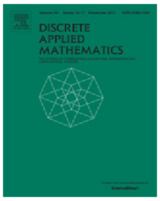
Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam



DISCRETE APPLIED MATHEMATICS

Lower bounds for monotone counting circuits[☆]

Stasys Jukna[1]

Institute of Computer Science, Goethe University, Frankfurt am Main, Germany

ARTICLE INFO

Article history:
Received 9 December 2014
Received in revised form 14 April 2016
Accepted 19 April 2016
Available online xxxx

Keywords:
Arithmetic circuits
Boolean circuits
Counting complexity
Lower bounds

ABSTRACT

A monotone arithmetic circuit *computes* a given multivariate polynomial f if its values on all nonnegative integer inputs are the same as those of f. The circuit *counts* f if this holds for 0–1 inputs; on other inputs, the circuit may output arbitrary values. The circuit *decides* f if it has the same 0–1 roots as f. We first show that some multilinear polynomials can be exponentially easier to count than to compute them, and that some polynomials can be exponentially easier to decide than to count them. Our main results are general lower bounds on the size of counting circuits.

1. Introduction

In this paper we consider computational complexity of multivariate polynomials with nonnegative integer coefficients:

$$f(x_1,\ldots,x_n)=\sum_{e\in E}c_e\prod_{i=1}^{n}x_i^{e_i}\,, \tag{1}$$

where $E\subset\mathbb{N}^n$ is a finite set of vectors of nonnegative integers, coefficients c_e are positive integers, and $x_i^0=1$; here and throughout, $\mathbb{N}=\{0,1,2,\ldots\}$. Each coefficient c_e stands for the number of times the *monomial* $p=\prod_{i=1}^{n}x_i^{e_i}$ appears in f; the *support* of such a monomial is the set $X_p=\{x_i\colon e_i\neq 0\}$ of variables appearing in it with nonzero exponents, and the *degree* of the monomial p is the sum $e_1+\cdots+e_n$ of its exponents. The polynomial is *multilinear* if $E\subseteq\{0,1\}^n$, and is *homogeneous* of degree d if all its monomials have the same degree d.

A natural model for compact representation of such polynomials (with nonnegative coefficients) is that of monotone arithmetic $(+,\times)$ circuits. Such a circuit is a directed acyclic graph with three types of nodes: input, addition $(+)$, and multiplication $(\times)$. Input nodes have fanin zero, and correspond to variables $x_1,\ldots,x_n$. All other nodes have fanin two, and are called *gates*. Each gate computes either the sum or product of its inputs. The *size* of a circuit is the number of gates in it.

Every such circuit syntactically *produces* a unique polynomial h with nonnegative integer coefficients in a natural way: the polynomial produced at an input gate x_i consists of a single monomial x_i, and the polynomial produced at a sum (product) gate is the sum (product) of polynomials produced at its inputs; we use distributivity to write a product of polynomials as a sum of monomials. The polynomial h produced by the circuit itself is the polynomial produced at its output gate. Given a polynomial $f(x_1,\ldots,x_n)$, we say that the circuit:

[☆] Research supported by the DFG grant SCHN 503/6-1.
E-mail address: jukna@thi.informatik.uni-frankfurt.de.
URL: http://www.thi.informatik.uni-frankfurt.de/~jukna/.

[1] Affiliated with Institute of Mathematics and Informatics, Vilnius University, Vilnius, Lithuania.

http://dx.doi.org/10.1016/j.dam.2016.04.024
0166-218X/© 2016 Elsevier B.V. All rights reserved.

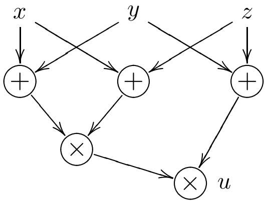


Fig. 1. A circuit of size 5 *computes* the polynomial $F = (x+y)(y+z)(x+z)$, *counts* the polynomial $f = 2xyz + 2xy + 2xz + 2yz$, and decides the polynomial $g = xy + xz + yz$. Gate u is the output gate.

- *computes* f (exactly) if $h(a) = f(a)$ holds for every $a \in \mathbb{N}^n$;
- *counts* f if $h(a) = f(a)$ holds for every $a \in \{0, 1\}^n$;
- *decides* f if for every $a \in \{0, 1\}^n$, $h(a) = 0$ exactly when $f(a) = 0$.

In this paper we are mainly interested in $(+, \times)$ circuits *counting* a given polynomial f. Such a circuit needs only to correctly compute the restriction $f : \{0, 1\}^n \to \mathbb{N}$ of f on 0–1 inputs. Note that, if the polynomial f is monic (has no coefficients > 1) then, on every 0–1 input $a \in \{0, 1\}^n$, the value $f(a)$ taken by f on a is the number of monomials of f satisfied by (evaluated to 1 on) a. For example, in the case of the *permanent polynomial*

$$\mathrm{Per}_n(x) = \sum_{\sigma} \prod_{i=1}^{n} x_{i,\sigma(i)} \tag{2}$$

with the summation over all permutations σ of $[n] = \{1, \dots, n\}$, its value $\mathrm{Per}_n(a)$ on every input $a \in \{0, 1\}^{n\times n}$ is the number of perfect matchings in the bipartite $n \times n$ graph G_a specified by a; nodes i and j are adjacent in G_a if and only if $a_{ij} = 1$. Thus, a circuit counting Per outputs the number of perfect matchings in G_a, whereas a circuit deciding this polynomial merely tells us whether G_a contains a perfect matching. On the other hand, *computing* circuits must actually solve the same counting problem but in the case when all nonnegative integers (not just 0 and 1) are allowed as weights.

Remark 1. Let us stress that we only consider *monotone* arithmetic circuits. The reason is that counting $(+, -, \times)$ circuits are already omnipotent: they are as powerful as boolean $\{\vee, \wedge, \neg\}$ circuits, for which no super-linear lower bounds are known so far. This holds because then each of the three boolean operations can be simulated over $\{0, 1\}$: $x \wedge y$ by $x \times y$, $\neg x$ by $1 - x$, and $x \vee y$ by $x + y - xy$.

If a $(+, \times)$ circuit computes, counts or only decides a given polynomial f, what can then be said about the structure of the *produced* by the circuit polynomial h? To answer these questions, we associate with every polynomial f the following three sets (this notation will be used throughout the paper):

- $\mathcal{M}(f)$ is the set of all monomials of f;
- $\mathcal{S}(f) = \left\{X_p : p \in \mathcal{M}(f)\right\}$ is the *support* of f;
- $\mathcal{L}(f) \subseteq \mathcal{S}(f)$ is the *lower support* of f consisting of all minimal sets of $\mathcal{S}(f)$; a set of a family of sets is *minimal* if it contains no other set of the family.

We have the following information about the structure of the produced by a circuit polynomial h (see Lemma 6). If the circuit:

- computes f then $h = f$, and hence, also $\mathcal{M}(h) = \mathcal{M}(f)$;
- counts f then $\mathcal{S}(h) = \mathcal{S}(f)$;
- decides f then $\mathcal{L}(h) = \mathcal{L}(f)$.

Thus, in the case of circuits exactly *computing* f we have a full knowledge about the produced by the circuit polynomial h: this polynomial must just coincide with f (the same monomials with the same coefficients). This ensures that no “invalid” monomials can be formed during the computation, and severely limits the power of such circuits. In particular, if the target polynomial f is homogeneous (all monomials have the same degree) then the circuit *itself* must be homogeneous: polynomials produced at its gates must be also homogeneous. If the target polynomial f is multilinear (no variable has degree larger than 1) then the circuit must be also multilinear: the polynomials produced at inputs of each product gate must depend on disjoint sets of variables. These limitations were essentially exploited in all known proofs of lower bounds for monotone arithmetic circuits, including [15,17,10,21,18,6,19,7].

In the case of *counting* circuits, $\mathcal{M}(h) = \mathcal{M}(f)$ needs not to hold, due to the multiplicative idempotence axiom $x^2 = x$ valid on 0–1 inputs. That is, here exponents (and hence, degrees of monomials) do not matter (see Fig. 1). For example, a polynomial $f = 2x + yz$ is counted by any circuit producing a polynomial of the form $h = 2x^a + y^b z^c$ with $a, b, c \in \mathbb{N} \setminus \{0\}$. That is, nonzero exponents of the monomials in produced by counting circuits polynomials may be arbitrary: we only know which sets of variables these monomials must contain, but we do not know their actual degrees. In *deciding* circuits, even $\mathcal{S}(h) = \mathcal{S}(f)$ needs not to hold, due to the additional absorption axiom $x + xy = x$.

Please cite this article in press as: S. Jukna, Lower bounds for monotone counting circuits, Discrete Applied Mathematics (2016), http://dx.doi.org/10.1016/j.dam.2016.04.024