Electronic Colloquium on Computational Complexity, Report No. 11 (2013)

Multilinear Complexity is Equivalent to Optimal Tester Size

Nader H. Bshouty
Technion, Israel
bshouty@cs.technion.ac.il

January 10, 2013

Abstract

In this paper we first show that Tester for an $\mathbb{F}$-algebra $\mathcal{A}$ and multilinear forms, [2], is equivalent to multilinear algorithm for the product of elements in $\mathcal{A}$, [3]. Our result is constructive in deterministic polynomial time. We show that given a tester of size ν for an $\mathbb{F}$-algebra $\mathcal{A}$ and multilinear forms of degree d one can in deterministic polynomial time construct a multilinear algorithm for the multiplication of d elements of the algebra of multilinear complexity ν and vise versa.

This with the constructions in [2] give the first polynomial time construction of a bilinear algorithm with linear bilinear complexity for the multiplication of two elements in any extension finite field.

We then study the problem of simulating a substitution of an assignment from an $\mathbb{F}$-algebra $\mathcal{A}$ in a degree d multivariate polynomials with substitution of assignments from the ground field $\mathbb{F}$. We give a complete classification of all algebras for which this can be done and show that this problem is equivalent to constructing symmetric multilinear algorithms [11] for the product of d elements in $\mathcal{A}$.

1 Introduction

Let $\mathbb{F}$ be a field and $\mathcal{A}$ be an $\mathbb{F}$-algebra of finite dimension with unity element $1_{\mathcal{A}}$. A *tester* for $\mathcal{A}$ and a class of multivariate polynomial $\mathcal{M}$ over n variables is a set L of (not necessarily linear) maps from $\mathcal{A}^n$ to $\mathbb{F}^n$ that preserve the property $f(\boldsymbol{a}) \neq 0$ for every $f \in \mathcal{M}$, i.e., for all $f \in \mathcal{M}$ and $\boldsymbol{a} \in \mathcal{A}^n$ if $f(\boldsymbol{a}) \neq 0$ then $f(\ell(\boldsymbol{a})) \neq 0$ for some $\ell \in L$. The *size* of the tester is $\nu := |L|$. In [2], we use tools from elementary algebra and algebraic

ISSN 1433-8092

function fields to construct testers of asymptotically optimal size in deterministic polynomial time. Testers have many applications. See [2] for more details.

The *multilinear complexity* of the multiplication of d elements in the $\mathbb{F}$-algebra $\mathcal{A}$ is the minimal μ such that there are $\lambda_{i,j} \in \mathcal{A}^*$, $i = 1, \ldots, \mu$, $j = 1, \ldots, d$ and $\gamma_1, \gamma_2, \ldots, \gamma_\mu \in \mathcal{A}$ where for every d elements $a_1, a_2, \ldots, a_d$ in $\mathcal{A}$ we have

$$a_1 a_2 \cdots a_d = \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d} \lambda_{i,j}(a_j).$$

When $d = 2$ the multilinear complexity is called the *bilinear complexity* of the multiplication of two elements in the algebra $\mathcal{A}$. Bilinear complexity is extensively studied in the literature. See for example [3, 6] and references within. Also, all the algorithms known for matrix multiplication are bilinear algorithms [8], i.e., algorithms of the above form.

In this paper we show that when $\mathcal{M}$ is the class of multilinear forms of degree d then the size of the optimal tester for $\mathcal{M}$ and the $\mathbb{F}$-algebra $\mathcal{A}$ is exactly equal to the multilinear complexity of the multiplication of d elements in the algebra $\mathcal{A}$. Our result is constructive in deterministic polynomial time. We show that given a tester for $\mathcal{M}$ and an $\mathbb{F}$-algebra $\mathcal{A}$ of size ν one can in polynomial time construct a multilinear algorithm for the multiplication of d elements of the algebra of multilinear complexity ν and vise versa.

One of the open problems in bilinear complexity is to give a polynomial time construction of a bilinear algorithm for the multiplication of two elements in the extension field $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ with bilinear complexity $O(n)$. Here $\mathbb{F}_q$ is the finite field with q elements. Many nonconstructive algorithms are known for this problem that has linear bilinear complexity [1, 12, 5, 4, 10]. See also all the references within [10]. In [7], Lemple et. al. gave a deterministic polynomial time construction of bilinear algorithm with bilinear complexity $O((\log^* n)n)$. In [2] we gave a deterministic polynomial time construction of a tester for the class of bilinear forms and the field $\mathbb{F}_{q^n}$ of size $O(n)$. This with the above result solve this open problem.

Another result that follows from our technique is the following. For a black box that contains degree d polynomial over any field $\mathbb{F}$, we show that given a symmetric multilinear algorithm [11] (see also definition in Section 2) for the multiplication of $d+1$ elements in the $\mathbb{F}$-algebra $\mathcal{A}$ of multilinear complexity μ, one can simulate a black box assignment substitution of elements of the algebra with μ black box assignment substitutions of elements of the ground field $\mathbb{F}$. Does multilinear algorithms give optimal simulation?

In this paper we give an affirmative answer to this question. We show that from any simulation that has complexity ν one can in polynomial time construct a symmetric multilinear algorithm of the multiplication of d elements in the algebra that has multilinear complexity ν. This means that symmetric multilinear algorithms give optimal simulations.

We believe that the results of this paper open the possibility to further study of multilinear complexities of algebras for dimension greater than 2 and to try to understand more the algebraic structure of such algorithms.

2 Definitions

In this section we define multivariate polynomials and forms, tester, the multilinear complexity of the multiplication of elements in an algebra and then simulators.

We note that throughout the paper, bold small letters, say $\boldsymbol{a}$, denotes vectors and its ith entry is denoted by a_i.

2.1 Multivariate Polynomial

In this section we define the set of multivariate polynomials over a field.

Let $\mathbb{F}$ be a field and $\boldsymbol{x} = (x_1, \ldots, x_n)$ be indeterminates (or variables) over the field $\mathbb{F}$. The ring of *multivariate polynomials* in the indeterminates $x_1, \ldots, x_n$ over $\mathbb{F}$ is $\mathbb{F}[x_1, \ldots, x_n]$ (or $\mathbb{F}[\boldsymbol{x}]$). Let $\boldsymbol{i} = (i_1, \ldots, i_n) \in \mathbf{N}^n$ where $\mathbf{N} = \{0, 1, 2, \ldots\}$. We denote by $\boldsymbol{x}^{\boldsymbol{i}}$ the *monomial* $x_1^{i_1} \cdots x_n^{i_n}$. Every multivariate polynomial in $f \in \mathbb{F}[\boldsymbol{x}]$ can be represented as

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{i} \in I} a_{\boldsymbol{i}} \boldsymbol{x}^{\boldsymbol{i}} \tag{1}$$

for some finite set $I \subset \mathbf{N}^n$ and $a_{\boldsymbol{i}} \in \mathbb{F} \backslash \{0\}$ for all $\boldsymbol{i} \in I$.

When the field $\mathbb{F}$ is infinite, then the representation in (1) is unique. Not every function $f : \mathbb{F}^n \to \mathbb{F}$ can be represented as multivariate polynomial. Take for example a function $f(x_1)$ with one variable that has infinite number of roots.

When the field $\mathbb{F}$ is finite, then every function $f : \mathbb{F}^n \to \mathbb{F}$ can be represented as multivariate polynomial $f \in \mathbb{F}[\boldsymbol{x}]$. There may be many representations for the same function $f : \mathbb{F}^n \to \mathbb{F}$ but a unique one that satisfies $\boldsymbol{i} \in I \subseteq \{0, 1, \ldots, |\mathbb{F}| - 1\}^n$. In the sequel we refer to this unique representation as the *minimal multivariate polynomial*.

When we say that $M = \boldsymbol{x}^{\boldsymbol{i}}$ is a *monomial in* f we mean that M is a monomial that appears in the minimal multivariate polynomial of f. The

constant $a_{\boldsymbol{i}} \in \mathbb{F}\backslash\{0\}$ in (1) is called the *coefficient* of the monomial $\boldsymbol{x}^{\boldsymbol{i}}$ in f. When $\boldsymbol{x}^{\boldsymbol{i}}$ is not a monomial in f then we say that its coefficient is 0.

For $f, g \in \mathbb{F}[\boldsymbol{x}]$ we define $\text{coef}(f, g)$ as follows: If $g = M$ is a monomial in f then $\text{coef}(f, M)$ is the coefficient of M in the minimal multivariate polynomial of f. Otherwise $\text{coef}(f, M) = 0$. If $g = cM$ where $c \in \mathbb{F}$ then $\text{coef}(f, cM) = c \cdot \text{coef}(f, M)$ and if $g = g_1 + g_2$ where $g_1, g_2 \in \mathbb{F}[\boldsymbol{x}]$ then $\text{coef}(f, g) = \text{coef}(f, g_1) + \text{coef}(f, g_2)$.

The *degree*, $\deg(M)$, of a monomial $M = \boldsymbol{x}^{\boldsymbol{i}}$ is $i_1 + i_2 + \cdots + i_n$. The *degree of* x_j *in* M, $\deg_{x_j}(M)$ is i_j. Therefore,

$$\deg(M) = \sum_{i=1}^{n} \deg_{x_i}(M).$$

Let $f \in \mathbb{F}[\boldsymbol{x}]$ and let g be the minimal multivariate polynomial of f. The *degree* (or *total degree*) $\deg(f)$ is the maximum degree of the monomials in g. The degree of x_i in f, $\deg_{x_i}(f)$, is the maximum degree of x_i in the monomials in g. The *variable degree* of f is the maximum over the degree of each variable in f, i.e., $\max_i \deg_{x_i}(f)$.

2.1.1 Classes of Multivariate Polynomials

In this section we will define the following classes of multivariate polynomials

1. $\mathcal{P}(\mathbb{F}, n)$ is the class of all multivariate polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ of variable degree at most $|\mathbb{F}| - 1$.

2. $\mathcal{P}(\mathbb{F}, n, (d, r))$ is the class of all multivariate polynomials in $\mathcal{P}(\mathbb{F}, n)$ of degree at most d and variable degree at most r.

3. $\mathcal{P}(\mathbb{F}, n, d) = \mathcal{P}(\mathbb{F}, n, (d, |\mathbb{F}| - 1))$ is the class of all multivariate polynomials in $\mathcal{P}(\mathbb{F}, n)$ of degree at most d.

We will also consider the following special classes of multivariate polynomials

1. $\mathcal{HP}(\mathbb{F}, n)$: The class of all homogeneous polynomials in $\mathcal{P}(\mathbb{F}, n)$. A multivariate polynomial is called *homogeneous multivariate polynomial* if all its monomials have the same degree. The classes $\mathcal{HP}(\mathbb{F}, n, (d, r))$ and $\mathcal{HP}(\mathbb{F}, n, d)$ are defined in the same way as above.

2. $\mathcal{LP}(\mathbb{F}, n)$: The class of all multilinear polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. A multivariate polynomial is called *multilinear polynomial* if $\deg_{x_i}(f) \leq 1$ for all $i = 1, \ldots, n$. The class $\mathcal{ML}(\mathbb{F}, n, d)$ is defined in the same way as above.

2.2 Multivariate Form

Let $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$ where $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,n})$ are indeterminates over $\mathbb{F}$ for $i = 1, \ldots, m$. A *multivariate form* in $\boldsymbol{y}$ is a multivariate polynomial in $\boldsymbol{y}$. That is, an element of

$$\mathbb{F}[y_{1,1}, \ldots, y_{1,n}, \ldots, y_{m,1}, \ldots, y_{m,n}].$$

We denote this class by $\mathbb{F}[\boldsymbol{y}]$ or $\mathbb{F}[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m]$.

Let $\boldsymbol{y}^{(i)} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{i-1}, \boldsymbol{y}_{i+1}, \ldots, \boldsymbol{y}_m)$. Every multivariate form $f \in \mathbb{F}[\boldsymbol{y}]$ can be represented as polynomial in $\boldsymbol{y}_i$ with coefficients in $\mathbb{F}[\boldsymbol{y}^{(i)}]$. That is, a multivariate polynomial in $\mathbb{F}[\boldsymbol{y}^{(i)}][\boldsymbol{y}_i]$. The degree $\deg_{\boldsymbol{y}_i}(f)$ is the degree of f in $\mathbb{F}[\boldsymbol{y}^{(i)}][\boldsymbol{y}_i]$. Every monomial M in $\mathbb{F}[\boldsymbol{y}]$ can be written as $M = M_1 M_2 \cdots M_m$ where M_i is a monomial in $\mathbb{F}[\boldsymbol{y}_i]$ for $i = 1, 2, \ldots, m$.

The following classes will be studied here

1. $\mathcal{F}(\mathbb{F}, n, m)$ is the class of all multivariate forms in $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$, where $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,n})$ are indeterminates over $\mathbb{F}$, of variable degree at most $|\mathbb{F}| - 1$.

2. $\mathcal{F}(\mathbb{F}, n, m, \boldsymbol{d})$ where $\boldsymbol{d} = (d_1, d_2, \ldots, d_m) \in \mathbf{N}^m$ is the class of all multivariate polynomials f in $\mathcal{F}(\mathbb{F}, n, m)$ where $\deg_{\boldsymbol{y}_i}(f) = d_i$ for all $i = 1, 2, \ldots, m$.

3. $\mathcal{HF}(\mathbb{F}, n, m)$ is the class of all multivariate forms f in $\mathcal{F}(\mathbb{F}, n, m)$ that are homogeneous in $\mathbb{F}[\boldsymbol{y}^{(i)}][\boldsymbol{y}_i]$ for all $i = 1, 2, \ldots, m$. That is, there is $\boldsymbol{d} \in \mathbf{N}^m$ such that every monomial in f is of the form $M = M_1 M_2 \cdots M_m$, where M_i is a monomial in $\mathbb{F}[\boldsymbol{y}_i]$ of degree d_i. The class $\mathcal{HF}(\mathbb{F}, n, m, \boldsymbol{d})$ is defined as above.

4. $\mathcal{HLF}(\mathbb{F}, n, m)$ is $\mathcal{HF}(\mathbb{F}, n, m, (1, 1, \ldots, 1))$. That is, the class of all *multilinear forms* f over $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$ where each monomial in f contains exactly one variable from $\boldsymbol{y}_i$ for every i.

Therefore a *multilinear form* in $\boldsymbol{y}$ of degree d is a multivariate polynomial of the form

$$\sum_{\boldsymbol{i} \in [n]^d} \alpha_{\boldsymbol{i}} \cdot y_{1,i_1} \cdots y_{d,i_d}.$$

where $\alpha_{\boldsymbol{i}} \in \mathbb{F}$ and $[n] = \{1, 2, \ldots, n\}$.

2.3 Tester

Definition 1. (**Tester,[2]**). *Let $\mathbb{F}$ be a field and $\mathcal{A}$ be an $\mathbb{F}$-algebra with unity element $1_{\mathcal{A}}$. Let $\mathcal{M} \subseteq \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a class of multivariate polynomials. Let $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\nu\}$ be a set of (not necessarily linear) maps $\mathcal{A}^n \to \mathbb{F}^n$. We denote by fL the map $\mathcal{A}^n \to \mathbb{F}^\nu$ where for $\boldsymbol{a} \in \mathcal{A}^n$,*

$$(fL)(\boldsymbol{a}) = (f(\boldsymbol{l}_1(\boldsymbol{a})), \ldots, f(\boldsymbol{l}_\nu(\boldsymbol{a}))).$$

We say that L is an $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester (or a tester for $\mathcal{M}$ and $\mathcal{A}$) if for every $\boldsymbol{a} \in \mathcal{A}^n$ and $f \in \mathcal{M}$ we have (Here $\mathbf{0} = 0^\nu$ is the zero vector of length ν)

$$(fL)(\boldsymbol{a}) = \mathbf{0} \implies f(\boldsymbol{a}) = 0.$$

The integer $\nu = |L|$ is called the *size of the tester*. The minimal size of an $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester is denoted by $\nu_{\mathbb{F}}(\mathcal{M}, \mathcal{A})$. When $\mathcal{A}$ and $\mathbb{F}$ are known from the context we then just say that L is a *tester for* $\mathcal{M}$.

We will also allow $L = \{l_1, \ldots, l_\nu\}$ to be a set of maps $\mathcal{A} \to \mathbb{F}$. In that case $\boldsymbol{l}_i : \mathcal{A}^n \to \mathbb{F}^n$ is defined as $\boldsymbol{l}_i(\boldsymbol{a}) = (l_i(a_1), \ldots, l_i(a_n))$ where $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathcal{A}^n$. In such case we call the tester *symmetric tester*.

We say that the tester is *componentwise tester* if $\boldsymbol{l}_i(\boldsymbol{a}) = (l_{i,1}(a_1), \ldots, l_{i,n}(a_n))$ for some $l_{i,j} : \mathcal{A} \to \mathbb{F}$. A componentwise tester is called *linear* if each $l_{i,j}$ is a linear map and is called *reducible* if $l_{i,j}(1_{\mathcal{A}}) = 1$ where $1_{\mathcal{A}}$ is the identity of the algebras $\mathcal{A}$.

In this paper we will study testers for the class of multilinear forms of degree d and multivariate polynomials of degree d.

We will use the following abbreviations

The Expression	Abbreviation
$\nu_{\mathbb{F}}(\mathcal{P}(\mathbb{F}, n, d), \mathcal{A})$	$\nu_{\mathbb{F}}^{\mathcal{P}}(d, \mathcal{A})$
$\nu_{\mathbb{F}}(\mathcal{P}(\mathbb{F}, n, (d, r)), \mathcal{A})$	$\nu_{\mathbb{F}}^{\mathcal{P}}((d, r), \mathcal{A})$
$\nu_{\mathbb{F}}(\mathcal{HP}(\mathbb{F}, n, d), \mathcal{A})$	$\nu_{\mathbb{F}}^{\mathcal{HP}}(d, \mathcal{A})$
$\nu_{\mathbb{F}}(\mathcal{F}(\mathbb{F}, n, m, \boldsymbol{d}), \mathcal{A})$	$\nu_{\mathbb{F}}^{\mathcal{F}}((\boldsymbol{d}, \mathcal{A})$
$\nu_{\mathbb{F}}(\mathcal{HF}(\mathbb{F}, n, m, \boldsymbol{d}), \mathcal{A})$	$\nu_{\mathbb{F}}^{\mathcal{HF}}(\boldsymbol{d}, \mathcal{A})$
$\nu_{\mathbb{F}}(\mathcal{HLF}(\mathbb{F}, n, m), \mathcal{A})$	$\nu_{\mathbb{F}}^{\mathcal{HLF}}(m, \mathcal{A})$

2.4 Multilinear Algorithms

A *multilinear algorithm* for the multiplication of d elements in the algebra $\mathcal{A}$ with *multilinear complexity* μ is a sequence $(\gamma_1, \boldsymbol{\lambda}_1), \ldots, (\gamma_\mu, \boldsymbol{\lambda}_\mu) \in \mathcal{A} \times (\mathcal{A}^*)^d$

and $\gamma_1, \gamma_2, \ldots, \gamma_\mu \in \mathcal{A}$ such that for every d elements $a_1, a_2, \ldots, a_d$ in $\mathcal{A}$ we have

$$a_1 a_2 \cdots a_d = \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d} \lambda_{i,j}(a_j).$$

The minimal μ is called the *multilinear complexity* of the multiplication of d elements in the algebra $\mathcal{A}$ and is denoted by $\mu_{\mathbb{F}}(d, \mathcal{A})$. When $\lambda_{i,j}$ are independent of j, i.e., the multilinear algorithm is of the form

$$a_1 a_2 \cdots a_d = \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d} \lambda_i(a_j),$$

then we call the multilinear algorithm *symmetric multilinear algorithm*. The symmetric multilinear complexity of the multiplication of d elements in the algebra $\mathcal{A}$ is denoted by $\mu^{s}_{\mathbb{F}}(d, \mathcal{A})$.

Let $1_{\mathcal{A}}$ be the identity element of the algebra $\mathcal{A}$. We may assume that $\lambda_{i,j}(1_{\mathcal{A}})$ is either 1 or 0. Because otherwise we can replace $\lambda_{i,j}$ with $\lambda_{i,j}/\lambda_{i,j}(1_{\mathcal{A}})$ and γ_i with $\gamma_i \lambda_{i,j}(1_{\mathcal{A}})$. Therefore, throughout the paper we assume $\lambda_{i,j}(1_{\mathcal{A}}) \in \{0, 1\}$.

If in a symmetric multilinear algorithm $\lambda_i(1_{\mathcal{A}}) = 1$ for all i then we call the multilinear algorithm *reducible symmetric multilinear algorithm*. The reducible symmetric multilinear complexity of the multiplication of d elements in the algebra $\mathcal{A}$ is denoted by $\mu^{rs}_{\mathbb{F}}(d, \mathcal{A})$.

In [2] (also follows from folklore results for bilinear algorithms [3]) we show that multilinear algorithms for the multiplication of d elements in any algebra always exist. In Section 7 we show that symmetric multilinear algorithms for the multiplication of d elements in an $\mathbb{F}$-algebra $\mathcal{A}$ exist if and only if either $|\mathbb{F}| \geq d$ or $|\mathbb{F}| < d$ and $a^{|\mathbb{F}|} = a$ for all $a \in \mathcal{A}$. Reducible symmetric multilinear algorithms for the multiplication of d elements in an $\mathbb{F}$-algebra $\mathcal{A}$ exist if and only if either $|\mathbb{F}| \geq d+1$ or $|\mathbb{F}| < d+1$ and $a^{|\mathbb{F}|} = a$ for all $a \in \mathcal{A}$.

When $d = 2$ the above multilinear complexity is called the *bilinear complexity* (resp. *symmetric bilinear algorithm*) of the multiplication of two elements in the algebra $\mathcal{A}$. The bilinear complexity were extensively studied in the literature. See for example [3, 6] and the references within. Symmetric bilinear complexity was first defined and studied in [11].

2.5 Simulators

Given a class of multivariate polynomial $\mathcal{M} \subseteq \mathbb{F}[x_1, \ldots, x_n]$. Given a black box that contains a function $f \in \mathcal{M}$ and can answer substitution oracle to

f. That is, for $\boldsymbol{b} \in \mathbb{F}^n$, the oracle answers $f(\boldsymbol{b})$. A *simulator for $\mathcal{M}$ and an algebra $\mathcal{A}$* is an algorithm that for every $\boldsymbol{a} \in \mathcal{A}^n$ generates $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_t \in \mathbb{F}^n$ such that for every $f \in \mathcal{M}$ the values $f(\boldsymbol{a}_1), \ldots, f(\boldsymbol{a}_t)$ uniquely determine $f(\boldsymbol{a})$. We say that the simulator is *polynomial time simulator* if the simulator generates $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_t$ from $\boldsymbol{a}$ in polynomial time and $f(\boldsymbol{a})$ from $f(\boldsymbol{a}_1), \ldots,$ $f(\boldsymbol{a}_t)$ in polynomial time. That is, the simulator runs in polynomial time. The maximal integer t over all $\boldsymbol{a} \in \mathcal{A}$ is called the *simulation complexity of the simulator* and is denoted by $\sigma_{\mathbb{F}}(\mathcal{M}, \mathcal{A})$. The simulator is called optimal if the simulation complexity is optimal.

We will use the same abbreviations used in Section 2.3. For example, $\sigma^{\mathcal{P}}_{\mathbb{F}}(d, \mathcal{A})$ is $\sigma_{\mathbb{F}}(\mathcal{P}(\mathbb{F}, n, d), \mathcal{A})$.

In Section 7 we show that simulators for multivariate polynomials of degree d and $|\mathbb{F}|$-algebra $\mathcal{A}$ exist if and only if either $|\mathbb{F}| \geq d+1$ or $|\mathbb{F}| < d+1$ and $a^{|\mathbb{F}|} = a$ for all $a \in \mathcal{A}$.

3 Results

The first result in this paper shows that testers for multilinear forms and multilinear algorithms are equivalent

Theorem 1. *Let $\mathcal{A}$ be $\mathbb{F}$-algebra and $\mathcal{M} = \mathcal{HLF}(\mathbb{F}, n, d)$ be a class of multilinear forms of degree d. Then*

1. $\nu^{\mathcal{HLF}}_{\mathbb{F}}(d, \mathcal{A}) = \mu_{\mathbb{F}}(d, \mathcal{A})$.
2. *Given a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν one can in polynomial time construct a multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν.*
3. *Given a multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν.*
4. *Given a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν one can in polynomial time construct a linear componentwise $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν.*

In [2] we gave a deterministic polynomial time construction of an ($\mathcal{HLF}(\mathbb{F}, n, d)$

$, \mathbb{F}_{q^n}, \mathbb{F}_q)$-tester of size $\nu = O(d^{\tau(d,q)}n)$ where

$$\tau(d,q) = \begin{cases} 3 & \text{if } q \geq cd^2, c > 1 \text{ constant }, q \text{ perfect square} \\ 4 & \text{if } q \geq cd, c > 1 \text{ constant} \\ 5 & \text{if } q \geq d+1 \\ 6 & \text{if } q = d \end{cases} \tag{2}$$

We also proved the lower bound $\Omega(dn)$ when $q \geq d$. Therefore our construction (for $q \geq d$) is within at most $O(d^5)$ of the optimal size tester.

Theorem 1 with (2), gives

Corollary 2. *For any $q \geq d$ there is a polynomial time construction of a multilinear algorithm for the multiplication of d elements in $\mathbb{F}_{q^n}$ with multilinear complexity $\mu = O(d^{\tau(d,q)}n)$.*

In particular, we also prove

Corollary 3. *For any q there is a polynomial time construction of a bilinear algorithm for the multiplication of two elements in $\mathbb{F}_{q^n}$ with bilinear complexity $\mu = O(n)$.*

This solves the open problem of deterministic polynomial time constructing a bilinear algorithm with linear bilinear complexity for the multiplication of two elements in finite fields [1, 12, 4].

To study testers over any multivariate polynomials we need the $\mathbb{F}$-algebra to be commutative $\mathbb{F}$-algebra. Otherwise, we have to deal with multivariate polynomials with noncommutative indeterminates, another line of research that is left for future study. Therefore in all the following results we assume that $\mathcal{A}$ is commutative $\mathbb{F}$-algebra.

In Section 5 we study testers for homogeneous multivariate polynomials and show that they are equivalent to symmetric multilinear algorithms. We prove

Theorem 2. *Let $\mathcal{A}$ be a commutative $\mathbb{F}$-algebra and $\mathcal{M} = \mathcal{HP}(\mathbb{F}, n, d)$ be a class of homogeneous multivariate polynomials of degree d. Then*

1. $\nu_{\mathbb{F}}^{\mathcal{HP}}(d, \mathcal{A}) = \mu_{\mathbb{F}}^{s}(d, \mathcal{A})$.
2. *Given a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν one can in polynomial time construct a symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν.*

3. *Given a symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν.*

4. *Given a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν one can in polynomial time construct a linear symmetric $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν.*

We note here that since (2) is also true for symmetric testers Corollaries 2 and 3 are also true for symmetric multilinear algorithms.

In Section 6 we study simulators and prove that symmetric multilinear algorithms are "almost" equivalent to simulators. We prove

Theorem 3. *we have the following*

1. *Given a reducible symmetric multilinear algorithm for the multiplication of d elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a polynomial time simulator for the class of polynomials of degree d and $\mathcal{A}$ of simulation complexity ν.*

   In particular,
   $$\sigma^{\mathcal{P}}_{\mathbb{F}}(d, \mathcal{A}) \leq \mu^{rs}_{\mathbb{F}}(d, \mathcal{A}).$$

2. *From any polynomial time simulator for the set of all multivariate polynomials of degree d and an $\mathbb{F}$-algebra $\mathcal{A}$ of simulation complexity ν one can construct in polynomial time a symmetric multilinear algorithm for the product of d elements in $\mathcal{A}$ of multilinear complexity ν.*

   In particular,
   $$\mu^{s}_{\mathbb{F}}(d, \mathcal{A}) \leq \sigma^{\mathcal{P}}_{\mathbb{F}}(d, \mathcal{A}).$$

We then study some connections between symmetric multilinear algorithms and reducible symmetric multilinear algorithms. We prove

Theorem 4. *we have the following*

1. *Given a symmetric multilinear algorithm for the multiplication of $d+1$ elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν. In particular*
   $$\mu^{rs}_{\mathbb{F}}(d, \mathcal{A}) \leq \mu^{s}_{\mathbb{F}}(d+1, \mathcal{A})$$

and

$$1 \leq \frac{\sigma^{\mathcal{P}}_{\mathbb{F}}(d,\mathcal{A})}{\mu^{s}_{\mathbb{F}}(d,\mathcal{A})} \leq \frac{\mu^{s}_{\mathbb{F}}(d+1,\mathcal{A})}{\mu^{s}_{\mathbb{F}}(d,\mathcal{A})}.$$

2. *If $|\mathbb{F}| \geq d+1$ then given a symmetric multilinear algorithm for the multiplication of d elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity $(d+1)\nu$. In particular, if $|\mathbb{F}| \geq d+1$ then*

$$\mu^{rs}_{\mathbb{F}}(d,\mathcal{A}) \leq (d+1)\cdot\mu^{s}_{\mathbb{F}}(d,\mathcal{A}) - d\cdot\mu^{rs}_{\mathbb{F}}(d-1,\mathcal{A})$$

and

$$1 \leq \frac{\sigma^{\mathcal{P}}_{\mathbb{F}}(d,\mathcal{A})}{\mu^{s}_{\mathbb{F}}(d,\mathcal{A})} \leq d+1.$$

3. *If $|\mathbb{F}| = \infty$ (or large enough) then given a symmetric multilinear algorithm for the multiplication of d elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν. In particular, if $|\mathbb{F}| = \infty$ then*

$$\sigma^{\mathcal{P}}_{\mathbb{F}}(d,\mathcal{A}) = \mu^{rs}_{\mathbb{F}}(d,\mathcal{A}) = \mu^{s}_{\mathbb{F}}(d,\mathcal{A}).$$

We believe that $\mu^{s}_{\mathbb{F}}(d+1,\mathcal{A}) = O(\mu^{s}_{\mathbb{F}}(d,\mathcal{A}))$ for any algebra $\mathcal{A}$. This will implies $\sigma^{\mathcal{P}}_{\mathbb{F}}(d,\mathcal{A}) = \Theta(\mu^{s}_{\mathbb{F}}(d,\mathcal{A}))$.

For $\mathcal{A} = \mathbb{F}_{q^n}$ over $\mathbb{F}_q$, Theorem 6 below shows that no simulator exists when $|\mathbb{F}| < d+1$. When $|\mathbb{F}| \geq d+1$ Theorem 4 shows that the complexity of simulators constructed from optimal symmetric multilinear algorithms for the multiplication of d elements in $\mathbb{F}_{q^n}$ are within a factor of $d+1$ of the optimal simulators complexity for $\mathcal{P}(\mathbb{F}_q, n, d)$ and $\mathbb{F}_{q^n}$.

This gives another motivation for further study of multilinear complexity of the product of d elements in an extended finite field for $d > 2$.

Another result that follows from Theorem 2 and Theorem 3 is

Corollary 4. *Given a tester for the class of all polynomials of degree $d+1$ and $\mathcal{A}$ of size ν that runs in polynomial time one can in polynomial time construct a simulator of simulation complexity ν that runs in polynomial time.*

This result is a surprising result since testers only test if $f(\boldsymbol{a}) = 0$ for $\boldsymbol{a} \in \mathcal{A}^n$ where simulators requires computing $f(\boldsymbol{a})$ which can be any element in $\mathcal{A}$.

It is known that every $\mathbb{F}$-algebra has a multilinear algorithm for the multiplication d elements of the algebra for any d. See the case $d = 2$ in [3]. The generalization to any d is trivial. In Section 7 we give a complete classification of all algebras that has symmetric multilinear algorithm and reducible symmetric algorithm. We prove

Theorem 5. *Let $\mathcal{A}$ be $\mathbb{F}$-algebra. There is a symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ if and only if $\mathcal{A}$ is commutative and one of the following conditions is true*

1. $|\mathbb{F}| \geq d$
2. $|\mathbb{F}| < d$ *and for every element* $a \in \mathcal{A}$, *we have* $a^{|\mathbb{F}|} = a$.

Theorem 6. *Let $\mathcal{A}$ be an $\mathbb{F}$-algebra. The following conditions are equivalent*

1. *There is a simulator algorithm for multivariate polynomials of degree d and $\mathcal{A}$.*
2. *$\mathcal{A}$ is commutative algebra and one of the following conditions is true*
   (a) $|\mathbb{F}| \geq d + 1$
   (b) $|\mathbb{F}| < d + 1$ *and for every element* $a \in \mathcal{A}$ *we have* $a^{|\mathbb{F}|} = a$.
3. *There is a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$.*

The proofs of Theorems 1 and 2 will be presented for the case $d = 2$, that is, for the bilinear forms, quadratic forms (i.e., homogenous multivariate polynomials of degree 2), multivariate polynomials of degree 2, bilinear complexity and symmetric bilinear complexity. The extension to any dimension is straightforward. The proofs of the other results will be presented for any d.

4 Testers and Multilinear Algorithms

In this section we prove the following Theorem 1 for $d = 2$. The proof for any d is straightforward generalization of the proofs in this section

Theorem 1. *Let $\mathcal{A}$ be $\mathbb{F}$-algebra and $\mathcal{M} = \mathcal{HLF}(\mathbb{F}, n, d)$ be a class of multilinear forms of degree d. Then*

1. $\nu_{\mathbb{F}}^{\mathcal{HLF}}(d, \mathcal{A}) = \mu_{\mathbb{F}}(d, \mathcal{A})$.

2. *Given a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν one can in polynomial time construct a multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν.*

3. *Given a multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν.*

4. *Given a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν one can in polynomial time construct a linear componentwise $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν.*

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^T$ be (column) vectors of distinct indeterminates. Let $\mathcal{B} = \mathcal{HLF}(\mathbb{F}, n, 2) = \{\boldsymbol{x}^T A \boldsymbol{y} \mid A \in \mathbb{F}^{n \times n}\} \subset \mathbb{F}[\boldsymbol{x}, \boldsymbol{y}]$ be the set of bilinear forms. For two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ we denote by $\boldsymbol{a} \otimes \boldsymbol{b}$ the Kronecker product of vectors. All the vectors in this paper are column vectors and ^{T} is the transpose of vectors (or matrices). For an $n \times m$ matrix A we denote by $vec(A)$ the the vector of length nm where $vec(A)_{(i-1)m+j} = A_{i,j}$. The standard basis is denoted by $\{\boldsymbol{e}_i\}_i$. For an integer n we denote $[n] = \{1, 2, \ldots, n\}$. For an $\mathbb{F}$-algebra $\mathcal{A}$ and a set of vectors $V = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\ell\} \subseteq \mathcal{A}^n$ we denote

$$\mathrm{Span}_{\mathcal{A}}(V) = \left\{ \sum_{i=1}^{\ell} \delta_i \boldsymbol{v}_i \,\middle|\, \delta_i \in \mathcal{A} \right\}.$$

We first prove

Lemma 5. *Let $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\nu\}$ be a set of (not necessarily linear) maps $\mathcal{A}^{2n} \to \mathbb{F}^{2n}$. Then $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\nu\}$ is a $(\mathcal{B}, \mathcal{A}, \mathbb{F})$-tester if and only if for every $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{A}^n$ we have*

$$\boldsymbol{a} \otimes \boldsymbol{b} \in \mathrm{Span}_{\mathcal{A}}\{\boldsymbol{l}_{i,1}(\boldsymbol{a}, \boldsymbol{b}) \otimes \boldsymbol{l}_{i,2}(\boldsymbol{a}, \boldsymbol{b})\}_{i=1}^{\nu} \tag{3}$$

where $\boldsymbol{l}_{i,1}(\boldsymbol{a}, \boldsymbol{b})$ (resp. $\boldsymbol{l}_{i,2}(\boldsymbol{a}, \boldsymbol{b})$) is the vector of length n that contains the first (resp. last) n entries of $\boldsymbol{l}_i(\boldsymbol{a}, \boldsymbol{b})$.

Proof. By the definition of tester, L is a $(\mathcal{B}, \mathcal{A}, \mathbb{F})$-tester if for every $A \in \mathbb{F}^{n \times n}$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{A}^n$

$$(\forall i)\ \boldsymbol{l}_{i,1}(\boldsymbol{a}, \boldsymbol{b})^T A\, \boldsymbol{l}_{i,2}(\boldsymbol{a}, \boldsymbol{b}) = 0 \implies \boldsymbol{a}^T A \boldsymbol{b} = 0.$$

Suppose (3) is true. Then for any $A \in \mathbb{F}^{n\times n}$ we have

$$\begin{aligned} \boldsymbol{a}^T A\boldsymbol{b} &= (\boldsymbol{a}\otimes\boldsymbol{b})^T \cdot vec(A) \\ &\in \mathrm{Span}_{\mathcal{A}}\{(\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})\otimes \boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b}))^T \cdot vec(A)\}_{i=1}^{\nu} \\ &= \mathrm{Span}_{\mathcal{A}}\{\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})^T A\ \boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b})\}_{i=1}^{\nu}. \end{aligned}$$

Therefore there are $\gamma_i \in \mathcal{A}$, $i = 1,\ldots,\nu$, such that

$$\boldsymbol{a}^T A\boldsymbol{b} = \sum_{i=1}^{\nu} \gamma_i \cdot \boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})^T A\ \boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b}).$$

Now if $\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})^T A\ \boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b}) = 0$ for all i then $\boldsymbol{a}^T A\boldsymbol{b} = 0$. Therefore L is a $(\mathcal{B},\mathcal{A},\mathbb{F})$-tester.

Now suppose L is a $(\mathcal{B},\mathcal{A},\mathbb{F})$-tester and assume for the contrary that (3) is not true. That is, there are $\boldsymbol{a},\boldsymbol{b} \in \mathcal{A}^n$ such that for $M := \{\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b}) \otimes \boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b})\}_{i=1}^{\nu}$ we have

$$\boldsymbol{a}\otimes\boldsymbol{b} \notin \mathrm{Span}_{\mathcal{A}}(M).$$

Notice that $M \subseteq \mathbb{F}^{n^2}$. We may assume w.l.o.g that M is linearly independent (over $\mathbb{F}$). Otherwise, if, say, $\boldsymbol{l}_{\nu,1}(\boldsymbol{a},\boldsymbol{b}) \otimes \boldsymbol{l}_{\nu,2}(\boldsymbol{a},\boldsymbol{b})$ is linearly dependent on $\{\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})\otimes\boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b})\}_{i=1}^{\nu-1}$ then $\boldsymbol{l}_{\nu,1}(\boldsymbol{a},\boldsymbol{b})^T\ A\boldsymbol{l}_{\nu,2}(\boldsymbol{a},\boldsymbol{b})$ is linearly dependent on $\{\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})^T A\boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b})\}_{i=1}^{\nu-1}$ for every A and if $\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})^T A\boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b}) = 0$ for $i = 1,\ldots,\nu-1$ then $\boldsymbol{l}_{\nu,1}(\boldsymbol{a},\boldsymbol{b})^T A\ \boldsymbol{l}_{\nu,2}(\boldsymbol{a},\boldsymbol{b}) = 0$. Therefore $\boldsymbol{l}_{\nu,1}(\boldsymbol{a},\boldsymbol{b}) \otimes \boldsymbol{l}_{\nu,2}(\boldsymbol{a},\boldsymbol{b})$ can be eliminated from the tester.

Consider the set $E = \{\boldsymbol{e}_i \otimes \boldsymbol{e}_j\}_{i,j\in[n]}$. Let $E' \subset E$ be of minimal size such that $\boldsymbol{a}\otimes\boldsymbol{b} \in \mathrm{Span}_{\mathcal{A}}(M \cup E')$. Such E' exists since $\boldsymbol{a}\otimes\boldsymbol{b} \in \mathrm{Span}_{\mathcal{A}}(E) = \mathcal{A}^{n^2}$. Again as before, $M \cup E'$ is linearly independent and E' is not empty. Since $\boldsymbol{a}\otimes\boldsymbol{b} \in \mathrm{Span}_{\mathcal{A}}(M\cup E')$ and $\boldsymbol{a}\otimes\boldsymbol{b} \notin \mathrm{Span}_{\mathcal{A}}(M)$ we have

$$\boldsymbol{a}\otimes\boldsymbol{b} = \sum_{\boldsymbol{v}\in M\cup E'} \delta_{\boldsymbol{v}}\boldsymbol{v}$$

where $\delta_{\boldsymbol{v}} \in \mathcal{A}$ and $\delta_{\boldsymbol{v}_0} \neq 0$ for some $\boldsymbol{v}_0 = \boldsymbol{e}_{i_0}\otimes\boldsymbol{e}_{j_0} \in E'$

Consider a vector $\boldsymbol{c} \in \mathbb{F}^{n^2}$ such that $\boldsymbol{v}^T\boldsymbol{c} = 0$ for all $\boldsymbol{v} \in (M\cup E')\backslash\{\boldsymbol{v}_0\} \subseteq \mathbb{F}^{n^2}$ and $\boldsymbol{v}_0^T\boldsymbol{c} = 1$. Let $A \in \mathbb{F}^{n\times n}$ be a matrix such that $vec(A) = \boldsymbol{c}$. Then for every i

$$\begin{aligned} \boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})^T A\boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b}) &= (\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})\otimes\boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b}))^T vec(A) \\ &= (\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})\otimes\boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b}))^T\boldsymbol{c} = 0 \end{aligned}$$

and

$$\begin{aligned} \boldsymbol{a}^T A \boldsymbol{b} &= (\boldsymbol{a} \otimes \boldsymbol{b})^T vec(A) \\ &= \sum_{\boldsymbol{v} \in M \cup E'} \delta_{\boldsymbol{v}} \boldsymbol{v}^T \boldsymbol{c} \\ &= \delta_{\boldsymbol{v}_0} \neq 0. \end{aligned}$$

Which is a contradiction to the fact that L is a $(\mathcal{B}, \mathcal{A}, \mathbb{F})$-tester. □

Now the following result with Lemma 5 show that Tester for bilinear forms gives bilinear algorithm for the multiplication of two elements in the algebra of bilinear complexity that is equal to the tester size.

Lemma 6. *Let $n \geq \dim \mathcal{A}$. Let $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\nu\}$ be a set of maps $\mathcal{A}^{2n} \to \mathbb{F}^{2n}$. If for every $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{A}^n$ we have*

$$\boldsymbol{a} \otimes \boldsymbol{b} \in \mathrm{Span}_{\mathcal{A}}\{\boldsymbol{l}_{i,1}(\boldsymbol{a}, \boldsymbol{b}) \otimes \boldsymbol{l}_{i,2}(\boldsymbol{a}, \boldsymbol{b})\}_{i=1}^{\nu} \tag{4}$$

where $\boldsymbol{l}_{i,1}(\boldsymbol{a}, \boldsymbol{b})$ (resp. $\boldsymbol{l}_{i,2}(\boldsymbol{a}, \boldsymbol{b})$) is the first (resp. last) n entries of $\boldsymbol{l}_i(\boldsymbol{a}, \boldsymbol{b})$, then there are $\lambda_{i,j} \in \mathcal{A}^*$, $i = 1, \ldots, \nu$, $j = 1, 2$ and $\gamma_1, \gamma_2, \ldots, \gamma_\nu \in \mathcal{A}$ such that for every two elements a_1, a_2 in $\mathcal{A}$ we have

$$a_1 a_2 = \sum_{i=1}^{\nu} \gamma_i \lambda_{i,1}(a_1) \lambda_{i,2}(a_2).$$

That is, (4) *implies that there is a bilinear algorithm for the multiplication of two elements in $\mathcal{A}$ of bilinear complexity ν.*

Proof. Suppose (4) is true. Consider a basis $\omega_1, \omega_2, \ldots, \omega_t$ for $\mathcal{A}$ over $\mathbb{F}$ where $t = \dim \mathcal{A}$. Consider the vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t, 0, \ldots, 0)^T \in \mathcal{A}^n$. By (4),

$$\boldsymbol{\omega} \otimes \boldsymbol{\omega} \in \mathrm{Span}_{\mathcal{A}}\{\boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega}) \otimes \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega})\}_{i=1}^{\nu}.$$

Therefore there are $\gamma_1, \ldots, \gamma_\nu \in \mathcal{A}$ such that

$$\boldsymbol{\omega} \otimes \boldsymbol{\omega} = \sum_{i=1}^{\nu} \gamma_i \cdot \boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega}) \otimes \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega}).$$

Let $a_1 = a_{1,1}\omega_1 + \cdots + a_{1,t}\omega_t$ and $a_2 = a_{2,1}\omega_1 + \cdots + a_{2,t}\omega_t$ be two elements in $\mathcal{A}$ where $a_{i,j} \in \mathbb{F}$, $i = 1, 2$ and $j = 1, \ldots, t$. Let $\boldsymbol{a}_1 =$

$(a_{1,1}, a_{1,2}, \ldots, a_{1,t}, 0, \ldots, 0)^T$ and $\boldsymbol{a}_2 = (a_{2,1}, a_{2,2}, \ldots, a_{2,t}, 0, \ldots, 0)^T$ in $\mathbb{F}^n$. Then

$$\begin{aligned} a_1 a_2 &= (\boldsymbol{a}_1^T \boldsymbol{\omega})(\boldsymbol{a}_2^T \boldsymbol{\omega}) \\ &= (\boldsymbol{a}_1 \otimes \boldsymbol{a}_2)^T (\boldsymbol{\omega} \otimes \boldsymbol{\omega}) \\ &= \sum_{i=1}^{\nu} \gamma_i \cdot (\boldsymbol{a}_1 \otimes \boldsymbol{a}_2)^T (\boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega}) \otimes \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega})) \\ &= \sum_{i=1}^{\nu} \gamma_i \cdot (\boldsymbol{a}_1^T \boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega}))(\boldsymbol{a}_2^T \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega})) \\ &= \sum_{i=1}^{\nu} \gamma_i \lambda_{i,1}(a_1) \lambda_{i,2}(a_2) \end{aligned}$$

where $\lambda_{i,1}(a_1) = \boldsymbol{a}_1^T \boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega})$ and $\lambda_{i,2}(a_2) = \boldsymbol{a}_2^T \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega})$ and $\lambda_{i,1}, \lambda_{i,2} \in \mathcal{A}^*$. □

The following result shows that a bilinear algorithm for the multiplication of two elements in the algebra of bilinear complexity μ gives a Tester for bilinear forms of size μ

Lemma 7. *If there are $\lambda_{i,j} \in \mathcal{A}^*$, $i = 1, \ldots, \mu$, $j = 1, 2$ and $\gamma_1, \gamma_2, \ldots, \gamma_\mu \in \mathcal{A}$ such that for every two elements α_1, α_2 in $\mathcal{A}$ we have*

$$\alpha_1 \alpha_2 = \sum_{i=1}^{\mu} \gamma_i \lambda_{i,1}(\alpha_1) \lambda_{i,2}(\alpha_2)$$

then the set of maps $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\mu\}$ where

$$\boldsymbol{l}_i(\boldsymbol{a}, \boldsymbol{b}) = (\lambda_{i,1}(a_1), \ldots, \lambda_{i,1}(a_n), \lambda_{i,2}(b_1), \ldots, \lambda_{i,2}(b_n))$$

$i = 1, \ldots, \mu$ is a $(\mathcal{B}, \mathcal{A}, \mathbb{F})$-tester.

Proof. Let $\boldsymbol{l}_{i,1}(\boldsymbol{a}, \boldsymbol{b})$ and $\boldsymbol{l}_{i,2}(\boldsymbol{a}, \boldsymbol{b})$ be the first and last n entries of $\boldsymbol{l}_i(\boldsymbol{a}, \boldsymbol{b})$,

respectively. Let $A \in \mathbb{F}^{n\times n}$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{A}^n$. Then

$$\begin{aligned}
\boldsymbol{a}^T A \boldsymbol{b} &= \sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j} a_i b_j \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j} \sum_{k=1}^{\mu} \gamma_k \lambda_{k,1}(a_i)\lambda_{k,2}(b_j) \\
&= \sum_{k=1}^{\mu} \gamma_k \sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j} \lambda_{k,1}(a_i)\lambda_{k,2}(b_j) \\
&= \sum_{k=1}^{\mu} \gamma_k \cdot \boldsymbol{l}_{k,1}(\boldsymbol{a},\boldsymbol{b})^T A \boldsymbol{l}_{k,2}(\boldsymbol{a},\boldsymbol{b}).
\end{aligned}$$

Therefore if $\boldsymbol{l}_{i,1}(\boldsymbol{a},\boldsymbol{b})^T A \boldsymbol{l}_{i,2}(\boldsymbol{a},\boldsymbol{b}) = 0$ for all $i = 1, \ldots, \mu$ then $\boldsymbol{a}^T A \boldsymbol{b} = 0$ and therefore L is a $(\mathcal{B}, \mathcal{A}, \mathbb{F})$-tester. □

We are now ready to prove Theorem 1.

Proof. We now prove 1 in Theorem 1 for any d. By Lemma 5 and 6 we have $\nu_{\mathbb{F}}^{\mathcal{HLCF}}(d, \mathcal{A}) \geq \mu_{\mathbb{F}}(d, \mathcal{A})$. By Lemma 7 we have $\nu_{\mathbb{F}}^{\mathcal{HLCF}}(d, \mathcal{A}) \leq \mu_{\mathbb{F}}(d, \mathcal{A})$. Therefore the result follows.

We now prove 2 in Theorem 1 for $d = 2$. The proof for any constant d is a simple extension of this proof. The proof for any d is given in the next subsection.

Given a $(\mathcal{B}, \mathcal{A}, \mathbb{F})$-tester $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\nu\}$ and a basis $\omega_1, \ldots, \omega_t$ for $\mathcal{A}$ over $\mathbb{F}$. Let $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t, 0, \ldots, 0)^T \in \mathcal{A}^n$. We compute $\boldsymbol{l}_i(\boldsymbol{\omega}, \boldsymbol{\omega})$ for every i. By Lemma 6 there are $\gamma_i \in \mathcal{A}$ such that

$$\boldsymbol{\omega} \otimes \boldsymbol{\omega} = \sum_{i=1}^{\nu} \gamma_i \cdot \boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega}) \otimes \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega}).$$

To find γ_i we write $\gamma_i = \gamma_{i,1}\omega_1 + \cdots + \gamma_{i,t}\omega_t$ where $\gamma_{i,j} \in \mathbb{F}$. Then write $\boldsymbol{\omega} \otimes \boldsymbol{\omega} = \boldsymbol{u}_1\omega_1 + \ldots + \boldsymbol{u}_t\omega_t$ where $\boldsymbol{u}_k \in \mathbb{F}^{n^2}$. Then for each $j = 1, 2, \ldots, t$ solve the following system of linear equation over $\mathbb{F}$

$$\boldsymbol{u}_j = \sum_{i=1}^{\nu} \gamma_{i,j} \cdot \boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega}) \otimes \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega})$$

to find $\gamma_{i,j}$. Now as in Lemma 6, for two elements $a_1 = a_{1,1}\omega_1 + \cdots + a_{1,t}\omega_t$, $a_2 = a_{2,1}\omega_1 + \cdots + a_{2,t}\omega_t$ in $\mathcal{A}$ where $a_{i,j} \in \mathbb{F}$, $i = 1, 2$ and $j = 1, \ldots, t$,

$\boldsymbol{a}_1 = (a_{1,1}, a_{1,2}, \ldots, a_{1,t}, 0, \ldots, 0)^T$ and $\boldsymbol{a}_2 = (a_{2,1}, a_{2,2}, \ldots, a_{2,t}, 0, \ldots, 0)^T$ in $\mathbb{F}^n$ we have

$$\begin{aligned} a_1 a_2 &= \sum_{i=1}^{\nu} \gamma_i \cdot (\boldsymbol{a}_1^T \boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega}))(\boldsymbol{a}_2^T \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega})) \\ &= \sum_{i=1}^{\nu} \gamma_i \lambda_{i,1}(a_1) \lambda_{i,2}(a_2) \end{aligned}$$

where $\lambda_{i,1}(a_1) = \boldsymbol{a}_1^T \boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \boldsymbol{\omega})$ and $\lambda_{i,2}(a_2) = \boldsymbol{a}_2^T \boldsymbol{l}_{i,2}(\boldsymbol{\omega}, \boldsymbol{\omega})$ and $\lambda_{i,1}, \lambda_{i,2} \in \mathcal{A}^*$.

Now the proof of 3 in Theorem 1 for any $d = 2$ follows from Lemma 7. The proof for any d is a simple extension of this proof.

The proof of 4 in Theorem 1 also an immediate consequence of Lemma 5, 6 and 7. □

Notice that the algorithm in the above proof of 2 in Theorem 1 for any d has time complexity $O(n^d)$. This is because finding γ_i in the equation

$$\boldsymbol{\omega} \otimes \overset{d}{\cdots} \otimes \boldsymbol{\omega} = \sum_{i=1}^{\nu} \gamma_i \cdot \boldsymbol{l}_{i,1}(\boldsymbol{\omega}, \overset{d}{\cdots}, \boldsymbol{\omega}) \otimes \cdots \otimes \boldsymbol{l}_{i,d}(\boldsymbol{\omega}, \overset{d}{\cdots}, \boldsymbol{\omega})$$

requires accessing vector of length n^d. Therefore the above construction is polynomial time only when d is constant. In the next subsection we give a deterministic polynomial time algorithm in n and d that solves this problem.

4.1 Polynomial Time Construction for any d

In this section we prove 2 in Theorem 1 for any d.

Given a $(\mathcal{HLF}(\mathbb{F}, n, d), \mathcal{A}, \mathbb{F})$-tester $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\nu\}$ and a basis $\omega_1, \ldots, \omega_t$ for $\mathcal{A}$ over $\mathbb{F}$. Since that $\mathcal{HLF}(\mathbb{F}, n, d)$ is the set of all multilinear function over $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_d)$ where $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,n})$, $i = 1, \ldots, d$ we have that $\boldsymbol{l}_i(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_d) : \mathcal{A}^{dn} \to \mathbb{F}^{dn}$, $i = 1, \ldots, \nu$.

Let $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t, 0, \ldots, 0)^T \in \mathcal{A}^n$. The algorithm first computes $\boldsymbol{l}_i(\boldsymbol{\omega}, \overset{d}{\ldots}, \boldsymbol{\omega})$ for every i. By Lemma 6 there are $\gamma_i \in \mathcal{A}$, $i = 1, \ldots, \nu$ such that for any d elements $a_j = a_{j,1}\omega_1 + \cdots + a_{j,t}\omega_t$, $j = 1, \ldots, d$ in $\mathcal{A}$ where $a_{j,i} \in \mathbb{F}$ for $j = 1, \ldots, d$ and $i = 1, \ldots, t$ and $\boldsymbol{a}_j = (a_{j,1}, a_{j,2}, \ldots, a_{j,t}, 0, \ldots, 0)^T \in \mathbb{F}^n$ we have

$$\begin{aligned} a_1 a_2 \cdots a_d &= \sum_{i=1}^{\nu} \gamma_i \cdot (\boldsymbol{a}_1^T \boldsymbol{l}_{i,1})(\boldsymbol{a}_2^T \boldsymbol{l}_{i,2}) \cdots (\boldsymbol{a}_d^T \boldsymbol{l}_{i,d}) \\ &= \sum_{i=1}^{\nu} \gamma_i \lambda_{i,1}(a_1) \lambda_{i,2}(a_2) \cdots \lambda_{i,d}(a_d) \end{aligned} \tag{5}$$

where $\boldsymbol{l}_{i,j}$ is the n-vector that contains the entries $(j-1)n+1, (j-1)n+2, \ldots, jn$ of $\boldsymbol{l}_i(\boldsymbol{\omega}, \overset{d}{.}, \boldsymbol{\omega})$ and $\lambda_{i,j}(a_1) = \boldsymbol{a}_j^T \boldsymbol{l}_{i,j}$, $j = 1, \ldots, d$ and $i = 1, \ldots, \nu$.

Now $\lambda_{i,j}$ for $j = 1, \ldots, d$ and $i = 1, \ldots, \nu$ can be found in polynomial time. The problem now is to find $\gamma_1, \ldots, \gamma_\nu$ that satisfies (5) in polynomial time. In what follows we give an algorithm that solves this problem.

Define the linear space

$$\boldsymbol{\Lambda} = \text{Span}_{\mathbb{F}}\{\Lambda_i \mid \Lambda_i : \mathcal{A}^d \to \mathbb{F}, \quad \Lambda_i(a_1, \ldots, a_d) := \lambda_{i,1}(a_1) \cdots \lambda_{i,d}(a_d)\}.$$

Every element $\Lambda' \in \boldsymbol{\Lambda}$ will be represented in the algorithm as $\Lambda' = \sum_{i=1}^{\nu} \beta_i \Lambda_i$ where $\beta_i \in \mathbb{F}$.

Now, the goal of the algorithm is to find in polynomial time a basis $\{\Lambda^{(1)}, \ldots, \Lambda^{(\nu')}\}$ for $\boldsymbol{\Lambda}$ and $\delta_1, \ldots, \delta_{\nu'} \in \mathcal{A}$ such that for every $a_1, \ldots, a_d \in \mathcal{A}$ we have

$$a_1 a_2 \ldots a_d = \sum_{i=1}^{\nu'} \delta_i \Lambda^{(i)}(a_1, \ldots, a_d). \tag{6}$$

Obviously, the algorithm can then finds γ_i that satisfies (5) in polynomial time.

We now show that if (6) holds and the algorithm knows all $\Lambda^{(i)}$ and $\delta_1, \ldots, \delta_w$ but not $\delta_{w+1}, \ldots, \delta_{\nu'}$ then it can replace $\Lambda^{(w+2)}, \ldots, \Lambda^{(\nu')}$ with $\Lambda^{'(w+2)}, \ldots, \Lambda^{'(\nu')}$ such that

$$\{\Lambda^{(1)}, \ldots, \Lambda^{(w+1)}, \Lambda^{'(w+2)}, \ldots, \Lambda^{'(\nu')}\} \tag{7}$$

is a basis for Λ and finds δ'_{w+1} such that for every $a_1, \ldots, a_d \in \mathcal{A}$

$$\begin{aligned} a_1 a_2 \ldots a_d &= \sum_{i=1}^{w} \delta_i \Lambda^{(i)}(a_1, \ldots, a_d) + \delta'_{w+1} \Lambda^{(w+1)}(a_1, \ldots, a_d) \\ &\quad + \sum_{i=w+2}^{\nu'} \delta_i \Lambda^{'(i)}(a_1, \ldots, a_d). \end{aligned} \tag{8}$$

Then repeating this procedure solves the problem.

Given (6). In the next Lemma we show how to find in polynomial time elements $b_1, \ldots, b_d \in \mathcal{A}$ such that $\Lambda^{(w+1)}(b_1, \ldots, b_d) = 1$. If no such $b_1, \ldots, b_d$ exist then the algorithm knows that $\Lambda^{(w+1)} \equiv 0$ and then it can choose $\delta'_{w+1} = 0$. Let $\Lambda^{(w+1)}(b_1, \ldots, b_d) = 1$ and $\Lambda^{(i)}(b_1, \ldots, b_d) = \alpha_i \in \mathbb{F}$ and consider the new basis in (7) where

$$\Lambda^{'(i)} = \Lambda^{(i)} - \alpha_i \Lambda^{(w+1)} \tag{9}$$

for $i = w+2, \ldots, \nu'$. Notice that for $i = w+2, \ldots, \nu'$

$$\Lambda'^{(i)}(b_1, \ldots, b_d) = \Lambda^{(i)}(b_1, \ldots, b_d) - \alpha_i \Lambda^{(w+1)}(b_1, \ldots, b_d) = 0.$$

By (9) and (6) we have (8) for some $\delta'_{w+1} \in \mathcal{A}$. Now notice that if we substitute $b_1, \ldots, b_d$ in (8) and get

$$\begin{aligned} b_1 b_2 \ldots b_d &= \sum_{i=1}^{w} \delta_i \Lambda^{(i)}(b_1, \ldots, b_d) + \delta'_{w+1} \Lambda^{(w+1)}(b_1, \ldots, b_d) \\ &\quad + \sum_{i=w+2}^{\nu'} \delta_i \Lambda'^{(i)}(b_1, \ldots, b_d) \\ &= \sum_{i=1}^{w} \delta_i \Lambda^{(i)}(b_1, \ldots, b_d) + \delta'_{w+1} \end{aligned}$$

and therefore

$$\delta'_{w+1} = b_1 b_2 \ldots b_d - \sum_{i=1}^{w} \delta_i \Lambda^{(i)}(b_1, \ldots, b_d).$$

can be computed by the algorithm in polynomials time.

It remains to prove

Lemma 8. *There is a polynomial time algorithm that for every*

$$\Lambda' \in \mathrm{Span}_{\mathbb{F}}\{\Lambda_i \mid \Lambda_i : \mathcal{A}^d \to \mathbb{F}, \quad \Lambda_i(a_1, \ldots, a_d) = \lambda_{i,1}(a_1) \cdots \lambda_{i,d}(a_d)\},$$

$\Lambda' \not\equiv 0$ *finds* $b_1, \ldots, b_d \in \mathcal{A}$ *such that* $\Lambda'(b_1, \ldots, b_d) = 1$.

Proof. We will find $b_1, \ldots, b_d \in \mathcal{A}$ such that $\lambda := \Lambda'(b_1, \ldots, b_d) \neq 0$. Then $\Lambda'(\lambda^{-1} b_1, b_2, \ldots, b_d) = 1$.

So our goal is to find $b_1, \ldots, b_d \in \mathcal{A}$ such that $\Lambda'(b_1, \ldots, b_d) \neq 0$. Let

$$\Lambda' = \sum_{i=1}^{\nu} \beta_i \Lambda_i$$

where $\beta_i \in \mathbb{F}$, $i = 1, \ldots, \nu$. Let $\omega_1, \ldots, \omega_t$ be any basis for $\mathcal{A}$ over $\mathbb{F}$.

The algorithm runs in stages. At stage ℓ the algorithm saves a set of ν functions $\Lambda^{(\ell,j)} = \sum_{i=1}^{\nu} \beta_{j,i} \Lambda_{\ell,i}$, $j = 1, \ldots, \nu$ where $\Lambda_{\ell,i}(a_{\ell+1}, \ldots, a_d) = \lambda_{i,\ell+1}(a_{\ell+1}) \cdots \lambda_{i,d}(a_d)$, $i = 1, \ldots, \nu$ and $\beta_{j,i} \in \mathbb{F}$. For each $\Lambda^{(\ell,j)}$ it also saves $\Omega_{\ell,j} = (\omega_{\ell,j,1}, \ldots, \omega_{\ell,j,\ell}) \in \{\omega_1, \ldots, \omega_t\}^{\ell}$ such that

1. $\Lambda^{(\ell,j)}(a_{\ell+1},\ldots,a_d) = \Lambda'(\Omega_{\ell,j}, a_{\ell+1},\ldots,a_d)$.

2. There is j and $b_{\ell+1},\ldots,b_d \in \mathcal{A}$ such that

$$\Lambda'(\Omega_{\ell,j}, b_{\ell+1},\ldots,b_d) = \Lambda^{(\ell,j)}(b_{\ell+1},\ldots,b_d) \neq 0.$$

For $\ell = 0$ we have $\Lambda^{(0,j)} = \Lambda'$ and $\Omega_{0,j} = ()$ for all j. So (1.) and (2.) are true for $\ell = 0$.

We now show how the algorithm runs in stage $\ell+1$ and in polynomial time generates $\Lambda^{(\ell+1,j)}$ and $\Omega_{\ell+1,j}$, $j = 1,\ldots,\nu$, that satisfy conditions (1.) and (2.). Notice that at stage $\ell = d$, by (2.), we have $\Lambda'(\Omega_{d,j}) \neq 0$. This achieves our goal.

For any $a_{\ell+1} := a_{\ell+1,1}\omega_1 + \cdots + a_{\ell+1,t}\omega_t$, $a_{\ell+1,j} \in \mathbb{F}$ and

$$\Lambda_{\ell+1,i}(a_{\ell+2},\ldots,a_d) := \lambda_{i,\ell+2}(a_{\ell+2})\cdots\lambda_{i,d}(a_d),$$

$i = 1,\ldots,\nu$, we have

$$\begin{aligned}
\Lambda'(\Omega_{\ell,j}, a_{\ell+1},\ldots,a_d) &= \Lambda^{(\ell,j)}(a_{\ell+1},\ldots,a_d) \\
&= \sum_{i=1}^{\nu} \beta_{j,i}\Lambda_{\ell,i}(a_{\ell+1},\ldots,a_d) \\
&= \sum_{i=1}^{\nu} \beta_{j,i} \sum_{r=1}^{t} a_{\ell+1,r}\Lambda_{\ell,i}(\omega_r, a_{\ell+2},\ldots,a_d) \\
&= \sum_{r=1}^{t} a_{\ell+1,r} \sum_{i=1}^{\nu} (\beta_{j,i}\lambda_{i,\ell+1}(\omega_r))\Lambda_{\ell+1,i}(a_{\ell+2},\ldots,a_d)
\end{aligned}$$

for all $j = 1,\ldots,\nu$. Consider

$$\Lambda^{(\ell+1,j,r)} = \sum_{i=1}^{\nu} (\beta_{j,i}\lambda_{i,\ell+1}(\omega_r))\Lambda_{\ell+1,i}$$

for $r = 1,\ldots,t$ and $j = 1,2,\ldots,\nu$. Then

$$\Lambda'(\Omega_{\ell,j}, a_{\ell+1},\ldots,a_d) = \sum_{r=1}^{t} a_{\ell+1,r}\Lambda^{(\ell+1,j,r)}(a_{\ell+2},\ldots,a_d).$$

Notice that

$$\Lambda'(\Omega_{\ell,j}, \omega_r, a_{\ell+2},\ldots,a_d) = \Lambda^{(\ell+1,j,r)}(a_{\ell+2},\ldots,a_d).$$

Obviously, if $\Lambda'(\Omega_{\ell,j}, b_{\ell+1}, \ldots, b_d) \neq 0$ for some $j = 1, \ldots, \nu$ then $\Lambda^{(\ell+1,j,r)}$ $(b_{\ell+2}, \ldots, b_d) \neq 0$ for some $r = 1, \ldots, t$ and $j = 1, \ldots, \nu$. We now show how to construct a set L of at most ν functions from $\Lambda^{(\ell+1,j,r)}$ that satisfies: if $\Lambda'(\Omega_{\ell,j}, b_{\ell+1}, \ldots, b_d) \neq 0$ for some $j = 1, \ldots, \nu$ then $\Lambda^{(\ell+1,j,r)}$ $(b_{\ell+2}, \ldots, b_d) \neq 0$ for some $\Lambda^{(\ell+1,j,r)} \in L$.

The algorithm starts with L that contains all $\Lambda^{(\ell+1,j,r)}$ and a set V of vectors that contains

$$v_{j,r} = (\beta_{j,1}\lambda_{1,\ell+1}(\omega_r), \ldots, \beta_{j,\nu}\lambda_{\nu,\ell+1}(\omega_r)) \in \mathbb{F}^\nu$$

for $j = 1, \ldots, \nu$ and $r = 1, \ldots, t$. If one of the vectors, say v_{j_0,r_0}, is linearly dependent on the other then if $\Lambda^{(\ell+1,j_0,r_0)}(b_{\ell+2}, \ldots, b_d) \neq 0$ for some $b_{\ell+2}, \ldots, b_d$ then $\Lambda^{(\ell+1,j,r)}(b_{\ell+2}, \ldots, b_d) \neq 0$ for some $(j, r) \neq (j_0, r_0)$. Therefore the algorithm can remove all the dependent vectors in V and their corresponding functions from L. Notice that since the dimension of $\mathrm{Span}(V)$ over $\mathbb{F}$ is at most ν the number of vectors that remain in V and functions that remain in L is at most ν. Let

$$L = \left\{\Lambda^{(\ell+1,1,r_1)}, \ldots, \Lambda^{(\ell+1,\nu,r_\nu)}\right\} \subset \{\Lambda^{(\ell+1,j,r)} \mid j = 1, \ldots, \nu, r = 1, \ldots, t\}$$

be the function that remains in L.

Denote $\Lambda^{(\ell+1,j)} = \Lambda^{(\ell+1,j,r_j)}$ and $\Omega_{\ell+1,j} = (\Omega_{\ell,j}, \omega_{r_j})$, $j = 1, \ldots, \nu$. Then

$$\begin{aligned}
\Lambda^{(\ell+1,j)}(a_{\ell+1}, \ldots, a_d) &= \Lambda^{(\ell+1,j,r_j)}(a_{\ell+1}, \ldots, a_d) \\
&= \Lambda'(\Omega_{\ell,j}, \omega_{r_j}, a_{\ell+2}, \ldots, a_d) \\
&= \Lambda'(\Omega_{\ell+1,j}, a_{\ell+2}, \ldots, a_d)
\end{aligned}$$

and (1.) follows for $\ell + 1$.

By the above argument if $\Lambda'(\Omega_{\ell,j}, b_{\ell+1}, \ldots, b_d) \neq 0$ for some $j = 1, \ldots, \nu$ then

$$\Lambda^{(\ell+1,j)}(b_{\ell+2}, \ldots, b_d) = \Lambda^{(\ell+1,j,r_j)}(b_{\ell+2}, \ldots, b_d) \neq 0$$

for some $j = 1, \ldots, \nu$. Since by (2.) there is $j = 1, \ldots, \nu$ and $b_{\ell+1}, \ldots, b_d \in \mathcal{A}$ such that

$$\Lambda'(\Omega_{\ell,j}, b_{\ell+1}, \ldots, b_d) = \Lambda^{(\ell,j)}(b_{\ell+1}, \ldots, b_d) \neq 0$$

there is $j = 1, \ldots, \nu$ and $b_{\ell+2}, \ldots, b_d$ such that $\Lambda^{(\ell+1,j)}(b_{\ell+2}, \ldots, b_d) \neq 0$. Then

$$\Lambda'(\Omega_{\ell+1,j}, b_{\ell+2}, \ldots, b_d) = \Lambda^{(\ell+1,j)}(b_{\ell+2}, \ldots, b_d) \neq 0.$$

This implies (2.) for $\ell + 1$. □

5 Symmetric Multilinear Complexity

In this section we prove the following

Theorem 2. *Let $\mathcal{A}$ be a commutative $\mathbb{F}$-algebra and $\mathcal{M} = \mathcal{HP}(\mathbb{F}, n, d)$ be a class of homogeneous multivariate polynomials of degree d. Then*

1. $\nu_{\mathbb{F}}^{\mathcal{HP}}(d, \mathcal{A}) = \mu_{\mathbb{F}}^{s}(d, \mathcal{A})$.
2. *Given a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν one can in polynomial time construct a symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν.*
3. *Given a symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν.*
4. *Given a set of polynomial time computable maps L that is $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν one can in polynomial time construct a linear symmetric $(\mathcal{M}, \mathcal{A}, \mathbb{F})$-tester of size ν.*

Again here the proof is for homogeneous multivariate polynomials of degree 2 (which are also called quadratic forms). The generalization to homogeneous multivariate polynomials of any degree d is straightforward.

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$ be a (column) vector of distinct indeterminates. Let $\mathcal{Q} = \mathcal{P}(\mathbb{F}, n, 2) = \{\boldsymbol{x}^T A \boldsymbol{x} \mid A \in \mathbb{F}^{n \times n}\} \subset \mathbb{F}[\boldsymbol{x}]$ be the set of quadratic forms.

Theorem 2 follows from the following three Lemmas

Lemma 9. *Let $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\nu\}$ be a set of maps $\mathcal{A}^n \to \mathbb{F}^n$. The set L is a $(\mathcal{Q}, \mathcal{A}, \mathbb{F})$-tester if and only if for every $\boldsymbol{a} \in \mathcal{A}^n$ we have*

$$\boldsymbol{a} \otimes \boldsymbol{a} \in \mathrm{Span}_{\mathcal{A}}\{\boldsymbol{l}_i(\boldsymbol{a}) \otimes \boldsymbol{l}_i(\boldsymbol{a})\}_{i=1}^{\nu}. \tag{10}$$

Proof. By the definition of tester, L is a $(\mathcal{Q}, \mathcal{A}, \mathbb{F})$-tester if for every $A \in \mathbb{F}^{n \times n}$ and $\boldsymbol{a} \in \mathcal{A}^n$

$$(\forall i)\ \boldsymbol{l}_i(\boldsymbol{a})^T A\ \boldsymbol{l}_i(\boldsymbol{a}) \quad \Longrightarrow \quad \boldsymbol{a}^T A \boldsymbol{a} = 0.$$

Suppose (10) is true. Then for any $A \in \mathbb{F}^{n \times n}$ we have

$$\begin{aligned} \boldsymbol{a}^T A \boldsymbol{a} &= (\boldsymbol{a} \otimes \boldsymbol{a})^T \cdot vec(A) \\ &\in \mathrm{Span}_{\mathcal{A}}\{(\boldsymbol{l}_i(\boldsymbol{a}) \otimes \boldsymbol{l}_i(\boldsymbol{a}))^T \cdot vec(A)\}_{i=1}^{\nu} \\ &= \mathrm{Span}_{\mathcal{A}}\{\boldsymbol{l}_i(\boldsymbol{a})^T A\ \boldsymbol{l}_i(\boldsymbol{a})\}_{i=1}^{\nu}. \end{aligned}$$

Therefore there are $\gamma_i \in \mathcal{A}$, $i = 1, \ldots, \nu$, such that

$$\boldsymbol{a}^T A \boldsymbol{a} = \sum_{i=1}^{\nu} \gamma_i \cdot \boldsymbol{l}_{i,1}(\boldsymbol{a})^T A \; \boldsymbol{l}_{i,2}(\boldsymbol{a}).$$

Now if $\boldsymbol{l}_i(\boldsymbol{a})^T A \; \boldsymbol{l}_i(\boldsymbol{a}) = 0$ for all i then $\boldsymbol{a}^T A \boldsymbol{a} = 0$. Therefore L is a $(\mathcal{Q}, \mathcal{A}, \mathbb{F})$-tester.

Now suppose L is a $(\mathcal{Q}, \mathcal{A}, \mathbb{F})$-tester and assume for the contrary that (10) is not true. Therefore there is $\boldsymbol{a} \in \mathcal{A}^n$ such that for $M := \{\boldsymbol{l}_i(\boldsymbol{a}) \otimes \boldsymbol{l}_i(\boldsymbol{a})\}_{i=1}^{\nu}$ $\boldsymbol{a} \otimes \boldsymbol{a} \notin M$. Notice that $M \subseteq \mathbb{F}^{n^2}$. As in the proof of Theorem 1, we may assume w.l.o.g that M is linearly independent over $\mathbb{F}$.

Since

$$\begin{aligned}
\boldsymbol{a} \otimes \boldsymbol{a} &= \left(\sum_{i=1}^{n} a_i \boldsymbol{e}_i\right) \otimes \left(\sum_{j=1}^{n} a_j \boldsymbol{e}_j\right) \\
&= \sum_{1 \le i < j \le n} a_i a_j \left((\boldsymbol{e}_i + \boldsymbol{e}_j) \otimes (\boldsymbol{e}_i + \boldsymbol{e}_j)\right) \\
&\quad + \sum a_i^2 (\boldsymbol{e}_i \otimes \boldsymbol{e}_i) \\
&\quad - \sum_{1 \le i < j \le n} a_i a_j ((\boldsymbol{e}_i \otimes \boldsymbol{e}_i) + (\boldsymbol{e}_j \otimes \boldsymbol{e}_j)),
\end{aligned}$$

we have $\boldsymbol{a} \otimes \boldsymbol{a} \in \mathrm{Span}_{\mathcal{A}}(E)$ where

$$E = \{\boldsymbol{e}_i \otimes \boldsymbol{e}_i\}_{i \in [n]} \cup \{(\boldsymbol{e}_i + \boldsymbol{e}_j) \otimes (\boldsymbol{e}_i + \boldsymbol{e}_j)\}_{i,j \in [n]}.$$

Let $E' \subset E$ be of minimal size such that $\boldsymbol{a} \otimes \boldsymbol{a} \in \mathrm{Span}_{\mathcal{A}}(M \cup E')$. Such E' exists since $\boldsymbol{a} \otimes \boldsymbol{a} \in \mathrm{Span}_{\mathcal{A}}(E)$. Again as before, $M \cup E'$ is linearly independent and E' is not empty. Since $\boldsymbol{a} \otimes \boldsymbol{a} \in \mathrm{Span}_{\mathcal{A}}(M \cup E')$ and $\boldsymbol{a} \otimes \boldsymbol{a} \notin \mathrm{Span}_{\mathcal{A}}(M)$ we have

$$\boldsymbol{a} \otimes \boldsymbol{a} = \sum_{\boldsymbol{v} \in M \cup E'} \delta_{\boldsymbol{v}} \boldsymbol{v}$$

where $\delta_{\boldsymbol{v}} \in \mathcal{A}$ and $\delta_{\boldsymbol{v}_0} \neq 0$ for some $\boldsymbol{v}_0 \in E'$

Consider a vector $\boldsymbol{c} \in \mathbb{F}^{n^2}$ such that $\boldsymbol{c}^T \boldsymbol{v}_0 = 1$ and $\boldsymbol{c}^T \boldsymbol{v} = 0$ for all $\boldsymbol{v} \in (M \cup E') \backslash \{\boldsymbol{v}_0\}$. Let $A \in \mathbb{F}^{n \times n}$ be the matrix such that $vec(A) = \boldsymbol{c}$. Then for every i

$$\begin{aligned}
\boldsymbol{l}_i(\boldsymbol{a})^T A \boldsymbol{l}_i(\boldsymbol{a}) &= (\boldsymbol{l}_i(\boldsymbol{a}) \otimes \boldsymbol{l}_i(\boldsymbol{a}))^T vec(A) \\
&= (\boldsymbol{l}_i(\boldsymbol{a}) \otimes \boldsymbol{l}_i(\boldsymbol{a}))^T \boldsymbol{c} = 0
\end{aligned}$$

and

$$\begin{aligned} \boldsymbol{a}^T A \boldsymbol{a} &= (\boldsymbol{a} \otimes \boldsymbol{a})^T vec(A) \\ &= \sum_{\boldsymbol{v} \in M \cup E'} \delta_{\boldsymbol{v}} \boldsymbol{v}^T \boldsymbol{c} \\ &= \delta_{\boldsymbol{v}_0} \neq 0. \end{aligned}$$

Which is a contradiction to the fact that L is a $(\mathcal{Q}, \mathcal{A}, \mathbb{F})$-tester. □

Now the following result with Lemma 9 show that Tester for quadratic forms gives symmetric bilinear algorithm for the multiplication of two elements in the algebra of bilinear complexity that is equal to the tester size.

Lemma 10. *Let $n \geq \dim \mathcal{A}$. Let $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\nu\}$ be a set of maps $\mathcal{A}^n \to \mathbb{F}^n$. If for every $\boldsymbol{a} \in \mathcal{A}^n$ we have*

$$\boldsymbol{a} \otimes \boldsymbol{a} \in \mathrm{Span}_{\mathcal{A}}\{\boldsymbol{l}_i(\boldsymbol{a}) \otimes \boldsymbol{l}_i(\boldsymbol{a})\}_{i=1}^{\nu} \tag{11}$$

then there are $\lambda_i \in \mathcal{A}^*$ and $\gamma_i \in \mathcal{A}$, $i = 1, \ldots, \nu$ such that for every two elements a_1, a_2 in $\mathcal{A}$ we have

$$a_1 a_2 = \sum_{i=1}^{\nu} \gamma_i \lambda_i(a_1) \lambda_i(a_2).$$

That is, (11) implies that there is a symmetric bilinear algorithm for the multiplication of two elements in $\mathcal{A}$ of bilinear complexity ν.

Proof. Suppose (11) is true. Consider a basis $\omega_1, \omega_2, \ldots, \omega_t$ for $\mathcal{A}$ over $\mathbb{F}$. Consider the vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t, 0, \ldots, 0)^T \in \mathcal{A}^n$. Then

$$\boldsymbol{\omega} \otimes \boldsymbol{\omega} \in \mathrm{Span}_{\mathcal{A}}\{\boldsymbol{l}_i(\boldsymbol{\omega}) \otimes \boldsymbol{l}_i(\boldsymbol{\omega})\}_{i=1}^{\nu}.$$

Therefore there are $\gamma_1, \ldots, \gamma_\nu \in \mathcal{A}$ such that

$$\boldsymbol{\omega} \otimes \boldsymbol{\omega} = \sum_{i=1}^{\nu} \gamma_i \cdot \boldsymbol{l}_i(\boldsymbol{\omega}) \otimes \boldsymbol{l}_i(\boldsymbol{\omega}).$$

Let $a_1 = a_{1,1}\omega_1 + \cdots + a_{1,t}\omega_t$ and $a_2 = a_{2,1}\omega_1 + \cdots + a_{2,t}\omega_t$ be two elements in $\mathcal{A}$ where $a_{i,j} \in \mathbb{F}$, $i = 1, 2$ and $j = 1, \ldots, t$. Let $\boldsymbol{a}_1 =$

$(a_{1,1}, a_{1,2}, \ldots, a_{1,t}, 0, \ldots, 0)^T$ and $\boldsymbol{a}_2 = (a_{2,1}, a_{2,2}, \ldots, a_{2,t}, 0, \ldots, 0)^T$ in $\mathbb{F}^n$. Then

$$\begin{aligned} a_1 a_2 &= (\boldsymbol{a}_1^T \boldsymbol{\omega})(\boldsymbol{a}_2^T \boldsymbol{\omega}) \\ &= (\boldsymbol{a}_1 \otimes \boldsymbol{a}_2)^T (\boldsymbol{\omega} \otimes \boldsymbol{\omega}) \\ &= \sum_{i=1}^{\nu} \gamma_i \cdot (\boldsymbol{a}_1 \otimes \boldsymbol{a}_2)^T (\boldsymbol{l}_i(\boldsymbol{\omega}) \otimes \boldsymbol{l}_i(\boldsymbol{\omega})) \\ &= \sum_{i=1}^{\nu} \gamma_i \cdot (\boldsymbol{a}_1^T \boldsymbol{l}_i(\boldsymbol{\omega}))(\boldsymbol{a}_2^T \boldsymbol{l}_i(\boldsymbol{\omega})) \\ &= \sum_{i=1}^{\nu} \gamma_i \lambda_i(a_1) \lambda_i(a_2) \end{aligned}$$

where $\lambda_i(a_1) = \boldsymbol{a}_1^T \boldsymbol{l}_i(\boldsymbol{\omega})$, $\lambda_i(a_2) = \boldsymbol{a}_2^T \boldsymbol{l}_i(\boldsymbol{\omega})$ and $\lambda_i \in \mathcal{A}^*$. □

The following result shows that symmetric bilinear algorithm for the multiplication of two elements in the algebra of bilinear complexity μ gives a Tester for quadratic forms of size μ

Lemma 11. *If there are $\lambda_i \in \mathcal{A}^*$ and $\gamma_i \in \mathcal{A}$, $i = 1, \ldots, \mu$, such that for every two elements α_1, α_2 in $\mathcal{A}$ we have*

$$\alpha_1 \alpha_2 = \sum_{i=1}^{\mu} \gamma_i \lambda_i(\alpha_1) \lambda_i(\alpha_2)$$

then the set of maps $L = \{\boldsymbol{l}_1, \ldots, \boldsymbol{l}_\mu\}$ where

$$\boldsymbol{l}_i(\boldsymbol{a}) = (\lambda_i(a_1), \ldots, \lambda_i(a_n))$$

$i = 1, \ldots, \mu$ is a $(\mathcal{Q}, \mathcal{A}, \mathbb{F})$-tester.

Proof. We have

$$\begin{aligned} \boldsymbol{a}^T A \boldsymbol{a} &= \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} a_i a_j \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} \sum_{k=1}^{\mu} \gamma_k \lambda_k(a_i) \lambda_k(a_j) \\ &= \sum_{k=1}^{\mu} \gamma_k \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} \lambda_k(a_i) \lambda_k(a_j) \\ &= \sum_{k=1}^{\mu} \gamma_k \cdot \boldsymbol{l}_k(\boldsymbol{a})^T A \boldsymbol{l}_k(\boldsymbol{a}). \end{aligned}$$

Therefore if $\boldsymbol{l}_k(\boldsymbol{a})^T A \boldsymbol{l}_k(\boldsymbol{a}) = 0$ for all $k = 1, \ldots, \mu$ then $\boldsymbol{a}^T A \boldsymbol{a} = 0$ and L is a $(\mathcal{Q}, \mathcal{A}, \mathbb{F})$-tester. □

In Section 7 we show that symmetric multilinear algorithms for the multiplication of d elements in an $|\mathbb{F}|$-algebra $\mathcal{A}$ exist if and only if either $|\mathbb{F}| \geq d$ or $|\mathbb{F}| < d$ and $a^{|\mathbb{F}|} = a$ for all $a \in \mathcal{A}$.

6 Simulators

In this section prove the results for simulators. We start with the proof of

Theorem 3. *we have the following*

1. *Given a reducible symmetric multilinear algorithm for the multiplication of d elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a polynomial time simulator for the class of polynomials of degree d and $\mathcal{A}$ of simulation complexity ν.*

   In particular,
   $$\sigma_{\mathbb{F}}^{\mathcal{P}}(d, \mathcal{A}) \leq \mu_{\mathbb{F}}^{rs}(d, \mathcal{A}).$$

2. *From any polynomial time simulator for the set of all multivariate polynomials of degree d and an $\mathbb{F}$-algebra $\mathcal{A}$ of simulation complexity ν one can construct in polynomial time a symmetric multilinear algorithm for the product of d elements in $\mathcal{A}$ of multilinear complexity ν.*

   In particular,
   $$\mu_{\mathbb{F}}^{s}(d, \mathcal{A}) \leq \sigma_{\mathbb{F}}^{\mathcal{P}}(d, \mathcal{A}).$$

Proof. We first prove *1.* Let

$$a_1 a_2 \cdots a_d = \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d} \lambda_i(a_j)$$

be a reducible symmetric multilinear algorithm for the multiplication of d elements in the $\mathbb{F}$-algebra $\mathcal{A}$. Then $\lambda_i(1_{\mathcal{A}}) = 1$ for all $i = 1, \ldots, \mu$.

Now for any $d' \leq d$ we have

$$\begin{aligned} a_1 a_2 \cdots a_{d'} &= a_1 a_2 \cdots a_{d'} \overbrace{1_{\mathcal{A}} \cdots 1_{\mathcal{A}}}^{d-d'} \\ &= \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d'} \lambda_i(a_j) \prod_{j=d'+1}^{d} \lambda_i(1_{\mathcal{A}}) \\ &= \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d'} \lambda_i(a_j). \end{aligned}$$

Therefore for any monomial M of degree at most d and any $\boldsymbol{a} \in \mathcal{A}^n$ we have

$$M(a_1, \ldots, a_n) = \sum_{i=1}^{\mu} \gamma_i M(\lambda_i(a_1), \ldots, \lambda_i(a_n)).$$

Let $f = \sum_{j=1}^{t} \beta_j M_j \in \mathbb{F}[x_1, \ldots, x_n]$ be any multivariate polynomial of degree at most d where each M_i is a monomial of degree at most d and $\beta_j \in \mathbb{F}$ for $j = 1, \ldots, t$. Then

$$\begin{aligned} f(a_1, \ldots, a_n) &= \sum_{j=1}^{t} \beta_j M_j(a_1, \ldots, a_n) \\ &= \sum_{j=1}^{t} \beta_j \sum_{i=1}^{\mu} \gamma_i M_j(\lambda_i(a_1), \ldots, \lambda_i(a_n)) \\ &= \sum_{i=1}^{\mu} \gamma_i \sum_{j=1}^{t} \beta_j M_j(\lambda_i(a_1), \ldots, \lambda_i(a_n)) \\ &= \sum_{i=1}^{\mu} \gamma_i f(\lambda_i(a_1), \ldots, \lambda_i(a_n)) \qquad (12) \end{aligned}$$

Now the simulator for multivariate polynomials of degree d and $\mathcal{A}$ runs as follows: For $\boldsymbol{a} \in \mathcal{A}^n$, it generates the assignments $\lambda_i(\boldsymbol{a}) := (\lambda_i(a_1), \ldots, \lambda_i(a_n)) \in \mathbb{F}^n$ and asks the assignment queries $\lambda_i(\boldsymbol{a})$ for $i = 1, 2, \ldots, \mu$. Then from $f(\lambda_i(\boldsymbol{a}))$ finds $f(\boldsymbol{a})$ with (12).

We now prove *2* in Theorem 3. Let S be a simulator algorithm for the set of all polynomials of degree d and $\mathcal{A}$. Suppose for $\boldsymbol{a} \in \mathcal{A}^n$ the algorithm generates $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\mu \in \mathbb{F}^n$. If $f(\boldsymbol{a}) \neq 0$ and $f(\boldsymbol{a}_i) = 0$ for all i then the algorithm cannot know the value of $f(\boldsymbol{a})$. This is because the zero function z also satisfies $z(\boldsymbol{a}_i) = 0$ for all i and $z(\boldsymbol{a}) = 0 \neq f(\boldsymbol{a})$. Therefore, if $f(\boldsymbol{a}) \neq 0$ then one of the values $f(\boldsymbol{a}_i)$ is not equal to zero. This shows that the simulator is a tester for the class of multivariate polynomials of degree d and $\mathcal{A}$. Now by Theorem 2 the result follows. □

We now prove

Theorem 4. *we have the following*

1. *Given a symmetric multilinear algorithm for the multiplication of $d+1$ elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a reducible symmetric multilinear algorithm*

for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν. In particular

$$\mu^{rs}_{\mathbb{F}}(d,\mathcal{A}) \leq \mu^{s}_{\mathbb{F}}(d+1,\mathcal{A})$$

and

$$1 \leq \frac{\sigma^{\mathcal{P}}_{\mathbb{F}}(d,\mathcal{A})}{\mu^{s}_{\mathbb{F}}(d,\mathcal{A})} \leq \frac{\mu^{s}_{\mathbb{F}}(d+1,\mathcal{A})}{\mu^{s}_{\mathbb{F}}(d,\mathcal{A})}.$$

2. *If $|\mathbb{F}| \geq d+1$ then given a symmetric multilinear algorithm for the multiplication of d elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity $(d+1)\nu$. In particular, if $|\mathbb{F}| \geq d+1$ then*

$$\mu^{rs}_{\mathbb{F}}(d,\mathcal{A}) \leq (d+1)\cdot\mu^{s}_{\mathbb{F}}(d,\mathcal{A}) - d\cdot\mu^{rs}_{\mathbb{F}}(d-1,\mathcal{A})$$

and

$$1 \leq \frac{\sigma^{\mathcal{P}}_{\mathbb{F}}(d,\mathcal{A})}{\mu^{s}_{\mathbb{F}}(d,\mathcal{A})} \leq d+1.$$

3. *If $|\mathbb{F}| = \infty$ (or large enough) then given a symmetric multilinear algorithm for the multiplication of d elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν one can in polynomial time construct a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ with multilinear complexity ν. In particular, if $|\mathbb{F}| = \infty$ then*

$$\sigma^{\mathcal{P}}_{\mathbb{F}}(d,\mathcal{A}) = \mu^{rs}_{\mathbb{F}}(d,\mathcal{A}) = \mu^{s}_{\mathbb{F}}(d,\mathcal{A}).$$

Proof. We first prove *1* in Theorem 4. Let

$$a_1a_2\cdots a_da_{d+1} = \sum_{i=1}^{\mu}\gamma_i\prod_{j=1}^{d+1}\lambda_i(a_j)$$

be a multilinear algorithm for the multiplication of $d+1$ elements in the $\mathbb{F}$-algebra $\mathcal{A}$. Suppose without less of generality $\lambda_i(1_{\mathcal{A}}) = 1$ for $i = 1,2,\ldots,\mu'$

and $\lambda_i(1_{\mathcal{A}}) = 0$ for $i = \mu' + 1, \ldots, \mu$. Then

$$\begin{aligned} a_1 a_2 \cdots a_d &= a_1 a_2 \cdots a_d 1_{\mathcal{A}} \\ &= \sum_{i=1}^{\mu} (\gamma_i \lambda_i(1_{\mathcal{A}})) \prod_{j=1}^{d} \lambda_i(a_j) \\ &= \sum_{i=1}^{\mu'} (\gamma_i \lambda_i(1_{\mathcal{A}})) \prod_{j=1}^{d} \lambda_i(a_j) \\ &= \sum_{i=1}^{\mu'} \gamma_i \prod_{j=1}^{d} \lambda_i(a_j). \end{aligned}$$

Since $\lambda_i(1_{\mathcal{A}}) = 1$ for $i = 1, 2, \ldots, \mu'$ the above is a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ of multilinear complexity μ. This completes the proof.

We now prove *2* in Theorem 4. Suppose $|\mathbb{F}| \geq d + 1$. Let

$$a_1 a_2 \cdots a_d = \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d} \lambda_i(a_j)$$

be a multilinear algorithm for the multiplication of d elements in the $\mathbb{F}$-algebra $\mathcal{A}$. Suppose without less of generality $\lambda_i(1_{\mathcal{A}}) = 1$ for $i = 1, 2, \ldots, \mu'$ and $\lambda_i(1_{\mathcal{A}}) = 0$ for $i = \mu' + 1, \ldots, \mu$. By the above argument

$$\mu' \geq \mu_{\mathbb{F}}^{rs}(d - 1, \mathcal{A}).$$

Consider any linear map $\lambda : \mathcal{A} \to \mathbb{F}$ such that $\lambda(1_{\mathcal{A}}) = 1$. Let $F = \{\beta_1, \ldots, \beta_{d+1}\} \subseteq \mathbb{F}$ be a set of size $d+1$. Consider $d+1$ variables $x, y_1, \ldots, y_d$ and the polynomial

$$P_r(x) = \prod_{i=1}^{d} (x \lambda_r(y_i) + \lambda(y_i))$$

where $r = \mu' + 1, \mu' + 2, \ldots, \mu$. Since $P_r(x)$ is a polynomial of degree d in x (with $d+1$ coefficient) the coefficient x^d of $P_r(x)$ (which is $\prod_{i=1}^{d} \lambda_r(y_i)$) can be computed by interpolation from $\{P_r(\beta)\}_{\beta \in F}$. Therefore there are $\alpha_j \in \mathbb{F}$, $j = 1, \ldots, d$ such that

$$\prod_{i=1}^{d} \lambda_r(y_i) = \sum_{j=1}^{d+1} \alpha_j \prod_{i=1}^{d} (\beta_j \lambda_r(y_i) + \lambda(y_i)) \tag{13}$$

for $r = \mu' + 1, \ldots, \mu$. Notice also that $\beta_j \lambda_r(1_{\mathcal{A}}) + \lambda(1_{\mathcal{A}}) = 1$.

We now replace the last $\mu - \mu'$ terms in the algorithm with the terms in (13) and get a reducible symmetric multilinear algorithm of complexity

$$\mu' + (d+1)(\mu - \mu') \leq (d+1) \cdot \mu^s_{\mathbb{F}}(d, \mathcal{A}) - d \cdot \mu^{rs}_{\mathbb{F}}(d-1, \mathcal{A}).$$

□

A slightly better bound can be obtain if we take

$$P_r(x) = \prod_{i=1}^{d} (x\lambda_r(y_i) + \lambda_s(y_i))$$

where $s \leq \mu'$. Then $\prod_{i=1}^{d} \lambda_r(y_i)$ and $\prod_{i=1}^{d} \lambda_r(y_i)$ can be computed from $\{P_r(\beta)\}_{\beta \in F}$ using interpolation. This gives the bound

$$\mu' + (d+1)(\mu - \mu') - \min(\mu', \mu - \mu') \leq$$

$$(d+1) \cdot \mu^s_{\mathbb{F}}(d, \mathcal{A}) - d \cdot \mu^{rs}_{\mathbb{F}}(d-1, \mathcal{A}) - \min(\mu^{rs}_{\mathbb{F}}(d-1, \mathcal{A}) \ , \ \mu^s_{\mathbb{F}}(d, \mathcal{A}) - \mu^{rs}_{\mathbb{F}}(d-1, \mathcal{A})).$$

We now prove *3* in Theorem 4. Let

$$a_1 a_2 \cdots a_d = \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d} \lambda_i(a_j)$$

be a symmetric multilinear algorithm for the multiplication of d elements in the $\mathbb{F}$-algebra $\mathcal{A}$. The idea of the proof is to find a unit (invertible) element $u \in \mathcal{A}$ such that $\lambda_i(u) \neq 0$ for all $i = 1, \ldots, \mu$. Then

$$\begin{aligned} a_1 a_2 \cdots a_d &= u^{-d}(ua_1)(ua_2)\cdots(ua_d) \\ &= \sum_{i=1}^{\mu} (u^{-d}\gamma_i) \prod_{j=1}^{d} \lambda_i(ua_j) \\ &= \sum_{i=1}^{\mu} \gamma'_i \prod_{j=1}^{d} \lambda'_i(a_j) \end{aligned}$$

where $\gamma'_i = (\lambda_i(u)^{-1}u)^{-d}\gamma_i$ and $\lambda'_i(x) = \lambda_i(ux)/\lambda_i(u)$. Since $\lambda'_i(1_{\mathcal{A}}) = \lambda_i(u1_{\mathcal{A}})/\lambda_i(u) = 1$ for all i the symmetric multilinear algorithm is reducible.

It remains to find a unit element $u \in \mathcal{A}$ such that $\lambda_i(u) \neq 0$ for all $i = 1, \ldots, \mu$. We first show how to find an element (not necessary unit) $v \in \mathcal{A}$ such that $\lambda_i(v) \neq 0$ for all i. Let $\omega_1, \ldots, \omega_t$ be a basis for $\mathcal{A}$. Since λ_i

is not the zero function there is j_i such that $\lambda_i(\omega_{j_i}) \neq 0$. Now suppose we have an element $\beta \in \mathcal{A}$ such that $\lambda_1(\beta) \neq 0, \ldots, \lambda_{r-1}(\beta) \neq 0$ and $\lambda_r(\beta) = 0$. Let $\delta \notin \Delta := \{-\lambda_i(\omega_{j_r})/\lambda_i(\beta) \mid i = 1, \ldots, r-1\}$. Consider the element $\beta' = \delta\beta + \omega_{j_r}$. For $i = 1, \ldots, r-1$ we have $\lambda_i(\beta') = \delta\lambda_i(\beta) + \lambda_i(\omega_{j_r}) \neq 0$ since $\delta \notin \Delta$. We also have $\lambda_r(\beta') = \delta\lambda_r(\beta) + \lambda_r(\omega_{j_r}) = \lambda_r(\omega_{j_r}) \neq 0$. Repeating the above gives an element $v \in \mathcal{A}$ such that $\lambda_i(v) \neq 0$ for all $i = 1, \ldots, \mu$.

Now consider the element $u = \delta v + 1_{\mathcal{A}}$ such that

$$\delta \notin \Delta' := \{-\lambda_i(1_{\mathcal{A}})/\lambda_i(v) \mid i = 1, \ldots, \mu\} \cup \{-1/\pi \mid \pi \text{ is an eigenvalue of } v\}.$$

Then $\lambda_i(u) = \lambda_i(\delta v + 1_{\mathcal{A}}) = \delta\lambda_i(v) + \lambda_i(1_{\mathcal{A}}) \neq 0$ and $u = \delta(v + (1/\delta)1_{\mathcal{A}})$ is unit because $-1/\delta$ is not an eigenvalue of v. This completes the proof. □

7 Classification

In this section we give a classification of all algebras that have symmetric multilinear algorithm, reducible symmetric multilinear algorithm and simulators.

We first prove

Theorem 5. *Let $\mathcal{A}$ be $\mathbb{F}$-algebra. There is a symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$ if and only if $\mathcal{A}$ is commutative and one of the following conditions is true*

1. $|\mathbb{F}| \geq d$
2. $|\mathbb{F}| < d$ *and for every element $a \in \mathcal{A}$, we have $a^{|\mathbb{F}|} = a$.*

Proof. ($\Rightarrow$). Let

$$a_1 a_2 \cdots a_d = \sum_{i=1}^{\mu} \gamma_i \prod_{j=1}^{d} \lambda_i(a_j) \tag{14}$$

be a symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$. By (14) we have $a_1 a_2 = a_1 a_2 1_{\mathcal{A}}^{d-2} = a_2 a_1 1_{\mathcal{A}}^{d-2} = a_2 a_1$ and therefore $\mathcal{A}$ is commutative algebra. Let $|\mathbb{F}| = q$. We now show that when $q < d$ then $a^q = a$ for all the elements $a \in \mathcal{A}$. Since $\lambda_i(1_{\mathcal{A}}) \in \{0, 1\}$, $\lambda_i(a) \in \mathbb{F}$,

$\lambda_i(a)^q = \lambda_i(a)$ and $q < d$ we have

$$\begin{aligned} a^q &= a^q 1_{\mathcal{A}}^{d-q} = \sum_{i=1}^{\mu} \gamma_i \lambda_i(a)^q \lambda_i(1_{\mathcal{A}})^{d-q} \\ &= \sum_{i=1}^{\mu} \gamma_i \lambda_i(a) \lambda_i(1_{\mathcal{A}})^{d-1} = a \cdot 1_{\mathcal{A}}^{d-1} = a. \end{aligned}$$

($\Leftarrow$). We now show that if $\mathcal{A}$ is commutative $\mathbb{F}$-algebra and one of the conditions is true then there is a symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$. We consider three cases

Case I. $|\mathbb{F}| \geq d+1$.

Let $\omega_1, \ldots, \omega_n$ be any basis for $\mathcal{A}$ over $\mathbb{F}$. Consider the multivariate polynomial

$$f = \prod_{i=1}^{d} (x_{i,1} y_1 + \cdots + x_{i,n} y_n)$$

where $x_{i,j}$ and y_j are distinct indeterminates. Let $F \subset \mathbb{F}$ be any set of size $d+1$. For $\boldsymbol{r} = (r_1, \ldots, r_n) \in [d]^n$, $r_1 + \cdots + r_n = d$, let $A_{\boldsymbol{r}} \in \mathbb{F}[\{x_{i,j}\}_{i,j}]$ be the coefficient of $y_1^{r_1} y_2^{r_2} \cdots y_n^{r_n}$ in f. By Lemma 12 in the next subsection we have

$$A_{\boldsymbol{r}} \in \mathrm{Span}_{\mathbb{F}} \left\{ \prod_{i=1}^{d} (x_{i,1} z_1 + \cdots + x_{i,n} z_n) \;\middle|\; \boldsymbol{z} \in F^n \right\}.$$

Therefore

$$\begin{aligned} \prod_{i=1}^{d} (x_{i,1}\omega_1 + \cdots + x_{i,n}\omega_n) &= \sum_{\boldsymbol{r} \in [d]^n} A_{\boldsymbol{r}} \omega_1^{r_1} \cdots \omega_n^{r_n} \\ &\in \mathrm{Span}_{\mathcal{A}} \left\{ \prod_{i=1}^{d} (x_{i,1} z_1 + \cdots + x_{i,n} z_n) \;\middle|\; \boldsymbol{z} \in F^n \right\}. \end{aligned}$$

Thus, for each $\boldsymbol{z} \in F^n$ there is $\gamma_{\boldsymbol{z}} \in \mathcal{A}$ such that

$$\prod_{i=1}^{d} (x_{i,1}\omega_1 + \cdots + x_{i,n}\omega_n) =$$

$$\sum_{\boldsymbol{z} \in F^n} \gamma_{\boldsymbol{z}} \prod_{i=1}^{d} (x_{i,1} z_1 + \cdots + x_{i,n} z_n). \tag{15}$$

Now for any d elements $a_i = a_{i,1}\omega_1 + \cdots + a_{i,n}\omega_n \in \mathcal{A}$, $i = 1, \ldots, d$ and $a_{i,j} \in \mathbb{F}$ substituting $x_{i,j} = a_{i,j}$ in (15) implies the result.

Case II. $|\mathbb{F}| = d$.

In this case we consider the function

$$g = \prod_{i=1}^{d}(x_{i,1}y_1 + \cdots + x_{i,n}y_n) - \sum_{k=1}^{n}\left(\prod_{i=1}^{d} x_{i,k}\right) y_k^d.$$

Now the degree of each variable y_i in g is at most $d-1$ and by Lemma 12, the interpolation of the coefficients of $y_1^{r_1}\cdots y_n^{r_n}$ in f where $r_i \leq d-1$ for all i is possible with the d elements of the field. We now proceed as in case I and get

$$\prod_{i=1}^{d}(x_{i,1}\omega_1 + \cdots + x_{i,n}\omega_n) - \sum_{k=1}^{n}\left(\prod_{i=1}^{d} x_{i,k}\right)\omega_k^d =$$

$$\sum_{\boldsymbol{z}\in\mathbb{F}^n}\gamma_{\boldsymbol{z}}\left(\prod_{i=1}^{d}(x_{i,1}z_1 + \cdots + x_{i,n}z_n) - \sum_{k=1}^{n}\left(\prod_{i=1}^{d} x_{i,k}\right) z_k^d\right).$$

Then

$$\prod_{i=1}^{d}(x_{i,1}\omega_1 + \cdots + x_{i,n}\omega_n) =$$

$$\sum_{\boldsymbol{z}\in\mathbb{F}^n}\gamma_{\boldsymbol{z}}\prod_{i=1}^{d}(x_{i,1}z_1 + \cdots + x_{i,n}z_n) - \sum_{k=1}^{n}\left(\left(\sum_{\boldsymbol{z}\in\mathbb{F}^n}\gamma_{\boldsymbol{z}}z_k^d\right) + \omega_k^d\right)\left(\prod_{i=1}^{d} x_{i,k}\right).$$

and as in Case I this gives a symmetric multilinear algorithm.

Case III. $q := |\mathbb{F}| < d$ and $a^q = a$ for all $a \in \mathcal{A}$.

For integers $r \geq 0$ and $q \geq 2$ we denote by $r \bmod_0(q-1)$ the integer $0 \leq r' \leq q-1$ such that $x^r \equiv x^{r'} \mod x^q - x$. For an integer vectors $\boldsymbol{r} = (r_1, \ldots, r_n)$ we denote $\boldsymbol{r} \bmod_0(q-1) = (r_1 \bmod_0(q-1), \ldots, r_n \bmod_0(q-1))$. This is equivalent to say that the monomial $y_1^{r_1}\cdots y_n^{r_n}$ is equal to the monomial $y_1^{r_1'}\cdots y_n^{r_n'}$ in $\mathbb{F}[\boldsymbol{y}]/(y_1^q - y_1, \ldots, y_n^q - y_n)$.

Let $\omega_1, \ldots, \omega_n$ be any basis for $\mathcal{A}$ over $\mathbb{F}$. Consider the multivariate polynomial

$$f = \prod_{i=1}^{d}(x_{i,1}y_1 + \cdots + x_{i,n}y_n)$$

in $F[\boldsymbol{x}][\boldsymbol{y}]$ where $x_{i,j}$ and y_j are distinct indeterminates. For $\boldsymbol{r} = (r_1, \ldots, r_n) \in \{0, 1, \ldots, q-1\}^n$ let $A_{\boldsymbol{r}} \in \mathbb{F}[\{x_{i,j}\}_{i,j}]$ be the sum of all the coefficients of

$y_1^{r_1'} y_2^{r_2'} \cdots y_n^{r_n'}$ in f where $\boldsymbol{r} = \boldsymbol{r}' \bmod_0 (q-1)$. By Lemma 13 in the next subsection we have

$$A_{\boldsymbol{r}} \in \mathrm{Span}_{\mathbb{F}} \left\{ \prod_{i=1}^{d} (x_{i,1} z_1 + \cdots + x_{i,n} z_n) \;\middle|\; \boldsymbol{z} \in \mathbb{F}^n \right\}.$$

Since $\omega_i^q = \omega_i$ in $\mathcal{A}$ we have

$$\prod_{i=1}^{d} (x_{i,1}\omega_1 + \cdots + x_{i,n}\omega_n) = \sum_{\boldsymbol{r} \in \{0,1,\ldots,q-1\}^n} A_{\boldsymbol{r}} \omega_1^{r_1} \cdots \omega_n^{r_n}$$

$$\in \mathrm{Span}_{\mathcal{A}} \left\{ \prod_{i=1}^{d} (x_{i,1} z_1 + \cdots + x_{i,n} z_n) \;\middle|\; \boldsymbol{z} \in \mathbb{F}^n \right\}.$$

Thus, there are $\gamma_{\boldsymbol{z}} \in \mathcal{A}$ for each $\boldsymbol{z} \in \mathbb{F}^n$ such that

$$\prod_{i=1}^{d} (x_{i,1}\omega_1 + \cdots + x_{i,n}\omega_n) =$$

$$\sum_{\boldsymbol{z} \in \mathbb{F}^n} \gamma_{\boldsymbol{z}} \prod_{i=1}^{d} (x_{i,1} z_1 + \cdots + x_{i,n} z_n). \tag{16}$$

Now for any d elements $a_i = a_{i,1}\omega_1 + \cdots + a_{i,n}\omega_n \in \mathcal{A}$, $i = 1, \ldots, d$ and $a_{i,j} \in \mathbb{F}$ substituting $x_{i,j} = a_{i,j}$ in (16) implies the result. □

The second result we prove in this section is

Theorem 6. *Let $\mathcal{A}$ be an $\mathbb{F}$-algebra. The following conditions are equivalent*

1. *There is a simulator algorithm for multivariate polynomials of degree d and $\mathcal{A}$.*
2. *$\mathcal{A}$ is commutative algebra and one of the following conditions is true*
   (a) *$|\mathbb{F}| \geq d + 1$*
   (b) *$|\mathbb{F}| < d + 1$ and for every element $a \in \mathcal{A}$ we have $a^{|\mathbb{F}|} = a$.*
3. *There is a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$.*

Proof. (1 $\Rightarrow$ 2). In Theorem 3 we showed that from a simulator for the set of all multivariate polynomials of degree d and an $\mathbb{F}$-algebra $\mathcal{A}$ of simulation complexity ν one can construct a symmetric multilinear algorithm for the product of d elements in $\mathcal{A}$ of multilinear complexity ν. Then from Theorem 5 it follows that $\mathcal{A}$ is commutative algebra.

Consider the case where $q := |\mathbb{F}| < d+1$ and let $f(x_1, \ldots, x_n) = x_1^q - x_1$. Then f is of degree $q \leq d$ and for any $\boldsymbol{a} \in \mathbb{F}^n$ we have $f(\boldsymbol{a}) = 0$. If for some $a_0 \in \mathcal{A}$ we have $a_0^q - a_0 \neq 0$ then no simulator can distinguish between $f(a_0, 0, \ldots, 0)$ and $z(a_0, 0, \ldots, 0)$ where z is the zero function. Therefore when $|\mathbb{F}| < d+1$ we must have $a^q = a$ for all $a \in \mathcal{A}$.

(2 $\Rightarrow$ 3). If either $|\mathbb{F}| \geq d+1$ or $|\mathbb{F}| < d+1$ and $a^{|\mathbb{F}|} = a$ for every $a \in \mathcal{A}$ then by Theorem 5 there is a symmetric multilinear algorithm for the multiplication of $d+1$ elements in $\mathcal{A}$. By Theorem 4 there is a reducible symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$.

(3 $\Rightarrow$ 1) Follows from Theorem 3. □

7.1 Interpolation

In this section we prove some results for interpolation of multivariate polynomials that were used in previous sections

Lemma 12. *(Folklore) Let $f(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}[\boldsymbol{x}][\boldsymbol{y}]$ be a multivariate polynomial where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$, each y_i is of degree at most d in f and $|\mathbb{F}| \geq d+1$. Let $\boldsymbol{r} = (r_1, \ldots, r_m)$ and $M_{\boldsymbol{r}} = y_1^{r_1} \cdots y_m^{r_m}$ be any monomial where $r_i \leq d$ for every i. Let $A_{\boldsymbol{r}} \in \mathbb{F}[x_1, \ldots, x_n]$ be the coefficient of $M_{\boldsymbol{r}}$ in f. Let $B = \{\beta_0, \beta_1, \ldots, \beta_d\} \subseteq \mathbb{F}$. Then*

$$A_{\boldsymbol{r}} \in \mathrm{Span}_{\mathbb{F}}\{f(\boldsymbol{x}, z_1, \ldots, z_m) \mid z_i \in \{\beta_0, \ldots, \beta_d\}, i = 1, \ldots, m\}.$$

Proof. Consider the Vandermonde matrix $V(\beta_0, \ldots, \beta_d)$ and its inverse $U = (u_{i,j})_{i,j \in \{0,\ldots,d\}}$ such that

$$\sum_{j=0}^{d} u_{i,j} \beta_j^k = \delta_{i,k}$$

where $\delta_{i,k} = 1$ if $i = k$ and $\delta_{i,k} = 0$ otherwise. Then

$$A_{\boldsymbol{r}} = \sum_{j_1=0}^{d} \cdots \sum_{j_m=0}^{d} u_{r_1,j_1} \cdots u_{r_m,j_m} f(\boldsymbol{x}, \beta_{j_1}, \ldots, \beta_{j_m})$$

and the result follows. □

For integers $r \geq 0$ and $q \geq 2$ we denote $r \bmod_0(q-1)$ the integer $0 \leq r' \leq q-1$ such that $x^r \equiv x^{r'} \mod x^q - x$. For an integer vectors $\boldsymbol{r} = (r_1, \ldots, r_n)$ we denote $\boldsymbol{r} \bmod_0(q-1) = (r_1 \bmod_0(q-1), \ldots, r_n \bmod_0(q-1))$. This is equivalent to say that the monomial $y_1^{r_1} \cdots y_n^{r_n}$ is equal to the monomial $y_1^{r'_1} \cdots y_n^{r'_n}$ in $\mathbb{F}[\boldsymbol{y}]/(y_1^q - y_1, \ldots, y_n^q - y_n)$.

Lemma 13. *Let $f(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}[\boldsymbol{x}][\boldsymbol{y}]$ be a multivariate polynomial where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$. Let $\boldsymbol{r} = (r_1, \ldots, r_m)$ and $M_{\boldsymbol{r}} = y_1^{r_1} \cdots y_m^{r_m}$ be any monomial where $r_i < |\mathbb{F}|$ for every i. Let $A_{\boldsymbol{r}} \in \mathbb{F}[x_1, \ldots, x_n]$ be the sum of all the coefficients of $M_{\boldsymbol{r}'}$ in f where $\boldsymbol{r}' = \boldsymbol{r} \bmod_0(|\mathbb{F}| - 1)$. Then*

$$A_{\boldsymbol{r}} \in \mathrm{Span}_{\mathbb{F}}\{f(\boldsymbol{x}, z_1, \ldots, z_m) \mid z_i \in \mathbb{F}\}.$$

Proof. Consider the Vandermonde matrix $V(\beta_0, \ldots, \beta_{q-1})$ where $\mathbb{F} = \{\beta_0, \ldots, \beta_{q-1}\}$ and $q = |\mathbb{F}|$. Consider its inverse $U = (u_{i,j})_{i,j \in \{0,1,\ldots,q-1\}}$ such that

$$\sum_{j=0}^{q-1} u_{i,j} \beta_j^k = \delta_{i,k}.$$

Since $\alpha^q = \alpha$ for all $\alpha \in \mathbb{F}$ we also have

$$\sum_{j=0}^{q-1} u_{i,j} \beta_j^k = \sum_{j=0}^{q-1} u_{i,j} \beta_j^{k \bmod_0(q-1)} = \delta_{i,k \bmod_0(q-1)}$$

for any non-negative integer k. Then

$$A_{\boldsymbol{r}} = \sum_{j_1=0}^{q-1} \cdots \sum_{j_m=0}^{q-1} u_{r_1,j_1} \cdots u_{r_m,j_m} f(\boldsymbol{x}, \beta_{j_1}, \ldots, \beta_{j_m})$$

and the result follows. □

8 Conclusion and Open Problems

In this paper we showed that testers for multilinear forms of degree d and an $\mathbb{F}$-algebra $\mathcal{A}$ are equivalent to multilinear algorithms of the product of d elements in $\mathcal{A}$. Testers are defined with the most general terms possible and the fact that it is equivalent to a well structured algebraic problem will lead to a better understanding of testers.

Such algorithms were extensively studied for $d = 2$. This research open the way to further study of multilinear algorithms for $d > 2$ that might

contribute to many combinatoric problems. See [2] for many applications of testers. For example, it is interesting to study the multilinear complexity of the multiplication of d $n \times n$-matrices.

Using the above we were able to show that for any q there is a polynomial time construction of a bilinear algorithm for the multiplication of two elements in $\mathbb{F}_{q^n}$ with bilinear complexity $\mu = O(n)$. All previous multilinear algorithms were nonconstructive. This solved the open problem in [1, 12, 4]. The constant in the $O(n)$ is less than 24 and it interesting to close the gap with the lower bound $(2 + 1/(q-1))n$ [7].

We then study testers for homogeneous multivariate polynomials. We prove that testers for the class of all homogeneous multivariate polynomials of degree d and $\mathcal{A}$ are equivalent to symmetric multilinear algorithm for the multiplication of d elements in $\mathcal{A}$. Symmetric testers were studied in [11] for $d = 2$. It is shown that for field of characteristic greater than 2 every multilinear algorithm of multilinear complexity μ can be turned into symmetric multilinear algorithm of multilinear complexity 2μ. This result can be extended to any dimension d. For field of characteristic greater than d every multilinear algorithm of multilinear complexity μ can be turned into symmetric multilinear algorithm of multilinear complexity $2^d\mu$. It is interesting to find a better bound.

In Section 6 we study simulators and proved that a symmetric multilinear algorithm for the multiplication of $d + 1$ elements in an $\mathbb{F}$-algebra $\mathcal{A}$ with multilinear complexity ν gives a simulator for the class of polynomials of degree d and $\mathcal{A}$ of simulation complexity ν. Then we showed that from any polynomial time simulator for the set of all multivariate polynomials of degree d and an $\mathbb{F}$-algebra $\mathcal{A}$ of simulation complexity ν one can construct in polynomial time a symmetric multilinear algorithm for the product of d elements in $\mathcal{A}$ of multilinear complexity ν. Assuming $\mu^s_{\mathbb{F}}(d+1, \mathcal{A}) = O(\mu^s_{\mathbb{F}}(d, \mathcal{A}))$ which we believe is true, simulators and symmetric multilinear algorithms are equivalent. It is interesting to solve this open problem.

References

[1] D. V. Chudnovsky, G. V. Chudnovsky: Algebraic complexities and algebraic curves over finite fields. J. Complexity 4(4): 285-316 (1988)

[2] N. H. Bshouty. Testers and their Applications. Electronic Colloquium on Computational Complexity (ECCC) 19: 11 (2012).

[3] P. Bürgisser, M. Clausen, M. A. Shokrollahi. Algebraic complexity theory. vol. 315, Springer-Verlag, (1997).

[4] S. Ballet. An improvement of the construction of the D.V. and G.V. Chudnovsky algorithm for multiplication in finite fields. Theor. Comput. Sci. 352(1-3): 293-305 (2006).

[5] S. Ballet and R. Rolland. On the Bilinear Complexity of the Multiplication in finite fields. Seminaires and Congres, 11, p. 179-188, (2005).

[6] H. F. de Groote. Lectures on the Complexity of Bilinear Problems. Springer, Feb 23, (1987).

[7] A. Lempel, G. Seroussi and S. Winograd. On the Complexity of Multiplication in Finite Fields. Theor. Comput. Sci. 22: 285-296 (1983).

[8] V. Pan. How to multiply matrices faster?. Springer-Verlag, (1984).

[9] R. S. Pierce. Associative Algebras. Graduate texts in mathematics. Springer-Verlag. 88. (1982).

[10] H. Randriambololona. Bilinear complexity of algebras and the Chudnovsky-Chudnovsky interpolation method. arXiv:1107.0336v5, (2011). Journal of Complexity Volume 28, Issue 4,, pp. 489-517, (2012).

[11] G. Seroussi, A. Lempel. On Symmetric Algorithms for Bilinear Forms over Finite Fields. J. Algorithms 5(3): 327-344 (1984)

[12] I. Shparlinski, M. Tsfasman and S. Vladut. Curves with many points and multiplication in finite fields, in Coding theory and algebraic geometry (H. Stichtenoth and M. Tsfasrnan, eds.), Lect. Sotes in Math., vol. 1518, Springer-Verlag,, p. 145-169, (1992).

ECCC ISSN 1433-8092

http://eccc.hpi-web.de