any constant c, a monotone multilinear Boolean circuit computing the Boolean function represented by P'_n has size smaller than the number of addition and multiplication gates divided by $ck(n)^2$ in a monotone arithmetic circuit computing a polynomial similar to P_n then we obtain a contradiction with Theorem 1. Similarly, if for any constant c, a monotone multilinear Boolean circuit computing the Boolean function represented by P'_n has a number of AND gates smaller than the number of multiplication gates divided by $ck(n)^2$ in a monotone arithmetic circuit computing a polynomial similar to P_n then again we obtain a contradiction with Theorem 1. □

Since the proofs of the lower bounds on the size of monotone arithmetic circuits for the monotone multivariate polynomials in [2, 9] work also for the

polynomials similar to them, i.e., they are invariant of the positive coefficients at the monomials of the polynomials, we obtain the following corollary.

Corollary 2. *The lower bounds on the number of addition gates and/or multiplication gates in monotone arithmetic circuits for several sequences of monotone arithmetic n-variable polynomials whose monomials consist of $k(n)$ distinct variables established by Schnorr in [9] and Jerrum and Snir in [2] carry over to analogous lower bounds on the number of OR and AND gates or just AND gates, respectively, divided by $O(k(n)^2)$, in monotone multilinear Boolean circuits computing the Boolean functions represented by the corresponding monotone Boolean polynomials (resulting from the replacement of additions and multiplications by disjunctions and conjunctions, respectively).*

Proof. Schnorr introduced the concept of a *separated* set of monomials of a monotone arithmetic multivariate polynomial in [9]. Such a set S is called separated if it has the following property: for any three monomials r, s, t in S, if the multi-set of variables in r is contained in the union of the multi-sets of variables in s and t then $r = s$ or $r = t$. Schnorr showed that any monotone arithmetic circuit computing a polynomial P whose set S of monomials is separated has to include at least $|S| - 1$ addition gates. Since the definition of a separated set of monomials is invariant of the positive coefficients at the monomials of P, the same lower bound on the number of additions holds for any polynomial similar to P.

Similarly as in [9], the lower bounds bounds in [2] are established for homogeneous and linear polynomials, i.e., all monomials of the polynomial have the same degree and no monomial contains two or more occurrences of the same variable. Again, the lower bounds on the monotone arithmetic complexity of specific polynomials established in Section 4 of [2] are invariant of the values of the positive coefficients at the monomials and can be immediately extended to any similar polynomials. Namely, the underlying lower bounds are established in Section 3 of [2] for *formal* homogeneous and linear polynomials over the Boolean semi-ring. Then, they are carried over in particular to the arithmetic semi-ring on non-negative reals by a homomorphism τ mapping any positive real onto the Boolean 1 and 0 onto the Boolean 0, respectively (see page 878 in [2]). □

The lower bounds on the size of monotone arithmetic circuits presented in [2, 9] (cf. [10]) include the clique polynomial $CL_{n,k}$, the permanent polynomial $Per_{n\times n}$, the Hamiltonian circuit polynomial $HC_{n\times n}$, and the spanning tree polynomial $ST_{n,n}$. Their Boolean counterparts are

$$BCL_{n,k} = \bigvee_{1\le v_1<\ldots<v_k\le n} \bigwedge_{1\le i<j\le k} x_{v_i,v_j},$$

$$BPer_{n\times n} = \bigvee_{\pi\in S(n)} \bigwedge_{i=1}^{n} x_{i,\pi(i)},$$

$$BHC_{n\times n} = \bigvee_{\pi\in C(n)} \bigwedge_{i=1}^{n} x_{i,\pi(i)},,$$

and

$$BST_{n,n} = \bigvee_{t \in T(n)} x_{2,t(2)} x_{3,t(3)} ... x_{n,t(n)},$$

respectively, where $S(n)$, $C(n)$ stand for the set of permutations and the set of cyclic permutations of $1, ..., n$, respectively, and $T(n)$ is the set of functions $t : \{2, 3, ..., n\} \to \{1, 2, ..., n\}$ such that for each $i \in \{2, ..., n\}$ there is q satisfying $t^q(i) = 1$.

Ponnuswami and Venkateswaran showed that any monotone multilinear Boolean circuit for the Boolean function represented by $BPer_{n \times n}$ has an exponential in n size [4]. By combining Corollary 2 with [2, 9], we obtain the following lower bounds on the size of monotone multilinear Boolean circuits for the Boolean functions represented by $BCL_{n,k}$, $BPer_{n \times n}$, $BHC_{n,n}$, and $BST_{n,n}$, respectively. For simplicity, we use the names of the Boolean polynomials to denote also the functions represented by them in the following corollary.

Corollary 3. *Any monotone multilinear Boolean circuit for $BCL_{n,k}$ has $\Omega(\binom{n}{k}/k^2)$ size. Next, any monotone multilinear circuit for $BPer_{n,n}$ includes $\Omega(n^{-1}(2^{n-1} - 1))$ AND gates, any monotone multilinear circuit for $BHC_{n,n}$ includes $\Omega(2^{n-3})$ AND gates, and any monotone multilinear circuit for $BST_{n,n}$ includes $\Omega(n^{-3}(4/3)^{n-1})$ AND gates.*

They lower bounds in [2, 9] also include monotone arithmetic circuits for such sets of polynomials as matrix product, convolution and their iterated versions. All results in this section can be easily generalized to include monotone arithmetic circuits computing sets of polynomials and multilinear Boolean circuits computing the corresponding Boolean forms, respectively.

4 Final remarks

The multilinearity restriction in monotone Boolean circuits is substantial. In particular, it makes impossible to use the full power of idempotency.

As the lower bounds in [2, 9] we rely on are basically asymptotically tight it follows easily that our concrete lower bounds in Corollary 3 are asymptotically tight up to the $O(k^2)$ factor, where k is the length of the prime implicants of the respective Boolean functions.

Most likely, the requirement in Corollary 1 that the lower bounds hold also for similar polynomials is unnecessary at least in case of the polynomials with symmetric monomials as those considered in [2, 9]. In case of the symmetry of the monomials it is hard to imagine how an asymmetry of the coefficients at the monomials could speed up the circuit computation.

Acknowledgments

Thanks go to Susanna de Rezende for bringing attention to the monotone Boolean circuit complexity of the Boolean permanent problem studied in [4, 8] and valuable discussions. The research was supported by Swedish Research Council grant 621-2017-03750.

References

1. N. Alon and R. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.
2. M. Jerrum and M. Snir. Some Exact Complexity Results for Straight-Line Computations over Semirings. *J. ACM*, vol. 29(3), pp. 874–897, 1982.
3. S. Jukna and A. Linga. Lower Bounds for DeMorgan Circuits of Bounded Negation Width. *Proc. of Symposium on Theoretical Aspects of Computer Science (STACS)* 2019, pp. 41:1-41:17.
4. A. Kumar Ponnuswami and H. Venkateswaran. Monotone Multilinear Boolean Circuits for Bipartite Perfect Matching RequireExponential Size. *Proc. of 24th International Conference on Foundations of Software Technology and Theoretical Computer Science (FST-TCS)*, pp. 16-18, 2004.
5. R. Raz Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size. *Electron. Colloquium Comput. Complex.(067)* (2003)
6. R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992.
7. A. A. Razborov. Lower bounds for the monotone complexity of some boolean functions. *Soviet Math. Dokl.*, 31:354–357, 1985.
8. A. A. Razborov. Lower bounds on monotone complexity of the logical permanent. *Math. Notes of the Acad. of Sci. of the USSR*, 37(6):485–493, 1985.
9. C.P. Schnorr. A Lower Bound on the Number of Additions in Monotone Computations. *Theoretical Compututer Science* 2(3), pp. 305–315, 1976.
10. E. Shamir and M. Snir. Lower bounds on the number of multiphcatlons and the number of additions in monotone computations. *Tech Rep RC* 6757, IBM Thomas J. Watson Research Center, Yorktown Heights, N Y , 1977.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il