

Monotone Multilinear Boolean Circuits for Bipartite Perfect Matching Require Exponential Size ^{*}

Ashok Kumar Ponnuswami and H. Venkateswaran

College of Computing,
Georgia Institute of Technology,
Atlanta GA 30309, USA
E-mail: {pashok, venkat}@cc.gatech.edu

Abstract. A monotone boolean circuit is said to be *multilinear* if for any *AND* gate in the circuit, the minimal representation of the two input functions to the gate do not have any variable in common. We show that monotone multilinear boolean circuits for bipartite perfect matching require exponential size. In fact we prove a stronger result by characterizing the structure of the smallest monotone multilinear boolean circuits for the problem. We also show that the upper bound on the minimum depth of monotone circuits for perfect matching in general graphs is $O(n)$.

1 Introduction

Since Razborov [7] showed a super-polynomial lower bound on the size of monotone circuits for perfect matching and established a super-polynomial gap between the power of general circuits and monotone circuits, there have been other functions in P [9] for which the gap has been shown to be exponential. But the lower bound for monotone circuits for perfect matching is still super-polynomial. It has been shown in [6] that monotone circuits for perfect matching require linear depth.

Let PM denote the problem of finding whether a graph has a perfect matching and let BPM denote the problem of finding whether a bipartite graph has a perfect matching. The upper bound on size of arithmetic circuits for permanent in [1] yields a $2^{O(n)}$ size monotone boolean circuit for BPM directly (replace the product and plus gates with *AND* and *OR* gates respectively) and can be generalized to an upper bound on size for PM also. But the depth of these circuits is $\Omega(n \log n)$. We show in Section 2 that linear depth monotone circuits can be constructed for PM .

Since attempts to show an exponential lower bound on the size of monotone circuits for BPM have not succeeded, it seems worthwhile to check if such a bound can be shown for restricted monotone circuits. In Section 3 we show

^{*} Revised version

that under two different restrictions on the function calculated by *AND* gates and *OR* gates, monotone circuits for *BPM* require exponential size. We call the circuits satisfying the restriction on the *AND* gates *multilinear* because of the similarity to the restriction of multilinearity in arithmetic circuits. As defined in [5], an arithmetic circuit is multilinear if at each gate the power of any variable in minimal representation of the polynomial computed is at most 1. Equivalently, for any product gate, the minimal representation of the polynomials of its two input gates have no variable in common. A recent result of Raz [5] shows a super-polynomial lower bound on multilinear arithmetic formulas for the permanent. Multilinearity for arithmetic circuits has been extensively studied for the lack of strong lower bounds for general arithmetic circuits and because they seem to be the most intuitive circuits for multilinear functions (see [4] and [5] for more references). We call a monotone boolean circuit *multilinear* if for any *AND* gate, the minimal representation of the of the two inputs to the gate do not have any variable in common. We say a monotone boolean circuit is in the *simplest form* if it satisfies the restriction on *OR* gates and a stronger restriction on *AND* gates than multilinearity. The circuits constructed in Section 2 and the circuits obtained from [1] for perfect matching are examples of circuits in the simplest form. It turns out that the smallest multilinear boolean circuits for *BPM* are also in the simplest form. The upper bound on size implied by Section 2 and the lower bound shown in Section 3 are very close showing that the analysis of the lower bound for monotone multilinear boolean circuits is quite tight. Since the circuits in the simplest form not only seem to be the most natural circuits for *PM* and *BPM*, but also attain the lower bound for multilinear boolean circuits and provide the best known upper bound on the size and depth of monotone circuits for these problems, it seems plausible to conjecture that these are the smallest monotone circuits for the perfect matching problem.

To the best of our knowledge, our lower bounds are not implied by any of the known lower bounds for arithmetic and boolean circuits. When a multilinear boolean circuit for *BPM* is converted to an arithmetic circuit in the natural way (by replacing the *AND* and *OR* gates by product and plus gates), it does not necessarily yield a multilinear arithmetic circuit for permanent because boolean circuits can use idempotence.

We use the notion of *minterm* defined in [2] to analyze the circuits. Our lower bounds use a direct argument for circuits computing *BPM*. One of the key lemmas (based on the idea that there are not “many” perfect matchings that do not have an edge crossing a “balanced” cut) is reminiscent of the tree separator theorem [3]. The tree separator theorem has had many applications such as small depth circuits for context-free language recognition [8] and showing the relationship between depth and size of monotone circuits (see [10]).

In what follows, we will assume that all circuits are monotone boolean circuits in which the *AND* and *OR* gates have fan-in 2. The inputs to the circuit correspond to edges of a graph G on a set $V = \{1, 2, \dots, n\}$ of n vertices, where n is even. That is, the input corresponding to pair $\{i, j\}$ is 1 if the edge is present in G and 0 otherwise.

2 Upper Bounds on Depth and Size of Monotone Boolean Circuits for PM

Let $S \subseteq V$ and $|S|$ be even. A subset m of edges is said to be an S -matching if m is a matching with an edge incident on each vertex of S and no edge in m has one end point in S and the other in $V - S$ (m may contain edges that have both end-points in $V - S$). We say m is an *exact S -matching* if m is an S -matching and m has no edge incident on a vertex in $V - S$.

We first describe the depth upper bound for PM .

Theorem 1. *PM has monotone circuits of $O(n)$ depth.*

Proof. For the sake of simplicity of this construction, we assume $n = 2^k$, though the result can be proven for any even n . If $n = 2$, the construction is trivial. So assume $n > 2$. Suppose for each $S \subseteq V, |S| = n/2$, we are given a circuit C_S that evaluates to 1 iff there is a S -matching in the input graph G . If we take the *AND* of C_S and C_{S^c} , we get a circuit C_P that evaluates to 1 iff there is a perfect matching that does not cross the partition $P = \{S, S^c\}$. If we take the *OR* of all circuits corresponding to the partitions of V into two sets of $n/2$ vertices, we get a circuit C_V for perfect matching on V (see Figure 1). Since the number of partitions of V into $n/2$ vertices is $\frac{n!}{2!(n/2)!^2}$, the depth of the *OR* gates at the output of C_V is $\lfloor \log n! - \log 2! - 2 \log(n/2)! \rfloor \leq an + b$ where a and b are positive constants independent of n . Therefore the depth of C_V is $an + b + 1 + \max_S \text{depth}(C_S)$ where $\text{depth}(C_S)$ is the depth of C_S .

The C_S may be recursively constructed the same way as C_V since C_S is a circuit for perfect matching on the graph induced by S . Therefore the depth of $C \leq (an + b + 1) + (an/2 + b + 1) + \dots = O(n)$. If n were not a power of 2, at each level of recursion, split the set of vertices into two sets of even sizes in the most balanced way. For example if $n = 36$, we need to consider all subsets of size 18 at the first level, all subsets of size 8 and 10 at the second level, all subsets of size 4 and 6 at the third level and so on. \square

The upper bound on size for PM obtained above is $n^{O(1)}2^{3n/2}$ as the level of recursion that contains the maximum number of gates is the second level with $O\left(\binom{n}{n/2} \binom{n/2}{n/4}\right)$ gates. Since finding if there is a perfect matching in a bipartite graph on n vertices is the same as checking whether the permanent of the $n/2 \times n/2$ incidence matrix is not zero, we can obtain a upper bound of $n^{O(1)}2^{n/2}$ on size for BPM by replacing the product and sum gates in the arithmetic circuit for permanent given in [1]. The same approach can be generalized to obtain a better size upper bound for PM as follows. In the construction in Theorem 1, instead of considering partitions $\{S, S^c\}$ into the most balanced even sets, consider partitions such that S is a set of two vertices, one of which is the vertex of smallest index in V . Now the circuit for each S^c can be constructed recursively. At level 2, we need at most $\binom{n-1}{n-2}$ subcircuits to compute the different possible values for S^c . In general, at level i , we need at most $\binom{n-i}{n-2i}$ subcircuits. Solving for the value for i that maximizes $\binom{n-i}{n-2i}$, we get that the circuit constructed has size $O(2^{0.695n})$.

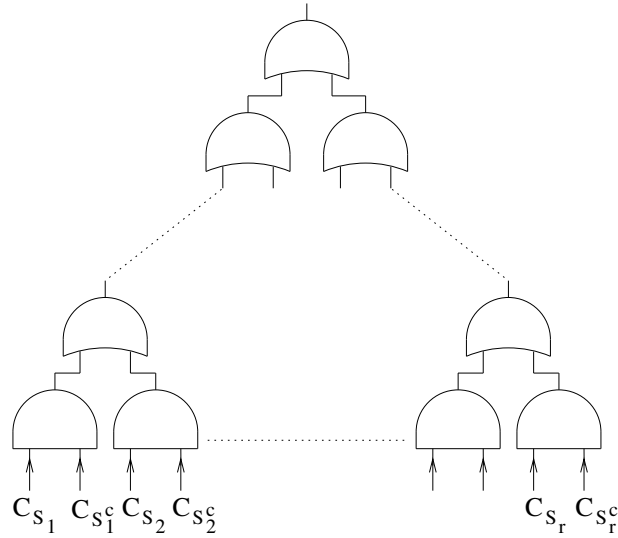


Fig. 1. Construction of circuit C_V for showing the depth upper bound

3 Lower Bound on Size of Restricted Monotone Circuits for BPM

A *minterm* of a monotone boolean function is a minimal set of variables that when set to 1, the function evaluates to 1 irrespective of the value of the variables. Let C be a circuit. Let g be a gate of C . The function computed by g is the boolean function representing the output of g in terms of the input gates. The set of minterms of g in circuit C , denoted by $\text{minterm}_C(g)$ (or $\text{minterm}(g)$ if the circuit is clear from the context) is the set of minterms of the function computed by g .

Throughout this section, we assume that the inputs to a circuit correspond to the edges of a bipartite graph G on a bipartition $\{V', V'^c\}$ of $V = \{1, 2, \dots, n\}$ into equal sets. That is, there is an input corresponding to each pair $(i, j) \in V' \times V'^c$ that is 1 if the edge between i and j is present in G and 0 otherwise. The *edge set* of gate g is the set of all edges that appear in some minterm of g . The *vertex set* of a subset m of the edges is the set of all end points of the edges in m . The *vertex set* of g is the vertex set of its edge set. The edge set and vertex set of a subcircuit of C are defined to be the respective values for the output gate of the subcircuit.

3.1 Lower Bound for Simple Circuits for BPM

A circuit is said to be *simple* if for any *OR* gate g in the circuit with input gates g_1 and g_2 , the vertex sets of g , g_1 and g_2 are the same. It can be seen by induction that in a simple circuit, the vertex set of any minterm of a gate is the

same as the vertex set of the gate. The statement is true for input gates. If it is true for the input gates g_1 and g_2 of an *OR* gate g , it is true for g as well since minterms of g are either minterms of g_1 or g_2 . If the statement is true for the input gates g_1 and g_2 of an *AND* gate g , all the minterms of g have the same vertex set since any minterm of g is the union of some minterm of g_1 with some minterm of g_2 and all minterms of g_1 have the same vertex set, as do minterms of g_2 .

Lemma 1. *Assume $n > 2$. Let C be a simple monotone circuit for BPM. For any perfect matching m , $\exists V_m \subseteq V$, $|V|/3 \leq |V_m| \leq 2|V|/3$ such that V_m is the vertex set of some gate g and m is a V_m -matching.*

Proof. Set $U = V$, $m' = m$ and let g be the output gate of C . At any stage later, we will ensure that (U, m', g) satisfy the following constraints:

- (1) U is the vertex set of g .
- (2) m' is a subset of m , m' is a minterm of g and U is the vertex set of m' (and hence m is a U -matching).

Also, let $U > 2|V|/3$, which is true initially when $U = V$. Since $|V| = n > 2$, the gate g can not be an input gate. If g is an *OR* gate, one of the two input gates g_1 or g_2 of g , say g_1 , has m' as a minterm. Let $g \leftarrow g_1$ and repeat. If g is an *AND* gate, let V_1 and V_2 be the vertex sets of its input gates g_1 and g_2 respectively. Without loss of generality, let $|V_1| \geq |V_2|$. There must be some minterm m_1 of g_1 and m_2 of g_2 such that $m' = m_1 \cup m_2$. From a remark above the lemma, we know that the vertex sets of m_1 and m_2 are V_1 and V_2 respectively, and hence $U = V_1 \cup V_2$. Therefore (V_1, m_1, g_1) satisfy conditions (1) and (2). If $|V|/3 \leq |V_1| \leq 2|V|/3$, then V_1 is the required value for V_m . If not, $|V_1| > 2|V|/3$, since otherwise $|V_1| + |V_2| \leq 2|V|/3$, contradicting $|U| > 2|V|/3$. Since C has a finite depth, and input gates have vertex set of size 2, we will successfully find a value for V_m . \square

The problem with proving the above lemma for general monotone circuits for BPM is that for a particular gate, there can be two minterms with different vertex sets. So we can not associate a ‘‘cut’’ with each gate such that none of its minterms crosses it.

Theorem 2. *Simple monotone circuits for BPM require exponential size.*

Proof. Let $n > 2$ and let C be a simple monotone circuit for perfect matching. Enumerate one set V_m satisfying the conditions of Lemma 1 for each possible perfect matching m on the bipartition $\{V', V'^c\}$. For each $U \subseteq V$ of size p , there are at most $(p/2)! \frac{n-p}{2}!$ perfect matchings that do not cross it (assuming G is a complete graph, this number is exactly $(p/2)! \frac{n-p}{2}!$ if U contains the same number of vertices from V' and V'^c and zero otherwise). Therefore, each $U \subseteq V$, $|V|/3 \leq |U| \leq 2|V|/3$ corresponds to at most $(n/6)!(n/3)!$ perfect matchings.

Since the number of perfect matchings in a complete bipartite graph is $(n/2)!$, we must have enumerated

$$\frac{(n/2)!}{(n/6)!(n/3)!} = n^{\Omega(1)} 2^{(n/2 \log 3 - n/3)} = \Omega(2^{.459n})$$

distinct subsets of V , each of them corresponding to a different gate of C . Therefore, the number of gates in C is $\Omega(2^{.459n})$. \square

3.2 Lower Bound for Multilinear Circuits for BPM

A circuit C is said to be *multilinear* if the edge set of any *AND* gate is the disjoint union of the edge set of its input gates. Unlike simple circuits, multilinear circuits are expressive enough to compute any monotone boolean function.

A circuit is said to be in the *simplest form* if it is simple and the vertex set of any *AND* gate is the disjoint union of the vertex set of its input gates. It can be seen that the circuits used to show upper bound on depth and size for *PM* were in the simplest form. Also note that circuits in simplest form are also multilinear.

Theorem 3. *A multilinear circuit of smallest possible size for BPM is also in the simplest form.*

Proof. Let C be a multilinear circuit for bipartite perfect matching on $\{V', V'^c\}$ of smallest possible size. We will define a *required vertex set* V_g for each gate g satisfying:

- (1) If g is an *AND* gate with input gates g_1 and g_2 , the required vertex of g is the disjoint union of the required vertex sets of g_1 and g_2 .
- (2) If g is an *OR* gate with input gates g_1 and g_2 , the required vertex sets of g , g_1 and g_2 are the same.
- (3) There is at least one minterm of g that is an exact V_g -matching.
- (4) All minterms of g have an edge to each vertex in V_g .
- (5) Let g be a gate with input gates g_1 and g_2 . Let the output from g_1 to g be replaced by the output of a new subcircuit that computes a function f . Assume that any minterm m of f is the superset of some minterm m' of g_1 , and all minterms of g_1 that are exact V_{g_1} -matchings are also minterms of f . We call f the *pruned* function of the output of g_1 (Note that *pruning* the output of a gate is defined with respect to the required vertex set of the gate). Let the output from g_2 to g also be replaced by a pruned function of the output of g_2 . Then the new output of g is a pruned function of the original output of g .

Intuitively, if an input gate g_1 to an *OR* gate g has no minterm that is an exact V_g -matching, we can replace the input from g_1 to g with a zero input without affecting the function calculated by the circuit. For gate g , the effect of this change is to increase the size of some minterms while some minterms drop out. But all minterms that are exact V_g -matchings are unaffected as they must

have been minterms of the other input gate to g . This pruning effect cascades all the way to the output gate. But the function calculated by the output gate will remain the same as all its minterms will be exact matchings on its required vertex set, namely V . We will apply a series of such changes.[†] Note that if the output of a gate is pruned multiple times, the resulting output function of the gate is also a pruning of the original output of the gate.

In the proof below, we might modify C by removing the connection between some gates. If at any time, there is no path from a gate to the output gate, we assume that the gate has been deleted without loss of generality (since such a gate can no longer affect the output of the circuit).

The required vertex set of the output gate is defined to be its vertex set V . It can be seen that it satisfies conditions (3) and (4). We define the required vertex set of a gate based on the required vertex sets of its parents (gates to which it supplies an input) in C . Once the required vertex set of a gate g is defined, it passes a *requirement*, a subset of V , to its children g_1 and g_2 as defined below:

Case 1: g is an AND gate: By property (3), $\exists m \in \text{minterm}(g)$ such that m is an exact V_g -matching. Therefore, $\exists m_1 \in \text{minterm}(g_1)$ and $\exists m_2 \in \text{minterm}(g_2)$ such that $m = m_1 \cup m_2$ and $m_1 \cap m_2 = \emptyset$ (since the edge sets of g_1 and g_2 are disjoint. This is because of the multilinearity of the original circuit and the fact that the subcircuit rooted at g has not yet been modified). The requirements passed to g_1 and g_2 are the vertex sets of m_1 and m_2 , say V_1 and V_2 , respectively.

Hence, m_1 is an exact V_1 -matching. Suppose some $m' \in \text{minterm}(g_1)$ does not have an edge to some $v \in V_1$. Then some subset m'' of $m' \cup m_2$ is a minterm of g . Also, m_2 does not have an edge to v since $m_1 \cup m_2$ was a $V_1 \cup V_2$ -matching and m_1 has an edge to v . But then m'' does not have an edge to a vertex v in the required vertex set of g , contradicting (4) for g . Therefore, all $m' \in \text{minterm}(g_1)$ have an edge to each vertex of V_1 .

This also means that any $m' \in \text{minterm}(g)$ which is an exact V_g -matching is produced by the disjoint union of $m'_1 \in \text{minterm}(g_1)$ and $m'_2 \in \text{minterm}(g_2)$ where m'_1 and m'_2 are exact V_1 -matching and exact V_2 -matching respectively (Since $m' = m'_1 \cup m'_2$ is an exact V_g -matching, if m'_1 is not an exact V_1 -matching, then $\exists e_1 \in m'_1$ such that e_1 has an endpoint $v_2 \in V_2$. But m'_2 has some edge e_2 incident on v_2 . For $m'_1 \cup m'_2$ to be a matching, e_1 and e_2 must be the same edge. But this contradicts the multilinearity of the original circuit). We will now show that condition (5) holds for g if we define the required vertex set of g_1 and g_2 to be V_1 and V_2 respectively. Suppose we replace the inputs from g_1 to g and g_2 to g in C with inputs from two new subcircuit C_1 and C_2 having output gates g'_1 and g'_2 respectively to get a new circuit C' . Let the outputs of g'_1 and g'_2 be pruned functions of the outputs of g_1 and g_2 respectively. Then if $m \in \text{minterm}_{C'}(g)$ then $\exists m_1 \in \text{minterm}_{C'}(g'_1)$ and $m_2 \in \text{minterm}_{C'}(g'_2)$ such

[†] In the original version of the paper, it is mentioned that applying one such change results in a multilinear circuit. This is not true since some minterms of the gates in the circuit can grow in the process. But we do get back multilinearity after applying all such possible changes. This is because all minterms of a gate that grew larger eventually drop out.

that $m = m_1 \cup m_2$. Since m_1 is the superset of some minterm of g_1 in C and m_2 is the superset of some minterm of g_2 in C , m is the superset of some minterm of g in C . That is, the output of g in C' is a pruned function of the output of g in C . Also, if $m \in \text{minterm}_C(g)$ and m is an exact $V_1 \cup V_2$ -matching, then $m \in \text{minterm}_{C'}(g)$ too. Therefore, the new output of the gate g is a pruned function of the original output of the gate.

Case 2: g is an OR gate: Suppose g_2 does not have any minterm that is an exact V_g -matching. Replace the input from g_2 to g with a zero input to get a circuit C' (This has the same affect as replacing the output from g with the output from g_1 and deleting g . But we will retain g for the moment to avoid changing parent-child relations we have already analyzed). If $m \in \text{minterm}_C(g)$ is an exact V_g -matching, then $m \notin \text{minterm}_C(g_2)$. Therefore $m \in \text{minterm}_C(g_1)$. This implies $m \in \text{minterm}_{C'}(g)$. If $m \in \text{minterm}_{C'}(g)$, then $m \in \text{minterm}_C(g_1)$. Therefore, $\exists m' \in \text{minterm}_C(g)$ such that $m' \subseteq m$. That is, the output of g in C' is a pruned function of the output of g in C . Therefore, applying property (5) to gate g and subsequently to all the gates along the paths to the output gate, we see that the outputs of circuits C and C' are the same. Replace the current circuit with the modified circuit C' . Note that the required vertex set of any gate for which the value had already been defined still satisfies conditions (3)-(5). The gate g then passes its required vertex set V_g as the requirement to g_1 . Gate g passes no requirement to g_2 since g is no longer its parent. To simplify the analysis later, we will assume that the zero input takes on the role of the second child g_2 of gate g . If both g_1 and g_2 have minterms that are exact V_g -matchings, g passes V_g as the requirement to both g_1 and g_2 . Below, we assume that g_1 has a minterm that is an exact V_g matching (since g has a minterm which is an exact V_g matching, one of g_1 and g_2 must have such a minterm) and hence was not disconnected from g .

Since every minterm of g_1 is the superset of some minterm of g , all minterms of g_1 have an edge to every vertex in V_g . The same holds for the minterms of g_2 .

Let us now analyze condition (5) for g assuming V_g is defined as the required vertex set of g_1 and g_2 . Let the inputs from g_1 and g_2 to g be replaced by inputs from two new subcircuits C_1 and C_2 with output gates g'_1 and g'_2 respectively. Let the output functions of g'_1 and g'_2 be pruned functions of the outputs of g_1 and g_2 respectively. Then if $m' \in \text{minterm}_{C'}(g)$, either $m' \in \text{minterm}_{C'}(g'_1)$ or $m' \in \text{minterm}_{C'}(g'_2)$. In either case, $\exists m \in \text{minterm}_C(g)$ such that $m \subseteq m'$. Also if $m \in \text{minterm}_C(g)$ is an exact V_g -matching, then either $m \in \text{minterm}_C(g_1)$ or $m \in \text{minterm}_C(g_2)$. Therefore, $m \in \text{minterm}_{C'}(g)$. That is, the output of g in C' is a pruned function of the output of g in C .

If for some gate g_1 , all its parents pass the same requirement V_0 , define V_0 to be the required vertex set for g_1 . Conditions (3)-(5) are satisfied for g_1 .

Suppose two parents g and g' of a gate g_1 pass different requirements V_1 and V_2 respectively. Without loss of generality, assume there exists vertex v satisfying $v \in V_1$, but $v \notin V_2$. Since g passed V_1 as requirement to g_1 , all minterms of g_1 have an edge to v . But since g' passed V_2 as requirement to g_1 , there exists a

minterm m of g_1 that is an exact V_2 -matching, and hence m does not have an edge to v , a contradiction. Therefore, all parents of g_1 pass the same requirement.

We will now show that the required vertex set of each gate is in fact its vertex set. Let g be a gate whose required vertex set is not the same as its vertex set. Therefore $\exists m \in \text{minterm}(g)$ with an edge e to a vertex outside its required vertex set. If g is an *OR* gate, one of its two input gates, say g' , has m as a minterm. Since the required vertex sets of g and g' are the same, g' too has an edge to a vertex v outside its required vertex set. If g is an *AND* gate with inputs g_1 and g_2 , then $\exists m_1 \in \text{minterm}(g_1)$ and $m_2 \in \text{minterm}(g_2)$ such that $m_1 \cup m_2 = m$. Let $e \in m_1$. Then g_1 has a minterm m_1 that has an edge to a vertex v outside its required vertex set (since the requirement passed by an *AND* gate to its input gate is a subset of its required vertex set). Since C has finite depth, we obtain a input gate whose required vertex set is not the same as its vertex set. But this is a contradiction of property (3).

Therefore, from conditions (1) and (2), the new circuit is in the simplest form. But if we removed the connection between an *OR* gate its child and replaced it by a zero input, we would have decreased the size of the circuit (because we can eliminate the *OR* gate as explained before). This would contradict the assumption that the original circuit was the smallest multilinear circuit for *BPM*. Hence a multilinear circuit for *BPM* of the smallest size is also in the simplest form. \square

Corollary 1. *Multilinear circuits for BPM require exponential size.*

Proof. This is easily seen from Theorem 2 and Theorem 3, since circuits in simplest form are also simple. \square

References

1. Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29:874–897, 1982.
2. Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 539–550. ACM Press, 1988.
3. P. M. Lewis II, R. E. Stearns, and J. Hartmanis. Memory bounds for recognition of context-free and context-sensitive languages. In *Conf. Record Switching Circ. Theory and Log. Des.*, pages 191–202, 1965.
4. Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. In *Proceedings of the 36th FOCS*, pages 16–25, 1996.
5. Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. In *Electronic Colloquium on Computational Complexity*, volume 67, 2003.
6. Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. In *ACM Symposium on Theory of Computing*, pages 287–292, 1990.
7. A. A. Razborov. Lower bounds on the monotone complexity of some boolean functions. *Doklady Akademii Nauk SSSR*, 281:798–801, 1985. In Russian. English translation in *Soviet Mathematics Doklady*, 31:354–357, 1985.

8. Walter L. Ruzzo. Tree-size bounded alternation. *Journal of Computer and System Sciences*, 21(2):218–235, 1980.
9. E. Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8:141–142, 1988.
10. Ingo Wegener. *The complexity of Boolean functions*. John Wiley & Sons, Inc., 1987.