Matemática Contemporânea, Vol. 7, 35–58
http://doi.org/10.21711/231766361994/rmc74

AN INTRODUCTION TO THE THEORY OF BILINEAR COMPLEXITY

M. A. Shokrollahi*

1. Introduction

In the course of these notes we shall investigate the problem, how many arithmetic operations are necessary to compute a finite set of multivariate polynomials over a field. One can assign different weights to different arithmetic operations; if, for example, large integers are involved in the computation, then it makes sense to give more weight to the multiplication of these numbers than to the addition. This stems from the fact that the best known algorithms for the multiplication of two numbers having n binary digits require $O(n \log n \log\log n)$ operations [8], whereas addition of two such numbers requires only $O(n)$ arithmetic operations. On the other side, if only computations with small rational numbers are involved, one should assign the same weight to addition and multiplication/division.

Different weightings of arithmetic operations lead usually to different theories. If, e.g., one wants to obtain the minimal number of additions necessary to compute a set of polynomials, one often uses different tools than those necessary for the study of the number of multiplications, say. The situation changes even more if one is interested in both of these numbers at the same time, i.e., if one is interested in the minimum (weighted) number of additions *and* multiplications necessary to compute a set of polynomials.

Before making things more precise, it is useful to clarify the problem we are interested in with the aid of some examples.

*This material was presented at the XII Escola de Álgebra, in Diamantina (Brazil) in August 1992.

Example 1.1. (*Multiplication of polynomials of degree* 1)

Here we are interested in an algorithm, which, given a pair of polynomials $p = p_1x + p_0$ and $q = q_1x + q_0$ over a field K, computes (the coefficiets of) their product $pq =: f = f_2x^2 + f_1x + f_0$ where

$$\begin{aligned} f_2 &= p_1q_1, \\ f_1 &= p_1q_0 + p_0q_1, \\ f_0 &= p_0q_0. \end{aligned}$$

Furthermore, this algorithm should use the minimum number of arithmetic operations. Since the coefficients of the polynomials p and q are not determined a priori and since the algorithm we are interested in should work for every set of coefficients in K, we can regard the coefficients as indeterminates over K. Because of notational reasons, we change the variables p_i to x_i and q_i to y_i. The computational problem is now to compute the set of polynomials

$$\begin{aligned} f_2 &= x_1y_1, \\ f_1 &= x_1y_0 + x_0y_1, \\ f_0 &= x_0y_0, \end{aligned}$$

using the least number of arithmetic operations.

Example 1.2. (*Multiplication of complex numbers*)

In this case, the problem is to give an algorithm to compute the product of two arbitrary complex numbers $c_1 = x_1 + iy_1$ and $c_2 = x_2 + iy_2$. Denote their product by $c_3 = f_1 + if_2$. Then

$$\begin{aligned} f_1 &= x_1x_2 - y_1y_2 \\ f_2 &= x_1y_2 + x_2y_1. \end{aligned}$$

Again, we can regard the coefficients of c_1 and c_2 as indeterminates over the field $\mathbf{R}$ of real numbers. The computation problem is to give an algorithm to compute the polynomials $f_1, f_2 \in \mathbf{R}[x_1, x_2, y_1, y_2]$, which uses the minimum number of arithmetic operations.

Example 1.3. (*Multiplication of* 2×2*-matrices*)

Given two arbitrary 2×2-matrices $X = (x_{ij})$ and $Y = (y_{ij})$ over a field K, compute their product $Z = XY = (z_{ij})$ using the minimum number of

arithmetic operations. Applying the rules for matrix multiplication, the problem consists of computing the following set of polynomials over K using the minimum number of arithmetic operations:

$$z_{ij} = x_{i1}y_{1j} + x_{i2}y_{2j}, \quad i,j \in \{1,2\}.$$

The problems introduced in the examples above may be stated in the following unified way:

> Let $x_1, \ldots, x_s$ be indeterminates over a field K. Further let $f_1, \ldots, f_m \in K[x_1, \ldots, x_s]$. It is required to give an optimal computation of $f_1, \ldots, f_m$ under the assumption that the elements of $K \cup \{x_1, \ldots, x_s\}$ can be computed without any cost.

2. Computation Sequences and Multiplicative Complexity

The problem at the end of the last section cannot be attacked yet, since it is by no means precisely stated. It is, for instance, not clear what is meant by a computation. In this section we shall make the concepts introduced intuitively in the last section more precise.

The following definition will clarify the concept of the "computation of polynomials".

Definition 2.1. *Let $x_1, \ldots, x_s$ be indeterminates over the field K. A finite set $(g_1, \ldots, g_r)$ in $K(x_1, \ldots, x_s)$ is called* a computation sequence of length r *if for all $\rho \leq r$ we have*

$$\exists\, u_\rho, v_\rho \in K + \sum_{i \leq s} Kx_i + \sum_{\sigma < \rho} Kg_\sigma : \quad g_\rho = u_\rho v_\rho \text{ or } g_\rho = u_\rho / v_\rho, v_\rho \neq 0.$$

If $f_1, \ldots, f_l$ are polynomials in $K[x_1, \ldots, x_s]$ and $(g_1, \ldots, g_r)$ is a computation sequence such that

$$\{f_1, \ldots, f_l\} \subseteq K + \sum_{i \leq s} Kx_i + \sum_{\rho \leq r} Kg_\rho,$$

then we say that $(g_1, \ldots, g_r)$ computes *the set* $\{f_1, \ldots, f_l\}$ *over* K. *A computation sequence* $(g_1, \ldots, g_r)$ *is called* division-free *if, with the above notations, each* g_ρ *is of the form* $g_\rho = u_\rho v_\rho$.

The length of a computation sequence is a measure for the cost of this sequence and hence may be used to define the concept of an optimal computation of a set of polynomials. But before going into details, let us say some words about the problem of weighting arithmetic operations in the model of computation induced by Definition 2.1. As is apparent from this definition, a computation sequence $(g_1, \ldots, g_r)$ computes any finite set of polynomials in

$$K + \sum_{i \leq s} K x_i + \sum_{\rho \leq r} K g_\rho.$$

This means that scalar multiplications or K-linear combinations of polynomials does not affect the cost of computation of these polynomials in the above model. Only when two nonconstant polynomials in the above set are multiplied or divided, one needs a longer computation sequence. If we denote this type of multiplications/divisions by *essential multiplications/divisions*[1] (as opposed to *scalar multiplications*, i.e., multiplications with elements of K), then the length of a computation sequence equals the number of essential multiplications/divisions in this sequence. The weighting of arithmetic operations is such that essential multiplications/divisions are weighted with 1 and scalar multiplications are weighted with 0, as are additions and subtractions.

Definition 2.2. *Let* $F := \{f_1, \ldots, f_l\} \subset K[x_1, \ldots, x_s]$. *A computation sequence of minimal length for* F *is called an* optimal computation *for* F. *The length of an optimal computation for* F *is called the* (non-scalar) complexity of F *and is denoted by* $L_{\{*,/\}}(F)$. *The length of an optimal division-free computation sequence for* F *is called the* multiplicative complexity *of* F *and is denoted by* $L_*(F)$ *or merely by* $L(F)$.

[1] In the sequel, we shall denote essential multiplications by $*$.

Note that we have suppressed the dependency of $L_{\{*,/\}}$ and L on the underlying field. See Example 2.6.

Example 2.3. (*Multiplication of polynomials of degree* 1)

We have already seen that computing the product of two polynomials of degree one over K is equivalent to computing the set $F = \{f_0, f_1, f_2\}$ of polynomials in $K[x_0, x_1, y_0, y_1]$ where

$$f_0 = x_0y_0, \quad f_1 = x_1y_0 + x_0y_1, \quad f_2 = x_1y_1.$$

A possible computation sequence for F is $(g_1, \ldots, g_4)$ where

$$\begin{aligned} g_1 &= x_0 * y_0 = f_0, \\ g_2 &= x_1 * y_0, \\ g_3 &= x_0 * y_1, \\ g_4 &= x_1 * y_1 = f_2. \end{aligned}$$

Since $f_1 = g_2 + g_3$, we have $L_{\{*,/\}}(F) \leq L(F) \leq 4$.

Example 2.4. (*Multiplication of complex numbers*)

As we have seen, this problem is equivalent to compute the polynomials

$$f_1 = x_1x_2 - y_1y_2, \quad f_2 = x_1y_2 + x_2y_1$$

in $\mathbf{R}[x_1, x_2, y_1, y_2]$. A possible computation sequence is $(g_1, \ldots, g_4)$ where

$$\begin{aligned} g_1 &= x_1 * x_2, \\ g_2 &= y_1 * y_2, \\ g_3 &= x_1 * y_2, \\ g_4 &= x_2 * y_1. \end{aligned}$$

Since $f_1 = g_1 - g_2$ and $f_2 = g_3 + g_4$, we have $L(\{f_1, f_2\}) \leq 4$.

Example 2.5. (*Multiplication of* 2×2*-matrices*)

The problem is to compute the polynomials

$$f_{ij} = x_{i1}y_{1j} + x_{i2}y_{2j}, \quad i, j \in \{1, 2\},$$

in $K[x_{ij}, y_{ij} \mid i, j \in \{1, 2\}]$. A possible computation sequence is given by

$$g_{ijk} := x_{ik} * y_{kj}, \quad i, j, k \in \{1, 2\}.$$

Since $f_{ij} = g_{ij1} + g_{ij2}$, we have $L(f_{ij} \mid i, j \in \{1, 2\}) \leq 8$.

Example 2.6. Usually, the complexity of a set of polynomials depends heavily on the field K. Consider for instance the polynomial $f := x_1^2 + x_2^2 \in \mathbf{R}[x_1, x_2]$. Let us compute the length $L_{\mathbf{R}}(f)$ of an optimal division-free computation sequence for f over $\mathbf{R}$. We claim that $L_{\mathbf{R}}(f) \geq 2$. Suppose not; then $L_{\mathbf{R}}(f) \leq 1$ and since $f \notin \mathbf{R} + \mathbf{R}x_1 + \mathbf{R}x_2$ we have $L_{\mathbf{R}}(f) = 1$. Hence, there exist $a_i, b_i, c_i \in \mathbf{R}, i = 1, 2, 3$ such that

$$x_1^2 + x_2^2 = (a_1 + b_1x_1 + c_1x_2)(a_2 + b_2x_1 + c_2x_2) + a_3 + b_3x_1 + c_3x_2$$

Comparing coefficients we get

$$b_1b_2 = c_1c_2 = 1,\ b_1c_2 + c_1b_2 = 0.$$

This implies that $(b_1c_2)^2 + 1 = 0$, which is impossible (over $\mathbf{R}$). Hence $L_{\mathbf{R}}(f) \geq 2$. On the other hand, the computation sequence

$$g_1 := x_1 * x_1, g_2 := x_2 * x_2$$

clearly computes f, i.e., $L_{\mathbf{R}}(f) = 2$. What about $L_{\mathbf{C}}(f)$? Note that over $\mathbf{C}$ we have $f = (x_1 + ix_2)(x_1 - ix_2)$, hence $L_{\mathbf{C}}(f) = 1$. This result may also be extended to any field: If K contains a primitive fourth root of unity, then $L_K(f) = 1$, otherwise $L_K(f) = 2$.

It may look strange that divisions may help when computing a set of polynomials. The following example shows that this indeed may be the case.

Example 2.7. Consider the polynomial $f = x^{31}$ over any field K. The following computation sequence of length 7 computes f:

$$g_1 := x * x, \quad g_2 = g_1 * x, \quad g_3 := g_1 * g_2, \quad g_4 := g_3 * g_1,$$
$$g_5 := g_4 * g_4, \quad g_6 := g_5 * g_5, \quad g_7 := g_6 * g_1$$

since $g_7 = x^{31}$. It can be shown that $L(x^{31}) = 7$. On the other hand, computing x^{32} by squaring x five times and then dividing x^{32} by x gives a computation sequence of length 6 for the computation of x^{31}. The division step thus gives a better computation.

We may ask how much divisions may help or even, whether there exist classes of polynomials such that for their computation divisions do not help at all?

This question has been answered by Strassen [10]. Before stating (a simplified version of) his result, we need a definition.

Definition 2.8. *Let $x_1, \ldots, x_s$ be indeterminates over the field K. A computation sequence $(g_1, \ldots, g_r)$ in $K(x_1, \ldots, x_s)$ is called* quadratic *if*

$$\forall i \leq r \; \exists u_i, v_i \in \sum_{i=1}^{s} K x_i: \qquad g_i = u_i v_i.$$

In this case we shall also write $((u_1, v_1), \ldots, (u_r, v_r))$ for $(g_1, \ldots, g_r)$.

Theorem 2.9. (Strassen) *Let K be an infinite field and F be a finite set of* quadratic polynomials *in $K[x_1, \ldots, x_s]$. Then there exists a quadratic computation sequence of length $L_{\{*,/\}}(F)$ which computes F. In particular, $L_{\{*,/\}}(F) = L(F)$.*

The reader may verify that the algorithms given so far for the multiplication of polynomials, multiplication of complex numbers and multiplication of 2×2-matrices, are all quadratic.

Example 2.10. Let us compute the multiplicative complexity of a single quadratic polynomial, or equivalently, a quadratic form over a field K of characteristic not equal to 2.

Let f be a quadratic form in n variables. It is well known (see the article of T. Smith in this volume) that

$$f \sim x_1 x_2 + \cdots + x_{2p-1} x_{2p} + h(x_{2p+1}, \ldots, x_n)$$

where $\sim$ is the usual equivalence of quadratic forms and h is the anisotropic part of f. It is easily seen that the multiplicative complexity is constant on equivalence classes of quadratic forms (this is left as an exercise), hence the

above decomposition shows that $L(f) \leq p + (2n - p) = n - p$, since $h \sim \alpha_{2p+1}x_{2p+1}^2 + \cdots + \alpha_n x_n^2$ for suitable $\alpha_{2p+1}, \ldots, \alpha_n$ in K.

Now we want to prove that $L(f) \geq n - p$ which will show $L(f) = n - p$. To this end, let a quadratic computation sequence $((u_1, v_1), \ldots, (u_r, v_r))$ of length r for f be given. Then $f(x) = \sum_{i=1}^{r} u_i(x)v_i(x)$. Let $\{u_1 = \cdots = u_r = 0\}$ denote the set of common zeros of $u_1, \ldots, u_r$. The dimension of this set is at least $n - r$, since $u_1, \ldots, u_r$ are linear forms. Since this set clearly lies in the set of zeros of f and the latter has dimension p by assumption, we obtain $p \geq n - r$, which implies the assertion.

As a consequence we obtain $L(x_1x_2 + \cdots + x_{2n-1}x_{2n}) = n$ and $L(x_1^2 + \cdots + x_n^2) = \lceil n \rceil / 2$ over the field $\mathbf{C}$ of complex numbers.

We may still ask for simpler algorithms for computing quadratic polynomials. To obtain these, we have to restrict ourselves to a subclass of quadratic polynomials.

Definition 2.11. *Let K be a field and $x_1, \ldots, x_s, y_1, \ldots, y_m$ be indeterminates over K. A polynomial $p \in K[x_1, \ldots, x_s, y_1, \ldots, y_m]$ is called* bilinear (with respect to $\underline{x} := (x_1, \ldots, x_s)$ and $\underline{y} := (y_1, \ldots, y_m)$) *if*

$$p(\underline{x}, \underline{y}) = \sum_{i,j} a_{ij}x_iy_j,$$

for some $a_{ij} \in K$.

Bilinear polynomials may be computed by simpler computation sequences than quadratic ones.

Definition 2.12. *Let K be a field and $x_1, \ldots, x_s, y_1, \ldots, y_m$ be indeterminates over K. A quadratic computation sequence $((u_1, v_1), \ldots, (u_r, v_r))$ in $K[x_1, \ldots, x_s, y_1, \ldots, y_m]$ is called* bilinear *if for all $i = 1, \ldots, r$ the u_i, resp. v_i are linear homogeneous in $x_1, \ldots, x_s$, resp. $y_1, \ldots, y_m$.*

For any finite set F of bilinear polynomials over K there exists a bilinear computation sequence which computes F: clearly, there exists by Theorem 2.9 a quadratic computation sequence $((U_1, V_1), \ldots, (U_q, V_q))$ for the computation of F. Hence $F \subset \sum_{i=1}^{q} KU_iV_i$. Now let for every $i = 1, \ldots, r$ $U_i = u_i + u_i'$, $V_i = v_i + v_i'$ where u_i, v_i are linear homogeneous in $x_1, \ldots, x_s$ and u_i', v_i' are linear homogeneous in $y_1, \ldots, y_t$. Then, since F is a set of bilinear polynomials, we have $F \subset \sum_{i=1}^{q} Ku_iv_i' + \sum_{i=1}^{q} Ku_i'v_i$. This shows that $((u_1, v_1'), \ldots, (u_q, v_q'), (u_1', v_1), \ldots, (u_q', v_q))$ is a bilinear computation sequence for F. The following definition thus makes sense.

Definition 2.13. *Let F be a finite set of bilinear polynomials over K. The minimum length of a bilinear computation sequence for F is called the* bilinear complexity *(or* rank*) of F and is denoted by $R(F)$.*

Clearly, $L(F) \leq R(F)$ for a set F of bilinear polynomials, since bilinear computation sequences form a subset of quadratic computation sequences. But the foregoing argumentation also shows the following.

Lemma 2.14. *Let F be a finite set of bilinear polynomials over K. Then*

$$L(F) \leq R(F) \leq 2L(F).$$

We finish this section with a couple of examples.

Example 2.15. (*Multiplication of polynomials of degree* 1)

We want to give an upper estimate for $R = R(f_0, f_1, f_2)$ where

$$f_0 = x_0y_0, \quad f_1 = x_1y_0 + x_0y_1, \quad f_2 = x_1y_1.$$

The trivial algorithm leads to $R \leq 4$, as was shown before (It is easy to check that the trivial algorithm is bilinear). The following computation sequence shows that $R \leq 3$:

$$\begin{aligned} g_1 &= x_0 * y_0 = f_0, \\ g_2 &= (x_0 + x_1) * (y_0 + y_1), \\ g_3 &= x_1 * y_1 = f_2. \end{aligned}$$

Since $f_1 = g_2 - g_1 - g_3$, the g_i constitute a bilinear computation sequence for $\{f_0, f_1, f_2\}$ of length 3.

Example 2.16. (*Multiplication of complex numbers*)

Again, let

$$f_1 = x_1x_2 - y_1y_2, \quad f_2 = x_1y_2 + x_2y_1$$

be the bilinear polynomials corresponding to the multiplication of complex numbers. The computation sequence

$$\begin{aligned} g_1 &= x_1 * (x_2 + y_2) \\ g_2 &= y_2 * (x_1 + y_1) \\ g_3 &= x_2 * (y_1 - x_1) \end{aligned}$$

computes $\{f_1, f_2\}$, since $f_1 = g_1 - g_2$ and $f_2 = g_1 + g_3$. Hence $R(\{f_1, f_2\}) \leq 3$.

Example 2.17. (*Multiplication of* 2×2*-matrices*)

Let again f_{ij} denote the bilinear polynomials corresponding to the multiplication of 2×2-matrices. We have already seen that $R(\{f_{ij} \mid i,j = 1,2\}) \leq 8$. Consider the following computation sequence:

$$\begin{aligned} g_1 &= (x_{11} + x_{22}) * (y_{11} + y_{22}) & g_2 &= (x_{21} + x_{22}) * y_{11} \\ g_3 &= x_{11} * (y_{12} - y_{22}) & g_4 &= x_{22} * (-y_{11} + y_{21}) \\ g_5 &= (x_{11} + x_{12}) * y_{22} & g_6 &= (-x_{11} + x_{21}) * (y_{11} + y_{12}) \\ g_7 &= (x_{12} - x_{22}) * (y_{21} + y_{22}). \end{aligned}$$

Since

$$\begin{aligned} f_{11} &= g_1 + g_4 - g_5 + g_7 & f_{12} &= g_3 + g_5 \\ f_{21} &= g_2 + g_4 & f_{22} &= g_1 - g_2 + g_3 + g_6, \end{aligned}$$

this sequence computes f_{ij} over any field (even over any ring!). The corresponding bilinear algorithm is known as Strassen's matrix multiplication algorithm [9]. The major point of this algorithm is that it is valid over any ring. Hence one may interpret x_{ij} and y_{ij} as matrices over a field K and use the algorithm recursively. This gives an algorithm for multiplying $n \times n$-matrices which uses asymptotically $O(n^{\log_2 7}) = O(n^{2.80735\ldots})$ multiplications (which is better than the naive algorithm which needs $O(n^3)$ multiplications).

In the examples above we have only given upper bounds for the bilinear complexities of the corresponding bilinear polynomials. The major problem is now to give good or even matching lower bounds for these quantities. This will be done in the next sections.

3. Rank of Bilinear Mappings

Bilinear polynomials in $(s+l)$ indeterminates over a field K can be viewed as bilinear forms of the vector space $K^s \times K^l$. Indeed, if

$$p(x_1, \ldots, x_s, y_1, \ldots, y_l) = \sum_{i,j} a_{ij} x_i y_j$$

is a bilinear polynomial over K, then for the pair $(e_1, \ldots, e_s)$ and $(e'_1, \ldots, e'_l)$ of natural bases of K^s, resp. K^l, p induces a bilinear form ϕ_p defined by $\phi_p(e_i, e'_j) := a_{ij}$. Analogously, linear homogeneous polynomials in m indeterminates may be viewed as linear forms of K^m.

A sequence $(z_1, \ldots, z_n)$ of bilinear polynomials (with respect to $x_1, \ldots, x_s$ and $y_1, \ldots, y_l$) over K induces a bilinear mapping of $\Phi: K^s \times K^l \to K^n$ by requiring that the kth coordinate of Φ be equal to the bilinear form induced by z_k. In this way, bilinear polynomials induce bilinear mappings. It is also possible to speak about bilinear algorithms for bilinear mappings. The next definition makes this concept precise.

Definition 3.1. *Let U, V, and W be finite dimensional vector spaces over the field K and $\Phi: U \times V \to W$ be a bilinear mapping. Denote by U^*, resp. V^*, the dual spaces of U, resp. V. The* bilinear complexity *or* rank *$R(\Phi)$ of Φ is the minimal number r such that there exist $u_1, \ldots, u_r \in U^*$, $v_1, \ldots, v_r \in V^*$, and $w_1, \ldots, w_r \in W$, such that*

$$\forall\, x \in U, y \in V: \quad \Phi(x,y) = \sum_{i=1}^{r} u_i(x) v_i(y) w_i.$$

The proof of many results on the rank of bilinear mappings become more transparent if one uses the terminology of tensors and tensor product.

Definition 3.2. *A tensor product $(U \otimes V, \tau)$ of U and V consists of a vector space $U \otimes V$ over K and a bilinear map $\tau: U \times V \to U \otimes V$ such that*

(1) *The K-span of the image of τ equals $U \otimes V$,*

(2) *(Universal mapping property) For every vector space W and every bilinear map $\Phi: U \times V \to W$ there exists a linear map $\phi: U \otimes V \to W$ such that $\Phi = \phi\tau$.*

While the uniqueness of the tensor product (up to isomorphism) follows directly from the definition, the proof of the existence involves a certain construction which can be read in any book on multilinear algebra or algebra (see, e.g., [7]). It can be proved easily that the tensor product is associative, i.e.,

$$(U \otimes V) \otimes W \simeq U \otimes (V \otimes W) \simeq: U \otimes V \otimes W.$$

for K-spaces U, V, and W.

Let U, V, and W be K-spaces. The connection between the K-space $\mathrm{Bil}(U \times V, W)$ and tensor products is given by the following isomorphism.

Lemma 3.3. $\mathrm{Bil}(U \times V, W) \simeq U^* \otimes V^* \otimes W$, *where U^* and V^* denote the dual spaces of U and V.*

Proof. (Sketch) It can be proved that the homomorphism $h: U^* \otimes V^* \otimes W \to \mathrm{Bil}(U \times V, W)$ defined by $u^* \otimes v^* \otimes w \mapsto ((a,b) \mapsto u^*(a)v^*(b)w)$ is an isomorphism [7]. □

To $\Phi \in \mathrm{Bil}(U \times V, W)$ corresponds a unique tensor $t \in U^* \otimes V^* \otimes W$ according to Theorem 3.3. Further, if $\Phi(a,b) = \sum_{\rho \leq r} u_\rho(a)v_\rho(b)w_\rho$ for all $(a,b) \in U \times V$, we have in view of the above isomorphism $\Phi = \sum_{\rho \leq r} u_\rho \otimes v_\rho \otimes w_\rho$. If we call an element $u \otimes v \otimes w \in U^* \otimes V^* \otimes W$ a *triad*, we obtain that the rank of a bilinear mapping Φ is the minimum number r such that the tensor corresponding to Φ can be represented as a sum of r triads. One can thus also speak of the *rank* of a tensor in $U^* \otimes V^* \otimes W$. The rank of the bilinear mapping Φ is sometimes also called the *tensor rank* of Φ. In the sequel we shall make frequent implicit use of the above isomorphism and mix up tensors and bilinear mappings.

Example 3.4. Here we want to show that the rank of a bilinear mapping is a generalization of the concept of the rank of a linear map. Let U and V be

K-spaces. If $\phi \in \mathrm{Bil}(U \times V, K)$, then for $a \in U$ the mapping ϕ_a which assigns to $b \in V$ the value $\phi_a(b) := \phi(a, b)$ is a linear form on V, i.e., $\phi_a \in V^*$. Then $\mathrm{Bil}(U \times V, K) \simeq \mathrm{Hom}(U, V^*)$ under the isomorphism $\phi \mapsto (h_\phi\colon\ a \mapsto \phi_a)$. We claim that $R(\phi) = \mathrm{rk}\,(h_\phi)$, which shows that the rank of a bilinear map is a generalization of the concept of rank of a linear map.

Suppose that $\phi = \sum_{\rho \le r} u_\rho \otimes v_\rho$, where $u_\rho \in U^*$, and $v_\rho \in V^*$. Then for any $a \in U$ we have

$$h_\phi(a) = \phi_a \in \sum_{\rho \le r} u_\rho(a) v_\rho,$$

hence $\mathrm{rk}\,(h_\phi) \le r$. Thus $\mathrm{rk}\,(h_\phi) \le R(\phi)$.

On the other hand, let $\mathrm{rk}\,(h_\phi) = r$ and $v_1, \ldots, v_r$ be a basis of the image of h_ϕ. Then, for any $a \in U$ there exist $u_1(a), \ldots, u_r(a) \in K$ such that $h_\phi(a) = \sum_{\rho \le r} u_\rho(a) v_\rho$. For the linear forms u_ρ thus defined we obtain by the definition of h_ϕ: $\phi = \sum_{\rho \le r} u_\rho \otimes v_\rho$, which shows that $R(\phi) \le \mathrm{rk}\,(h_\phi)$. All in all we obtain $R(\phi) = \mathrm{rk}\,(h_\phi)$.

In the rest of this section we shall develop a simple but powerful tool for proving lower or upper bounds for the bilinear complexity of a bilinear mapping, by reducing it to the bilinear complexity of some other bilinear mapping. We have first to recall some basic facts.

Suppose that U_i, V_i, $i = 1, 2$, are finite dimensional K-spaces and $f \in \mathrm{Hom}(U_1, U_2)$, $g \in \mathrm{Hom}(V_1, V_2)$. Let $(U_i \otimes V_i, \tau_i)$, $i = 1, 2$ be the tensor product of U_i and V_i. Then $\tau_2(f \times g)$ is a bilinear mapping from $U_1 \times V_1$ to $U_2 \otimes V_2$, hence there exists a unique homomorphism h from $U_1 \otimes V_1$ to $U_2 \otimes V_2$ such that $h(u_1 \otimes v_1) = f(u_1) \otimes g(v_1)$. We denote this homomorphism by $f \otimes g$.

Let U and V be K-spaces and $\varphi \in \mathrm{Hom}(U, V)$. Then $\varphi^*\colon V^* \to U^*$ defined by $\varphi^*(\lambda) := \lambda\varphi$ is a homomorphism.

Lemma 3.5. *Let U_i, V_i, W_i, $i = 1, 2$, be finite dimensional K-spaces and $\phi \in U_1^* \otimes V_1^* \otimes W_1$. Suppose that $\varphi^* \in \mathrm{Hom}(U_1^*, U_2^*)$, $\psi^* \in \mathrm{Hom}(V_1^*, V_2^*)$, and $\eta \in \mathrm{Hom}(W_1, W_2)$. Then $R((\varphi^* \otimes \psi^* \otimes \eta)(\phi)) \le R(\phi)$.*

Proof. Let $\phi = \sum_{\rho \le r} u_\rho \otimes v_\rho \otimes w_\rho$, where $r = R(\phi)$, $u_\rho \in U_1^*$, $v_\rho \in V_1^*$, and $w_\rho \in W_1^*$ and $\Psi := (\varphi^* \otimes \psi^* \otimes \eta)(\phi)$. Then $\Psi = \sum_{\rho \le r} \varphi^*(u_\rho) \otimes \psi^*(v_\rho) \otimes \eta(w_\rho)$. Noting that $\varphi^*(u_\rho) \in U_2^*$, $\psi^*(v_\rho) \in V_2^*$, and $\eta(\dot{w}_\rho) \in W_2$ we obtain a representation of Ψ as the sum of $R(\phi)$ triads which implies that $R(\Psi) \le R(\phi)$. □

The following lemma shows how this reduction technique can be used.

Lemma 3.6. *Let U_i, V_i, W_i, $i = 1, 2$, be finite dimensional K-spaces and $\phi_i \in \mathrm{Bil}(U_i \times V_i, W_i)$, $i = 1, 2$. Further let $\varphi \in \mathrm{Hom}(U_1, U_2)$, $\psi \in \mathrm{Hom}(V_1, V_2)$, and $\eta \in \mathrm{Hom}(W_1, W_2)$ be such that the following diagram commutes:*

$$\begin{array}{ccccc} U_1 & \times & V_1 & \xrightarrow{\phi_1} & W_1 \\ \downarrow{\scriptstyle \varphi} & & \downarrow{\scriptstyle \psi} & & \downarrow{\scriptstyle \eta} \\ U_2 & \times & V_2 & \xrightarrow{\phi_2} & W_2 \end{array}.$$

Then we have:

(1) *If φ and ψ are surjective, then $R(\phi_2) \le R(\phi_1)$.*

(2) *If η is injective, then $R(\phi_1) \le R(\phi_2)$.*

Proof. Let $\phi_i = \sum_{\rho \le r_i} u_\rho^{(i)} \otimes v_\rho^{(i)} \otimes w_\rho^{(i)}$, $i = 1, 2$, where $u_\rho^{(i)} \in U_i^*$, $v_\rho^{(i)} \in V_i^*$, and $w_\rho^{(i)} \in W_i$, and $r_i := R(\phi_i)$. The condition that the above diagram commutes translates to

$$\sum_{\rho \le r_1} u_\rho^{(1)} \otimes v_\rho^{(1)} \otimes \eta(w_\rho^{(1)}) = \sum_{\rho \le r_2} u_\rho^{(2)} \varphi \otimes v_\rho^{(2)} \psi \otimes w_\rho^{(2)}.$$

It is easily seen that the above condition is equivalent to

$$(\mathrm{id} \otimes \mathrm{id} \otimes \eta)(\phi_1) = (\varphi^* \otimes \psi^* \otimes \mathrm{id})(\phi_2). \tag{1}$$

(1) The surjectivity of φ and ψ implies the existence of $\varphi^{-1} \in \mathrm{Hom}(U_2, U_1)$, $\psi^{-1} \in \mathrm{Hom}(V_2, V_1)$ such that $\varphi\varphi^{-1} = \mathrm{id}_{U_2}$ and $\psi\psi^{-1} = \mathrm{id}_{V_2}$. Application of $((\varphi^{-1})^* \otimes (\psi^{-1})^* \otimes \mathrm{id})$ to Equation (1) yields

$$((\varphi^{-1})^* \otimes (\psi^{-1})^* \otimes \eta)(\phi_1) = \phi_2.$$

Now Lemma 3.5 implies that $R(\phi_2) \leq R(\phi_1)$.

(2) The injectivity of η implies that there exists $\eta^{-1} \in \mathrm{Hom}(W_2, W_1)$ such that $\eta^{-1}\eta = \mathrm{id}_{W_1}$. Application of $(\mathrm{id} \otimes \mathrm{id} \otimes \eta^{-1})$ to Equation (1) yields

$$\phi_1 = (\varphi^* \otimes \psi^* \otimes \eta^{-1})(\phi_2).$$

Hence Lemma 3.5 implies that $R(\phi_1) \leq R(\phi_2)$. □

We shall use the above methods in the next chapter.

4. Lower Bounds for some Computational Problems

As was said before, the aim of (algebraic) complexity theory is to give in some way the minimum number of operations necessary to compute (algebraic) quantities. Hence, one tries to make assertions about *all possible* algorithms for the problem to solve.

To determine the exact minimum number of operations necessary to compute an algebraic problem, one has—in some way—to estimate from below this number and at the same time to find a *matching upper bound*, i.e., to give an algorithm (or prove the existence of an algorithm) which uses this number of arithmetic operations. So the problem is divided into two parts: Proving *lower bounds* and finding (matching) *upper bounds*. The second problem is usually connected to the design of algorithms and is generally considered to be easier than the first, for which only a few general techniques are known. Proving (nontrivial) lower bounds for algebraic computation problems is one of the most challenging topics in complexity theory.

Below we shall discuss some bilinear problems and give nontrivial lower bounds for their bilinear complexity. Since we have not developed the tools for proving most of these bounds, we shall content ourselves to the proofs of the most simple lower bounds.

Let K be a field and D be a finite division algebra over K. We consider the bilinear mapping $\mu: D \times D \to D$ defined by $\mu(a,b) := ab$, i.e., we consider the multiplication in D, where D is regarded as a vector space over K.

Definition 4.1. *The multiplicative complexity, resp. rank of the bilinear mapping μ as above is called the* multiplicative complexity, resp. rank of D/K *and is denoted by $L(D/K)$, resp. $R(D/K)$ or merely $L(D)$, resp. $R(D)$ if K is clear from the context.*

We identify μ with the tensor in $D^* \otimes D^* \otimes D$ under the isomorphism given in Theorem 3.3. Suppose that there exists a bilinear algorithm of length r for μ. We can thus represent μ as a sum of r triads:

$$\mu = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i,$$

where $u_i, v_i \in D^*$ and $w_i \in D$. Let x be a nonzero element of K. Then

$$\mu(x, D) = xD \subseteq \sum_{i=1}^{r} u_i(x) v_i(D) w_i.$$

Denote the dimension of D over K by n. There exists a nonzero x in L such that $u_1(x) = \cdots = u_{n-1}(x) = 0$. For this x we have $xD \subseteq \sum_{i=n}^{r} u_i(x) v_i(D) w_i \subseteq \sum_{i=n}^{r} K w_i$. Since D is a division algebra and x is nonzero, $xD = D$ and hence the K-dimension of xD is n, we see that $r - n + 1 \geq n$, hence $r \geq 2n - 1$. We have thus proved the following:

Theorem 4.2. *Let K be a field and D be a division algebra of degree n of K. Then we have $R(D/K) \geq 2n - 1$.*

In order to compare the order of difficulty, we shall give here a proof for $L(D/K) \geq 2n - 1$, too.

Theorem 4.3. *Let K be a field and D be a division algebra of degree n of K. Then we have $L(D/K) \geq 2n - 1$.*

Proof. Let Φ denote the multiplication in D. For $\rho = 1 \ldots, r$ let $u_\rho, v_\rho \in (D \oplus D)^*$, $w_\rho \in D$ be such that $\Phi(a,b) = ab = \sum_{\rho=1}^{r} u_\rho(a,b)v_\rho(a,b)w_\rho$ for all $a, b \in D$. We may suppose that $r = L(D)$, i.e., we consider a minimal algorithm for the multiplication in D.

First, we prove that $r \geq n$. To this end, note that $\operatorname{Im}\Phi = D$ and that $\operatorname{Im}\Phi \subseteq \sum_{\rho=1}^{r} Kw_\rho$, hence $r \geq n$. Now assume that $r < 2n-1$. Let $W := \sum_{\rho=1}^{n-1} Kw_\rho$. After interchanging some u_ρ with some v_ρ, we obtain p with $n-1 \leq p \leq r$ such that

- $v_n \ldots, v_p$ are linearly independent on $0 \oplus D$,
- $u_{p+1}, \ldots, u_n, v_{p+1}, \ldots, v_n$ are linearly dependent on $v_n, \ldots, v_p$ on $0 \oplus D$.

Since $p + 1 - n < n$, there exists $0 \neq y \in D$ such that $v_n(0,y) = \cdots = v_p(0,y) = 0$. The conditions above imply that $v_n(0,y) = \cdots = v_r(0,y) = 0$, and $u_{p+1}(0,y) = \cdots = u_r(0,y) = 0$.

Let $a \in D$ be arbitrary and let $b \in D$ be such that $v_n(a,b) = \cdots = v_p(a,b) = 0$. Then we obtain

$$\begin{aligned} ay &= a(b+y) - ab \\ &= \sum_{\rho=1}^{r} u_\rho(a,b+y)v_\rho(a,b+y)w_\rho - \sum_{\rho=1}^{r} u_\rho(a,b)v_\rho(a,b)w_\rho \\ &= \sum_{\rho=1}^{r} (\underbrace{u_\rho(0,y)}_{\substack{=0 \\ \text{for } \rho > p}} \underbrace{v_\rho(a,b)}_{\substack{=0 \\ \text{for } n \leq \rho \leq p}} + (u_\rho(a,b) + u_\rho(0,y)) \underbrace{v_\rho(0,y)}_{\substack{=0 \\ \text{for } \rho \geq n}}) w_\rho \\ &\in W = \sum_{\rho=1}^{n-1} Kw_\rho. \end{aligned}$$

This implies that $Dy \subseteq W \neq D$, which is a contradiction since D is assumed to be a division algebra. □

What is $R(D/K)$ for a finite dimensional division algebra D over K? Let us first discuss the case where D is a simple field extension of K. The question of upper bounds for the rank $R(D/K)$ is very much related to the rank of polynomial multiplication. To be more precise, for a natural number l let $K[x]_l$

denote the K-space of polynomials of degree less than l over K. Let $\Phi_K^{l,m} \in \mathrm{Bil}(K[x]_l \times K[x]_m, K[x]_{l+m-1})$ be the polynomial multiplication map. Then $R(D/K)$ is related to $R(\Phi_K^{n,n})$. This is the context of the following lemmas.

Lemma 4.4. *Let D be a simple field extension of degree n of the field K. Then $R(D/K) \leq R(\Phi_K^{n,n})$.*

Proof. Let $p(x)$ be a monic irreducible polynomial of degree n over K such that $D \simeq K[x]/(p(x))$ and κ be the residue class mapping $K[x] \to K[x]/(p(x)) \simeq D$. We obtain the following commutative diagram

$$\begin{array}{ccccc} K[x]_n & \times & K[x]_n & \xrightarrow{\Phi_K^{n,n}} & K[x]_{2n-1} \\ \downarrow{\scriptstyle \kappa_1} & & \downarrow{\scriptstyle \kappa_1} & & \downarrow{\scriptstyle \kappa_2} \\ D & \times & D & \xrightarrow{\nu} & D \end{array}$$

where ν is the multiplication in D, κ_1 is the restriction of κ to $K[x]_n$, and κ_2 is the restriction of κ to $K[x]_{2n-1}$. Now, since κ_1 is surjective, Lemma 3.6(1) implies that $R(D/K) = R(\nu) \leq R(\Phi_K^{n,n})$. □

Lemma 4.5. *Let l and m be positive integers and K be a field such that $|K| \geq l + m - 2$. Then $R(\Phi_K^{l,m}) \leq l + m - 1$.*

Proof. Let $\alpha_1, \ldots, \alpha_{l+m-2}$ be pairwise different elements of K. For $k \geq 1$ we define $\gamma_k \colon K[x]_k \to K^{l+m-1}$ by $\gamma_k(f) := (f(\alpha_1), \ldots, f(\alpha_{l+m-2}), f(\infty))$, where $f(\infty)$ stands for the coefficient of x^{k-1} of f. It is clear that γ_k is injective if and only if $k \leq l + m - 1$ and bijective if and only if $k = l + m - 1$. Now consider the following commutative diagram

$$\begin{array}{ccccc} K[x]_l & \times & K[x]_m & \xrightarrow{\Phi_K^{l,m}} & K[x]_{l+m-1} \\ \downarrow{\scriptstyle \gamma_l} & & \downarrow{\scriptstyle \gamma_m} & & \downarrow{\scriptstyle \gamma_{l+m-1}} \\ K^{l+m-1} & \times & K^{l+m-1} & \xrightarrow{\mu} & K^{l+m-1} \end{array}$$

where μ is component-wise multiplication. Since γ_{l+m-1} is bijective,we obtain $R(\Phi_K^{l,m}) \leq R(\mu) \leq l+m-1$ by Lemma 3.6 (2). □

As a corollary we obtain from Theorem 4.2, Lemma 4.4, and Lemma 4.5 the following.

Corollary 4.6. *Let D be a simple field extension of K of degree n and $|K| \geq 2n-2$. Then $R(D/K) = 2n-1$.*

Applying this corollary to the case $K = \mathbf{R}$ and $D = \mathbf{C}$, we see that the bilinear algorithm for multiplication of complex numbers introduced in Example 2.16 is optimal (in the sense of bilinear complexity) and that $R(\mathbf{C}/\mathbf{R}) = 3$. Also, application of Lemma 4.5 to the multiplication of polynomials of degree less or equal to one shows that the bilinear algorithm introduced in Example 2.15 is optimal and that $R(\Phi_K^{2,2}) = 3$ for any field K.

We state without proof a theorem of Baur on the rank of central division algebras over a field K.

Theorem 4.7. *Let D be a central K-division algebra of dimension n. Then $R(D) \geq 2n-2+\sqrt{n}$.*

If $D = \mathbf{H}$ is the algebra of real quaternions, we obtain $R(\mathbf{H}) \geq 8$. In fact, $R(\mathbf{H}) = 8$.

The leading problem of bilinear complexity is that of matrix multiplication. Here one wants to compute the rank of the bilinear mapping which assigns to every two square matrices their product. More precisely, if n is a positive integer, we define $L(K^{n\times n})$, resp. $R(K^{n\times n})$ as the multiplicative complexity, resp. rank of the multiplication map in $K^{n\times n}$. Concerning lower bounds for $R(K^{n\times n})$ we have the following.

Theorem 4.8. *For any field K we have $R(K^{n\times n}) \geq 2n^2-1$.*

Proof. The following proof has been taken from [2]. During this proof we denote by A the ring $K^{n\times n}$. We shall need the following facts about A.

(i) Any minimal left (right) ideal of A has K-dimension n.

(ii) Any maximal left (right) ideal of A has K-dimension $n^2 - n$.

(iii) No right ideal $R \neq 0$ of A is contained in a left ideal $L \neq A$ of A.

The proofs of these assertions are left as an exercise.

Suppose that $r := R(K^{n\times n}) < 2n^2 - 1$. For $\rho = 1, \ldots, r$ let $u_\rho, v_\rho \in A^*$, and $w_\rho \in A$ be such that

$$\forall\, a, b \in A: \quad ab = \sum_{\rho=1}^{r} u_\rho(a) v_\rho(b) w_\rho. \tag{2}$$

Observe first that $\sum_{\rho=1}^{r} K u_\rho = A^*$. Otherwise, there exists $0 \neq a \in A$ such that $u_\rho(a) = 0$ for all $1 \leq \rho \leq r$. This implies $ab = 0$ for all $b \in A$ in view of Equation (2). Hence $a = 0$, a contradiction. We may assume w.l.o.g. that $u_1, \ldots, u_{n^2}$ are linearly independent (note that A has dimension n^2 over K). Since $r < 2n^2 - 1$, $\sum_{\rho=n^2}^{r} K v_\rho \neq A^*$. Hence, there exists $0 \neq b \in A$ such that $v_{n^2}(b) = \cdots = v_r(b) = 0$. By Equation (2) we have then $Ab \subseteq \sum_{\rho=1}^{n^2-1} K w_\rho$. Thus, Ab is a proper left ideal of A, hence it is contained in a maximal left ideal L of A. We may assume that $\sum_{\rho=1}^{n-1} K w_\rho \subseteq A\, b$. By the same argumentation as above, $v_1, \ldots, v_r$ generate A^*, hence we may assume that $v_n, \ldots, v_{n^2-1}$ are linearly independent over L, since L has K-dimension $n^2 - n$ by (ii) (note that after this choice, $u_1, \ldots, u_{n^2}$ need not be linearly independent anymore, but we don't need this in the sequel). This implies that for any $y \in A$ there exists $c \in L$ such that

$$v_n(c) = v_n(y), \ldots, v_r(c) = v_r(y).$$

Since $r < 2n^2 - 1$, there exists $0 \neq a \in A$ such that $u_{n^2}(a) = \cdots = u_r(a) = 0$. Hence, Equation (2) implies

$$\forall\, y \in A\ \exists\, c \in L: \quad ay - ac = a(y - c) \in \sum_{\rho=1}^{n-1} K w_\rho \subseteq L.$$

Thus we have $ay \in L$ for all $y \in A$, hence $aA \subseteq L$, a contradiction to (iii). □

Lafon and Winograd have proved that even $L(K^{n\times n}) \geq 2n^2 - 1$ over any field K. Their proof is beyond the scope of these notes.

Applying Theorem 4.8 to the case $n = 2$, we get $R(K^{2\times 2}) \geq 7$. On the other hand, Strassen's algorithm introduced in Example 2.17 implies $R(K^{2\times 2}) \leq 7$ which shows that $R(K^{2\times 2}) = 7$ and that Strassen's algorithm for multiplication of 2×2-matrices is optimal. For values of n different from 2, it is not known whether Theorem 4.8 is sharp.

The *asymptotic* bilinear complexity of matrix multiplication is characterized by the so-called *exponent of matrix multiplication*, usually denoted by ω_K, which is defined as

$$\omega_K := \inf\{\gamma \mid R(K^{n\times n}) = O(n^\gamma)\}.$$

Note that by Lemma 2.14 we have $R(K^{n\times n}) \geq L(K^{n\times n}) \geq R(K^{n\times n})/2$, hence $R(K^{n\times n})$ and $L(K^{n\times n})$ are asymptotically equal. It is known that ω_K at most depends on the characteristic of K [6].

Although the model of bilinear complexity neglects operations in the field of scalars, one can show that the number of all arithmetic operations for multiplication of $n \times n$-matrices is of the same order of magnitude as $R(K^{n\times n})$, i.e., if we denote by $M_K(n)$ the minimum number of arithmetic operations necessary to multiply two $n \times n$-matrices over K, then $M_K(n) = O(R(K^{n\times n}))$ (see for instance [6, p. 57–58]). Hence $\omega_K = \inf\{\gamma \mid M_K(n) = O(n^\gamma)\}$.

The trivial algorithm for multiplying matrices implies that $\omega_K \leq 3$ over any field K. Recursion applied to Strassen's algorithm for 2×2-matrix multiplication shows that $\omega_K \leq \log_2 7$ for any field. The present world record for ω_K is held by Coppersmith and Winograd [4] who have shown that $\omega_K \leq 2.38$ over any field K; Theorem 4.8 implies that $\omega_K \geq 2$ over any field K.

The complexity of many problems in linear algebra, like inversion of non-singular matrices or solving systems of linear equations is directly related to the complexity of matrix multiplication. Knowing the latter is therefore of fundamental interest.

We can now generalize the problems introduced in this section in the following way: Let K be a field and A be a finite dimensional associative algebra over K, i.e., A is a finite dimensional vector space over K endowed with a multiplication which is bilinear and associative. We consider bilinear map from $A \times A$

to A which assigns to every pair of elements in A their product, and ask for the multiplicative complexity $L(A/K)$ or the rank $R(A/K)$ of this bilinear map. (We may also write $L(A)$ or $R(A)$ if K is known from the context.) There is a general lower bound for this quantity which is due to Alder and Strassen [1]:

Theorem 4.9. (Alder-Strassen) *Let A be a finite dimensional associative algebra over K. Then*

$$L(A/K) \geq 2\dim_K(A) - t,$$

where t is the number of maximal two-sided ideals of A.

The proof of this theorem is beyond the scope of these notes. Let us apply Theorem 4.9 to the problems stated before: If A is a division algebra of dimension n over K, then the only maximal two-sided ideal of A is the zero ideal, hence $L(A/K) \geq 2n - 1$, in accordance with Theorem 4.2.

Let A be the ring of $n \times n$-matrices over K. It is an easy exercise to prove that the only maximal two-sided ideal of A is the zero ideal. Since the dimension of A over K is n^2, we obtain $L(K^{n\times n}/K) \geq 2n^2 - 1$ which is the result of Lafon and Winograd.

Another application of Theorem 4.9 is as follows:

Let G be a finite group. The group ring $\mathbf{C}[G]$ of G is defined to be the ring of all complex valued functions on G. For all elements $\sigma \in G$ we identify σ with the characteristic function of $\{\sigma\}$. Then it is clear that every element $f \in \mathbf{C}[G]$ has a unique representation as $f = \sum_{\sigma\in G} a_\sigma \sigma$ where $a_\sigma \in \mathbf{C}$. Extending the multiplication of G by linearity $\mathbf{C}[G]$ becomes $\mathbf{C}$-algebra of dimension $|G|$, where $|G|$ denotes the number of elements in G. By Wedderburn's theorem, the number of maximal two-sided ideals of $\mathbf{C}[G]$ equals the number $h(G)$ of conjugacy classes of G, hence, $L(\mathbf{C}[G]/\mathbf{C}) \geq 2|G| - h(G)$.

5. Final Remarks

These notes are meant to serve as a first introduction to the complexity theory of bilinear problems. We have tried to introduce the common language used

and the general settings of the problems discussed there. There are several connections between this theory and other branches of algebra, like algebraic geometry, coding theory, or algebraic curves, to name a few. Since we have tried to keep the notes elementary, it was even impossible to prove some of the results stated in the last section.

Nevertheless, we hope that the reader has become interested to know more about this mathematical discipline. A good book to start with is [6]. The survey articles [5, 11] give a very good insight into the different topics of algebraic complexity theory and provide a detailed list of published material about this subject.

Acknowledgment

I want to thank P. Bürgisser for helpful comments and providing me with Reference [2]. I also greatly acknowledge the grant I received through the GMD-CNPq convention by which it was possible to present these notes at the XII Escola de Álgebra in Diamantina.

References

[1] A. Alder, V. Strassen: *On the algorithmic complexity of associative algebras.* Theoretical Comp. Science, **15**, (1981), 201–211.

[2] W. Baur: *Algebraische Berechnungskomplexität.* Lectures at the Mathematics Department of the University of Konstanz, (unpublished manuscript), 1990.

[3] P. Bürgisser, M. Clausen, Th. Lickteig, M. A. Shokrollahi: *Algebraic Complexity Theory.* In preparation.

[4] D. Coppersmith, S. Winograd: *Matrix multiplication via arithmetic progressions.* Proc. 19th ACM STOC, New York, (1987), 1–6.

[5] J. von zur Gathen: *Algebraic complexity theory,* Annual Review of Computer Science, **3**, (1988), 317–347.

[6] H. F. de Groote: *Lectures on the Complexity of Bilinear Problems.* Lecture Notes in Computer Science, **245**, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1985.

[7] S. Lang: *Algebra.* Addison-Wesley, 1984

[8] A. Schönhage, V. Strassen: *Schnelle Multiplikation großer Zahlen.* Computing, **7**, (1971), 281–292.

[9] V. Strassen: *Gaussian elimination is not optimal.* Num. Math., **13**, (1969), 354–356.

[10] V. Strassen: *Vermeidung von Divisionen.* Journal für die reine und angewandte Mathematik, **264**, (1973), 184–202.

[11] V. Strassen: Algebraic complexity theory, in: *Handbook of Theoretical Computer Science,* (J. van Leeuwen, A. Meyer, M. Nivat, M. Paterson, D. Perrin, eds.), Volume A, Elsevier Science Publishers 1990, 635–672.

M. A. Shokrollahi
Institut für Informatik
Universität Bonn, Römerstr. 164
53117 Bonn, Germany