# Evaluation properties of symmetric polynomials

Pierrick Gaudry, LIX, École polytechnique
91128 Palaiseau, France
gaudry@lix.polytechnique.fr

Éric Schost, STIX, École polytechnique
91128 Palaiseau, France
schost@stix.polytechnique.fr

Nicolas M. Thiéry,
Laboratoire de Probabilités, Combinatoire et Statistiques,
Université Claude Bernard Lyon I, France
nthiery@users.sourceforge.net

October 21, 2004

**Abstract**

By the fundamental theorem of symmetric polynomials, if $P \in \mathbb{Q}[X_1, \ldots, X_n]$ is symmetric, then it can be written $P = Q(\sigma_1, \ldots, \sigma_n)$, where $\sigma_1, \ldots, \sigma_n$ are the elementary symmetric polynomials in $n$ variables, and $Q$ is in $\mathbb{Q}[S_1, \ldots, S_n]$.

We investigate the complexity properties of this construction in the straight-line program model, showing that the complexity of evaluation of $Q$ depends only on $n$ and on the complexity of evaluation of $P$.

Similar results are given for the decomposition of a general polynomial in a basis of $\mathbb{Q}[X_1, \ldots, X_n]$ seen as a module over the ring of symmetric polynomials, as well as for the computation of the Reynolds operator.

## 1 Introduction

Already known to Newton, the fundamental theorem of symmetric polynomials asserts that any symmetric polynomial is a polynomial in the elementary symmetric polynomials. To be more precise, let us define the symmetric polynomials $\overline{\sigma}_1, \ldots, \overline{\sigma}_n$ by letting $\overline{\sigma}_i$ be the coefficient of $T^{n-i}$ in the polynomial $(T - X_1) \cdots (T - X_n)$; that is, $\overline{\sigma}_i = (-1)^i \sigma_i$, where $\sigma_1, \ldots, \sigma_n$ are the usual elementary symmetric polynomials (this sign convention happens to simplify some of the subsequent developments).

Then if $P \in \mathbb{Q}[X_1, \ldots, X_n]$ is a symmetric polynomial in $n$ variables, it is known that there exists a unique polynomial $Q \in \mathbb{Q}[S_1, \ldots, S_n]$ such that the equality $P = Q(\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$ holds. For this point, as well as for other questions related to symmetric polynomials, our general reference will be [12].

From the complexity viewpoint, one may wonder what properties pass from $P$ to $Q$. For instance, the (weighted) degree is preserved. On the other hand, important features such as sparseness are lost: Consider $P = X_1^d + X_2^d \in \mathbb{Q}[X_1, X_2]$, and the polynomial $Q$ such that $P = Q(\overline{\sigma}_1, \overline{\sigma}_2)$ with $\overline{\sigma}_1 = -X_1 - X_2, \overline{\sigma}_2 = X_1 X_2$; then the number of monomials of $Q$ is linear in $d$.

This phenomenon is intimately related to the basic approach on symmetric polynomials by means of rewriting techniques. Indeed, the classical proof of the fundamental theorem involves an explicit rewriting process for a suitable elimination order [17, 5], and such techniques do not preserve sparseness.

In this note, we adopt a different point of view, working in the *straight-line program* model. Roughly speaking, a straight-line program is a sequence of basic instructions $(+, -, \times)$ that are used to compute a given polynomial; the relevant complexity measure of such an object is its *size*, *i.e.* the number of its instructions (see Subsection 2.1 for definitions). The *complexity of evaluation* of a polynomial $P$ is then the minimum size of a straight-line program that computes $P$.

Straight-line programs have proved to be an appropriate data-structure to derive complexity estimates in polynomial elimination theory (see references below). One of the salient results is that the complexity of evaluation remains stable throughout elimination processes: eliminating polynomials (e.g., Chow forms) that are obtained from polynomials with a low complexity of evaluation also have a low complexity of evaluation. This is the key to the algorithms with the best known complexity for polynomial system solving.

Our main goal is to present results in a similar vein for computations with symmetric polynomials: If $P$ is a symmetric polynomial in $\mathbb{Q}[X_1, \ldots, X_n]$ with a good complexity of evaluation, and $Q$ is such that $P = Q(\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$, then $Q$ itself has a good complexity of evaluation. The precise form of this result is given below. The statement involves a quantity denoted by $\Delta(n)$, which will be defined in Subsection 3.2 as the complexity of multiplication in a suitable algebra; for the moment, we can content ourselves with the estimate $\Delta(n) \leq 4^n (n!)^2$.

**Theorem 1.** *Let $P$ in $\mathbb{Q}[X_1, \ldots, X_n]$ be a symmetric polynomial that can be computed by a straight-line program of size $L$. Let $Q$ be the unique polynomial in $\mathbb{Q}[S_1, \ldots, S_n]$ such that $P = Q(\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$. Then $Q$ can be computed by a straight-line program of size $\Delta(n)L + 2$.*

Note that the degree of $P$ does not appear in this estimate: passing from $P$ to $Q$, the complexity of evaluation increases by a factor that only depends on $n$. As an application, consider again the polynomials $P = X_1^d + X_2^d$ and $Q$ such that $P = Q(-X_1 - X_2, X_1 X_2)$. Using binary powering techniques, $P$ can be computed by a straight-line program of size $O(\log(d))$. Theorem 1 then shows that this is also the case for $Q$; this should be compared with the number of monomials of $Q$, which is linear in $d$.

Our interest for this topic originates from [8], where a problem of solving some polynomial systems with symmetries is raised (a more general version of that question was already discussed in [4]). To solve that particular problem, the above theorem suffices. However, the proof techniques easily give more general results.

Let us write $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$ for the algebra of symmetric polynomials. Then the polynomial ring $\mathbb{Q}[X_1, \ldots, X_n]$ becomes a free module of rank $n!$ over $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$. Thus, a first generalization is to determine the coordinates of any polynomial $P$ in a basis

of this free module. The proof of Theorem 1 readily gives this generalization for a standard monomial basis, but other bases are of interest, such as, for instance, the Schubert basis. Such bases have cardinality $n!$ and are thus commonly indexed by permutations; we will then use this indexation below. Besides, the techniques can be generalized to other families of algebra generators for $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$, such as the complete symmetric polynomials and power sums.

We obtain results that generalize those of Theorem 1: roughly speaking, the complexity of evaluation only grows by the factor $\Delta(n)$, up to an additional factor that depends on the chosen bases. To give a precise statement, fix $n \geq 1$, and consider a family $\mathfrak{b} = (b_1, \ldots, b_n)$ of $\mathbb{Q}$-algebra generators of $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$ and a basis $\mathfrak{c} = (c_s)_{s \in \mathfrak{S}_n}$ of the $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-module $\mathbb{Q}[X_1, \ldots, X_n]$. The most general form of Theorem 1 involves some constants depending on $\mathfrak{b}$ and $\mathfrak{c}$, denoted by $L(\mathfrak{b})$ and $L(\mathfrak{c})$.

**Theorem 2.** *Let $n \geq 1$, $\mathfrak{b}$ and $\mathfrak{c}$ be as above. Then there exists $L(\mathfrak{b})$ and $L(\mathfrak{c})$ in $\mathbb{N}$ with the following property: Let $P$ be a polynomial in $\mathbb{Q}[X_1, \ldots, X_n]$ and let $(P_s)_{s \in \mathfrak{S}_n}$ be the unique polynomials in $\mathbb{Q}[B_1, \ldots, B_n]$ such that*

$$P = \sum_{s \in \mathfrak{S}_n} P_s(b_1, \ldots, b_n) c_s.$$

*If $P$ can be computed by a straight-line program of size $L$, then there exists a straight-line program of size $\Delta(n)L + L(\mathfrak{b}) + L(\mathfrak{c}) + 2$ which computes all the polynomials $P_s$.*

Theorem 1 is actually a particular case of this result, when $P$ is symmetric, $b_1, \ldots, b_n$ are the symmetric polynomials $\overline{\sigma}_1, \ldots, \overline{\sigma}_n$, and $\mathfrak{c}$ is the standard monomial basis; in this case we have $L(\mathfrak{b}) = L(\mathfrak{c}) = 0$ (of course, we could incorporate the term $+2$ that appears in the estimate of the theorem in either $L(\mathfrak{b})$ or $L(\mathfrak{c})$, but this would conflict with this last statement).

Our last question of interest is the computation of the Reynolds operator, which we treat as an application of the previous results. The Reynolds map $R$ is a projector $\mathbb{Q}[X_1, \ldots, X_n] \to \mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$, so for any $P$ in $\mathbb{Q}[X_1, \ldots, X_n]$, there exists $Q$ in $\mathbb{Q}[S_1, \ldots, S_n]$ such that $R(P) = Q(\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$ (other choices of algebra generators for $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$ would do as well, of course). Based on our previous results, the last theorem shows that if $P$ can be computed in time $L$, then $R(P)$ can be computed in time $\Delta(n)L$, up to about $n!$ additional operations.

**Theorem 3.** *Let $n \geq 1$, and $P$ in $\mathbb{Q}[X_1, \ldots, X_n]$ that can be computed by a straight-line program of size $L$. Let $Q$ be the unique polynomial in $\mathbb{Q}[S_1, \ldots, S_n]$ such that $R(P) = Q(\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$. Then $Q$ can be computed by a straight-line program of size $\Delta(n)L + n! + 2 \cdot 8^n$.*

In the above theorems, we used $\mathbb{Q}$ for base field; we mention however that all results extend any base ring of characteristic zero; all results that involve neither the Reynolds operator nor the power sums actually extend to any base ring.

**Related work.** Many techniques used below, notably the so-called algebra of universal decomposition and the related Cauchy modules, were already used in the study of symmetric polynomials. Explicitly, the idea of obtaining the expression of a symmetric

polynomial in terms of the elementary symmetric ones by reduction modulo what we call Cauchy modules is already present in [6, 15], and is discussed in details (without using the same denomination) in [7], together with some generalizations to other groups.

However, none of the above references mentions complexity. Our contribution is a first exploration of the complexity-related aspects of this question, showing that evaluation techniques give an appropriate computational model for handling symmetric polynomials.

It turns out that our basic algorithms are somehow relevant from polynomial elimination. Then, the fact that evaluation techniques are the key to good complexity results supports ideas initiated by Giusti, Heintz, Pardo and collaborators in [11, 10, 9], who showed that, generally speaking, straight-line programs are an appropriate data-structure for algorithms in effective elimination theory.

Of course, we expect that our results generalize to finite groups actions, even though several closed form formulas (e.g., explicit descriptions of the Cauchy modules) that are available here have probably no equivalent in the more general case. Then, we might have to rely on effective elimination theory tools.

**Optimal bounds.** At the moment, we do not know whether the factor $\Delta(n)$, which grows polynomially with $n!$, is optimal. To put it more precisely, let us write $L(A)$ for the minimal size of a straight-line program that computes a polynomial $A$. Let then $\delta(n)$ be the supremum of the ratios $L(Q)/L(P)$, where $P$ runs through the symmetric polynomials in $\mathbb{Q}[X_1, \ldots, X_n]$ and $Q$ is such that $P = Q(\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$. Theorem 1 shows that $\delta(n) \leq \Delta(n) \in (n!)^{O(1)}$; an open question is to give a non-trivial lower bound for $\delta(n)$.

**Organization of the paper.** In Section 2, we define our computational model, and give the details of our construction on the example $P = X_1^d + X_2^d$. In Section 3, we give the proofs of Theorems 1, 2 and 3.

# 2 Preliminaries

## 2.1 Straight-line programs

From an informal point of view, straight-line programs enable us to represent polynomials by means of a sequence of operations $(+, -, \times)$ without test nor division. Formally, let $k$ be a field and $L \geq 0, n \geq 1$. Following [3], we define a straight-line program $\Gamma$ in $k[X_1, \ldots, X_n]$ as a sequence of polynomials $G_{-n+1}, \ldots, G_L$ in $k[X_1, \ldots, X_n]$. For $-n+1 \leq i \leq 0$, we take $G_i = X_{i+n}$; for $i > 0$, suppose that $G_{-n+1}, \ldots, G_{i-1}$ are defined. Then, we require that one of the following holds:

- $G_i = \lambda$, with $\lambda \in k$.
- $G_i = \lambda + G_{a_i}$, $G_i = \lambda - G_{a_i}$ or $G_i = \lambda G_{a_i}$, with in any case $\lambda \in k$ and $-n+1 \leq a_i < i$.
- $G_i = G_{a_i} + G_{b_i}$, $G_i = G_{a_i} - G_{b_i}$ or $G_i = G_{a_i} G_{b_i}$, with in any case $-n+1 \leq a_i, b_i < i$.

In this situation, we say that $\Gamma$ *computes* $G_{-n+1}, \ldots, G_L$ and has *size* $L$. If $F_1, \ldots, F_m$ are polynomials in $k[X_1, \ldots, X_n]$, then we say that $F_1, \ldots, F_m$ can be *computed by a*

*straight-line program of size $L$* (or *in time $L$*) if there exists a straight-line program that computes polynomials $G_{-n+1}, \ldots, G_L$ such that $\{F_1, \ldots, F_m\}$ is included in the set $\{G_{-n+1}, \ldots, G_L\}$.

The following lemma gives a basic property of the straight-line model, which is useful in the sequel.

**Lemma 1 (Composition of straight-line programs).** *Let $\mathbf{a} = (a_1, \ldots, a_n)$ be polynomials in $k[X_1, \ldots, X_m]$ and let $\mathbf{b} = (b_1, \ldots, b_m)$ be polynomials in $k[Y_1, \ldots, Y_s]$. If $\mathbf{a}$ (resp. $\mathbf{b}$) can be computed in time $L_{\mathbf{a}}$ (resp. in time $L_{\mathbf{b}}$), then $a_1(b_1, \ldots, b_m), \ldots, a_n(b_1, \ldots, b_m)$ can be computed in time $L_{\mathbf{a}} + L_{\mathbf{b}}$.*

*Proof.* Let $\Gamma$ (resp. $\Lambda$) be a straight-line program of size $L_{\mathbf{a}}$ (resp $L_{\mathbf{b}}$) which computes $\mathbf{a}$ (resp. $\mathbf{b}$), and let $G_{-m+1}, \ldots, G_{L_{\mathbf{a}}}$ (resp. $H_{-s+1}, \ldots, H_{L_{\mathbf{b}}}$) be the associated polynomials. For $i = 1, \ldots, L_{\mathbf{a}}$, define $K_i = G_i(b_1, \ldots, b_m)$. Then the sequence $H_{-s+1}, \ldots, H_{L_{\mathbf{b}}}, K_1, \ldots, K_{L_{\mathbf{a}}}$ satisfies our requirement. □

## 2.2 A detailed example

We now show the use of the straight-line program representation for handling symmetric polynomials, by computing the symmetrized form of the polynomial $P = X_1^8 + X_2^8 \in \mathbb{Q}[X_1, X_2]$. Let us thus consider a sequence of instructions that computes $P$:

$$G_1 = X_1^2; \quad G_2 = G_1^2; \quad G_3 = G_2^2; \quad H_1 = X_2^2; \quad H_2 = H_1^2; \quad H_3 = H_2^2;$$

so that $P = G_3 + H_3 = X_1^8 + X_2^8$. We now show how to compute the unique polynomial $Q \in \mathbb{Q}[S_1, S_2]$ such that $P = Q(-X_1 - X_2, X_1X_2)$.

Let us introduce two new indeterminates $S_1$ and $S_2$ and the ideal $I$ generated by $S_1 - (-X_1 - X_2)$ and $S_2 - X_1X_2$ in $\mathbb{Q}[S_1, S_2][X_1, X_2]$. Our strategy is to compute the coordinates of the polynomials $G_i$ and $H_i$ in the $\mathbb{Q}[S_1, S_2]$-algebra $K = \mathbb{Q}[S_1, S_2][X_1, X_2]/I$. From this, we will recover the polynomial $Q$.

The monomials $(1, X_1)$ form a basis of $K$ as a $\mathbb{Q}[S_1, S_2]$-algebra and the relation $X_1^2 + S_1X_1 + S_2 = 0$ holds in $K$. We deduce that for all $A_0, A_1, B_0, B_1$ in $\mathbb{Q}[S_1, S_2]$, the multiplication law in $K$ is given by the following rule:

$$(A_0 + A_1X_1)(B_0 + B_1X_1) = (A_0B_0 - S_2A_1B_1) + (A_1B_0 + A_0B_1 - S_1A_1B_1)X_1.$$

This multiplication can be written using the following straight-line program $\Gamma$, which uses Karatsuba's trick to lower the number of multiplications:

$$\Gamma \left| \begin{array}{llll} V_1 = A_0B_0; & V_2 = A_1B_1; & V_3 = A_0 + A_1; & V_4 = B_0 + B_1; \\ V_5 = V_3V_4; & V_6 = V_5 - V_1; & V_7 = V_6 - V_2; & V_8 = S_2V_2; \\ V_9 = -S_1V_2; & V_{10} = V_1 - V_8; & V_{11} = V_7 + V_9. & \end{array} \right.$$

Then $V_{10}$ and $V_{11}$ are respectively the polynomials $(A_0B_0 - S_2A_1B_1)$ and $(A_1B_0 + A_0B_1 - S_1A_1B_1)$; note that $\Gamma$ performs 11 operations.

For $i = 1, 2, 3$, let us write $G_i \bmod I = G_{i,0} + G_{i,1}X_1$, with $G_{i,0}$ and $G_{i,1}$ in $\mathbb{Q}[S_1, S_2]$. Using $\Gamma$, we first design a straight-line program that computes the polynomials $G_{i,0}$

and $G_{i,1}$, for $i = 1, 2, 3$. To this effect, let us take $G_{0,0} = 0$ and $G_{0,1} = 1$, so that $G_{0,0} + G_{0,1}X_1 = X_1$. Since $G_1 = X_1^2$, we can adapt $\Gamma$ to obtain $G_{1,0}$ and $G_{1,1}$:

$$\begin{array}{llll} V_{1,1} = 0; & V_{1,2} = 1; & V_{1,3} = 1; & V_{1,4} = 1; \\ V_{1,5} = V_{1,3}V_{1,4}; & V_{1,6} = V_{1,5} - V_{1,1}; & V_{1,7} = V_{1,6} - V_{1,2}; & V_{1,8} = S_2V_{1,2}; \\ V_{1,9} = -S_1V_{1,2}; & G_{1,0} = V_{1,1} - V_{1,8}; & G_{1,1} = V_{1,7} + V_{1,9}. & \end{array}$$

Iterating the process, we obtain $G_{2,0}, G_{2,1}$ and $G_{3,0}, G_{3,1}$ in a similar fashion.

$$\begin{array}{llll} V_{2,1} = G_{1,0}^2; & V_{2,2} = G_{1,1}^2; & V_{2,3} = G_{1,0} + G_{1,1}; & V_{2,4} = G_{1,0} + G_{1,1}; \\ V_{2,5} = V_{2,3}V_{2,4}; & V_{2,6} = V_{2,5} - V_{2,1}; & V_{2,7} = V_{2,6} - V_{2,2}; & V_{2,8} = S_2V_{2,2}; \\ V_{2,9} = -S_1V_{2,2}; & G_{2,0} = V_{2,1} - V_{2,8}; & G_{2,1} = V_{2,7} + V_{2,9}; & \end{array}$$

$$\begin{array}{llll} V_{3,1} = G_{2,0}^2; & V_{3,2} = G_{2,1}^2; & V_{3,3} = G_{2,0} + G_{2,1}; & V_{3,4} = G_{2,0} + G_{2,1}; \\ V_{3,5} = V_{3,3}V_{3,4}; & V_{3,6} = V_{3,5} - V_{3,1}; & V_{3,7} = V_{3,6} - V_{3,2}; & V_{3,8} = S_2V_{3,2}; \\ V_{3,9} = -S_1V_{3,2}; & G_{3,0} = V_{3,1} - V_{3,8}; & G_{3,1} = V_{3,7} + V_{3,9}. & \end{array}$$

We are now almost done: we have obtained polynomials $G_{3,0}$ and $G_{3,1}$ in $\mathbb{Q}[S_1, S_2]$ such that $G_3 = X_1^8 = G_{3,0} + G_{3,1}X_1$ holds modulo $I$. Starting from $X_2 = -S_1 - X_1$, we can use the same techniques to obtain polynomials $H_{3,0}$ and $H_{3,1}$ such that $H_3 = X_2^8 = H_{3,0} + H_{3,1}X_1$ holds modulo $I$. The sum $G_3 + H_3$ being symmetric, it equals $G_{3,0} + H_{3,0}$ modulo $I$, so that $G_{3,0} + H_{3,0}$ is the polynomial $Q$ we are looking for.

Computing $Q$ requires $2 \times 3 \times 11 + 3 = 69$ operations $(+, -, \times)$. Had we considered the polynomial $X_1^{16} + X_2^{16}$ instead, the cost would be 91, due to an additional squaring in $K$. Similarly, treating the polynomial $X_1^{2^k} + X_2^{2^k}$ would require $22k + 3$ instructions. In particular, in view of Waring's formula, this shows that given $X_1 + X_2$ and $X_1X_2$, one can evaluate the sum

$$X_1^d + X_2^d = \sum_{j=0}^{\lfloor d/2 \rfloor} (-1)^j \frac{d}{d-j} \binom{n-j}{j} (X_1X_2)^j (X_1 + X_2)^{d-2j}$$

within $O(\log(d))$ arithmetic operations, whereas the sum has a number of terms linear in $d$.

The following section provides with a generalization of this process; Subsection 3.5 also shows how to save a constant factor (here, that would be 2), using the Reynolds operator.

# 3 Proof of the main results

Let us denote by $E$ the set of multi-indices

$$E = \{\alpha = (\alpha_1, \ldots, \alpha_n) \mid 0 \leq \alpha_i < n - i + 1,\ 1 \leq i \leq n\};$$

then the set of monomials $X^E = \{X^\alpha = X_1^{\alpha_1} \cdots X_n^{\alpha_n} \mid \alpha = (\alpha_1, \ldots, \alpha_n) \in E\}$ form a basis of the $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-module $\mathbb{Q}[X_1, \ldots, X_n]$, which we call the *standard monomial basis*. The key result of this section is the following proposition, which contains Theorem 1 as a special case when $P$ is symmetric, and is the basis to Theorems 2 and 3. The statement involves the quantity $\Delta(n)$, which is defined below in Subsection 3.2.

**Proposition 1.** *Let $P$ be in $\mathbb{Q}[X_1, \ldots, X_n]$ and let $(P_\alpha)_{\alpha \in E}$ be the unique polynomials in $\mathbb{Q}[S_1, \ldots, S_n]$ such that the equality*

$$P = \sum_{\alpha \in E} P_\alpha(\overline{\sigma}_1, \ldots, \overline{\sigma}_n) X_1^{\alpha_1} \cdots X_n^{\alpha_n} \tag{1}$$

*holds. Suppose that $P$ can be computed by a straight-line program of size $L$. Then there exists a straight-line program of size $\Delta(n)L+2$ which computes all the polynomials $(P_\alpha)_{\alpha \in E}$.*

The basic setting of the proof, the so-called algebra of universal decomposition, is introduced in Subsection 3.1; the notation $\Delta(n)$ is then defined in Subsection 3.2. The proof of Proposition 1 is given in Subsection 3.3; in Subsection 3.4 and 3.5, we finally deduce Theorems 2 and 3 as corollaries.

## 3.1 The Cauchy modules

To make effective the $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-algebra structure on $\mathbb{Q}[X_1, \ldots, X_n]$, we consider two sets of indeterminates $S_1, \ldots, S_n$ and $X_1, \ldots, X_n$ and work in the polynomial ring $\mathbb{Q}[S_1, \ldots, S_n][X_1, \ldots, X_n]$, taking $\mathbb{Q}[S_1, \ldots, S_n]$ for base ring. We then introduce the following polynomials in $\mathbb{Q}[S_1, \ldots, S_n][X_1, \ldots, X_n]$:

$$F_i \ : S_i - \overline{\sigma}_i(X_1, \ldots, X_n), \quad i = 1, \ldots, n.$$

Let $I$ be the ideal $(F_1, \ldots, F_n) \subset \mathbb{Q}[S_1, \ldots, S_n][X_1, \ldots, X_n]$. Then, the quotient $K = \mathbb{Q}[S_1, \ldots, S_n][X_1, \ldots, X_n]/I$ is a free $\mathbb{Q}[S_1, \ldots, S_n]$-algebra of rank $n!$, called the universal decomposition algebra in [1] (see also [6]), and which is isomorphic to the $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-algebra $\mathbb{Q}[X_1, \ldots, X_n]$.

We next define a family of polynomials $(T_1, \ldots, T_n)$ which are better suited to computations. We first set

$$T_1(X_1) = X_1^n + S_1 X_1^{n-1} + \cdots + S_{n-1} X_1 + S_n.$$

We then inductively define $T_2, \ldots, T_n$ by the following "divided difference" relation:

$$T_{i+1}(X_1, \ldots, X_{i+1}) = \frac{T_i(X_1, \ldots, X_{i-1}, X_{i+1}) - T_i(X_1, \ldots, X_{i-1}, X_i)}{X_{i+1} - X_i}, \quad 1 \leq i < n.$$

It is immediate to check that, for $1 \leq i \leq n$, $T_i$ belongs to $\mathbb{Q}[S_1, \ldots, S_n][X_1, \ldots, X_i]$ and is monic of degree $n - i + 1$ in $X_i$. These polynomials are sometimes called the *Cauchy modules*, see [6, 15]; see also [13, 12] for more on divided differences.

The ideal $I = (F_1, \ldots, F_n)$ equals $(T_1, \ldots, T_n)$, see [6, Theorem 6]. Furthermore, as mentioned above, the standard monomials $X^E$ form a $\mathbb{Q}[S_1, \ldots, S_n]$-basis of $K$. We deduce that the polynomials $P_\alpha$ given in Equation (1) are also characterized by the relation

$$P \bmod (T_1, \ldots, T_n) = \sum_{\alpha \in E} P_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}.$$

This decomposition is the basis of all following developments.

## 3.2 Complexity of the multiplication in $K$

We can now define the quantity $\Delta(n)$, as a mean to estimate the complexity of the multiplication in $K$. Let

$$\mathfrak{A} = \sum_{\alpha \in E} A_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}, \quad \mathfrak{B} = \sum_{\alpha \in E} B_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}$$

be two polynomials with new indeterminates $\mathbf{A} = (A_\alpha)_{\alpha \in E}$ and $\mathbf{B} = (B_\alpha)_{\alpha \in E}$ as respective coefficients. Thus, the base ring is now $\mathbb{Q}[\mathbf{A}, \mathbf{B}, \mathbf{S}]$, where we write for conciseness $\mathbf{S} = S_1, \ldots, S_n$, and $\mathfrak{A}$ and $\mathfrak{B}$ are in $\mathbb{Q}[\mathbf{A}, \mathbf{B}, \mathbf{S}][X_1, \ldots, X_n]$. There exist unique polynomials $\mathbf{C} = (C_\alpha)_{\alpha \in E}$ in $\mathbb{Q}[\mathbf{A}, \mathbf{B}, \mathbf{S}]$ such that the following equality holds in $\mathbb{Q}[\mathbf{A}, \mathbf{B}, \mathbf{S}][X_1, \ldots, X_n]$:

$$\mathfrak{A}\mathfrak{B} \bmod (T_1, \ldots, T_n) = \sum_{\alpha \in E} C_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}.$$

The cost $\Delta(n)$ of the multiplication modulo $(T_1, \ldots, T_n)$ is formally defined as the minimal size of a straight-line program that computes the polynomials $\mathbf{C}$; note in particular that $\Delta(n) \geq n!$. The example of Subsection 2.2 is a particular case of this construction, which showed that $\Delta(2) \leq 11$.

The following lemma then gives the basic way to make use of this notion.

**Lemma 2.** *Let $\mathbf{a} = (a_\alpha)_{\alpha \in E}$ and $\mathbf{b} = (b_\alpha)_{\alpha \in E}$ be in $\mathbb{Q}[S_1, \ldots, S_n]$ and write*

$$\mathfrak{a} = \sum_{\alpha \in E} a_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}, \quad \mathfrak{b} = \sum_{\alpha \in E} b_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}.$$

*Let us define the polynomials $\mathbf{c} = (c_\alpha)_{\alpha \in E}$ in $\mathbb{Q}[S_1, \ldots, S_n]$ by:*

$$\mathfrak{a}\mathfrak{b} \bmod (T_1, \ldots, T_n) = \sum_{\alpha \in E} c_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}.$$

*If both families of polynomials $\mathbf{a}$ and $\mathbf{b}$ can be computed by a straight-line program $\Gamma$ of size $L$, then there exists a straight-line program of size $L + \Delta(n)$ that computes the same polynomials as $\Gamma$ as well as the polynomials $\mathbf{c}$.*

*Proof.* The polynomials $\mathbf{c}$ are obtained by evaluating the polynomials $\mathbf{C}$ at $(\mathbf{a}, \mathbf{b}, \mathbf{S})$; thus, by Lemma 1, their complexity of evaluation is at most the sum of those of $\mathbf{C}$, $\mathbf{a}$ and $\mathbf{b}$. □

We shall now give estimates for the function $\Delta$. Our first estimate relies on an iterated version of Euclidean division:

**Proposition 2.** *For $n \geq 1$, the inequality $\Delta(n) \leq 4^n (n!)^2$ holds.*

We will use the fact that the coefficients of the polynomials $T_1, \ldots, T_n$ have a low complexity of evaluation: see for instance [17, Theorem 1.2.7] for a proof of the following lemma (which actually gives a more precise statement).

**Lemma 3.** *All coefficients of all polynomials $T_1, \ldots, T_n$ are in $\{1, S_1, S_2, \ldots, S_n\}$.*

*Proof of the proposition.* The proof is an inductive process; to prepare the induction, for $1 \leq i \leq n$, we define the quotient $K_i = \mathbb{Q}[\mathbf{S}][X_1, \ldots, X_i]/(T_1, \ldots, T_i)$, so that $K_n$ is the quotient $K$ defined above. To simplify the notation, for $1 \leq i \leq n$, denote $d_i = \deg_{X_i} T_i = n - i + 1$. Then for all $i$, $K_i$ is a free $\mathbb{Q}[\mathbf{S}]$-algebra of rank $d_1 \cdots d_i$. Let $E_i$ be the set

$$\{\alpha = (\alpha_1, \ldots, \alpha_i) \mid 0 \leq \alpha_j < d_j,\ 1 \leq j \leq i\};$$

then the set of all monomials $\{X_1^{\alpha_1} \cdots X_i^{\alpha_i} \mid \alpha = (\alpha_1, \ldots, \alpha_i) \in E_i\}$ is a $\mathbb{Q}[\mathbf{S}]$-basis of $K_i$.

For any $i \in 1, \ldots, n$, let $\mathbf{A}_i = (A_\alpha)_{\alpha \in E_i}$ and $\mathbf{B}_i = (B_\alpha)_{\alpha \in E_i}$ be some indeterminates. Using them as coefficients, we define

$$\mathfrak{A}_i = \sum_{\alpha \in E_i} A_\alpha X_1^{\alpha_1} \cdots X_i^{\alpha_i}, \quad \mathfrak{B}_i = \sum_{\alpha \in E_i} B_\alpha X_1^{\alpha_1} \cdots X_i^{\alpha_i}$$

and the polynomials $\mathbf{C}_i = (C_\alpha)_{\alpha \in E_i}$ in $\mathbb{Q}[\mathbf{A}_i, \mathbf{B}_i, \mathbf{S}]$ as follows:

$$\mathfrak{A}_i \mathfrak{B}_i \bmod (T_1, \ldots, T_i) = \sum_{\alpha \in E_i} C_\alpha X_1^{\alpha_1} \cdots X_i^{\alpha_i}.$$

We now establish the following claim: *There exists a straight-line program of size* $4^i (d_1 \cdots d_i)^2$ *that computes the polynomials* $\mathbf{C}_i$. Taking $i = n$ proves the lemma.

The proof of this claim comes by induction. First, let us consider $i = 1$, and let $\mathfrak{A}_1$ and $\mathfrak{B}_1$ be as above. We first multiply $\mathfrak{A}_1$ and $\mathfrak{B}_1$ as plain polynomials in $\mathbb{Q}[\mathbf{S}][X_1]$: this requires to perform $2d_1^2$ operations on their coefficients [18], either additions or multiplications. In a second time, we reduce the product $\mathfrak{A}_1 \mathfrak{B}_1$ modulo $T_1$; this requires $2d_1^2$ additional operations, involving the coefficients of both $\mathfrak{A}_1 \mathfrak{B}_1$ and $T_1$, see again [18]. This prove our claim for $i = 1$.

Let us now perform the inductive step: we assume that the induction claim holds for index $i - 1$, and prove it for index $i$. Let thus $\mathfrak{A}_i$ and $\mathfrak{B}_i$ be as above; we now estimate the cost of computing all coefficients of their product modulo $(T_1, \ldots, T_i)$.

To this effect, we first consider $\mathfrak{A}_i$ and $\mathfrak{B}_i$ as univariate polynomials in $K_{i-1}[X_i]$, and multiply them as such. This requires $2d_i^2$ operations in $K_{i-1}$, either additions or multiplications. An addition in $K_{i-1}$ requires to perform $d_1 \cdots d_{i-1}$ operations on the coefficients. Using an analogue of Lemma 2 for the quotient $K_i$, we deduce that a multiplication in $K_{i-1}$ requires at most $4^{i-1}(d_1 \cdots d_{i-1})^2$ operations. Thus, all coefficients of the product $\mathfrak{A}_i \mathfrak{B}_i$ in $K_{i-1}[X_i]$ can be evaluated using at most $2 \times 4^{i-1}(d_1 \cdots d_i)^2$.

We finally reduce $\mathfrak{A}_i \mathfrak{B}_i$ modulo $T_i$. Thus, we must see $T_i$ as a polynomial in $K_{i-1}[X_i]$; note that all coefficients of $T_i$ are already reduced modulo $(T_1, \ldots, T_{i-1})$, so no additional reduction is needed. Then reducing $\mathfrak{A}_i \mathfrak{B}_i$ modulo $T_i$ requires $2d_i^2$ operations in $K_{i-1}$, involving the coefficients of $\mathfrak{A}_i \mathfrak{B}_i$ and $T_i$. Due to Lemma 3, no operation is required to compute the coefficients of $T_i$, so we deduce as above that each operation takes at most $4^{i-1}(d_1 \cdots d_{i-1})^2$ operations, concluding the proof. □

To conclude this subsection, we mention some improved bounds for the function $\Delta$. To this effect, let us introduce the function $\mathcal{M} : \mathbb{N} \to \mathbb{N}$, such that $\mathcal{M}(d)$ is the complexity of univariate polynomial multiplication in degree $d$ (the precise definition is similar to that of the function $\Delta$ above). Thus, $\mathcal{M}(d) = 2d^2$ for the naive multiplication algorithm, but one can take $\mathcal{M}(d) \in O(d \log d \log \log d)$ using FFT multiplication [18].

Using this notation, one can prove the following result: *there exists a universal constant $C$ such that $\Delta(n) \in O(C^n \mathcal{M}(n!))$*. We do not offer a proof of this claim, as it is rather technical, using Kronecker's substitution to reduce to operations on univariate polynomials. The closest reference we are aware of is Lemma 2.2 in [19], where a similar question is treated in the case of $n = 2$ variables.

## 3.3 Proof of Proposition 1

We can now give the proof of Proposition 1. To this effect, recall that the polynomials $P_\alpha$ in Equation (1) can also be defined as the unique polynomials in $\mathbb{Q}[S_1, \ldots, S_n]$ satisfying the equality

$$P \bmod (T_1, \ldots, T_n) = \sum_{\alpha \in E} P_\alpha X_1^{\alpha_1} \cdots X_n^{\alpha_n}$$

in $\mathbb{Q}[S_1, \ldots, S_n][X_1, \ldots, X_n]$.

First, we note that the coefficients of all variables $X_i$ on the standard monomial basis of $K$ can be computed by a straight-line program of size 2. Indeed, only the variable $X_n$ rewrites non-trivially: since $T_n = X_1 + \cdots + X_n + S_1$ (see [17]), $X_n$ rewrites as $-S_1 - X_1 - \cdots - X_{n-1}$. Thus, we need only compute $-1$ and $-S_1$.

Let now $\Gamma$ be a straight-line program of size $L$ that computes $P$. We suppose that $\Gamma$ computes the sequence of polynomials $G_{-n+1}, \ldots, G_L$ in $\mathbb{Q}[X_1, \ldots, X_n]$, where for $-n + 1 \leq i \leq 0$, $G_i = X_{i+n}$. We define the polynomials $G_{i,\alpha}$ as the unique polynomials in $\mathbb{Q}[S_1, \ldots, S_n]$ such that the equalities

$$G_i \bmod (T_1, \ldots, T_n) = \sum_{\alpha \in E} G_{i,\alpha} X_1^{\alpha_1} \cdots X_n^{\alpha_n}$$

hold in $\mathbb{Q}[S_1, \ldots, S_n][X_1, \ldots, X_n]$, for $-n + 1 \leq i \leq L$. Using an induction on $L$, we conclude by proving that *all polynomials $G_{i,\alpha}$, can be computed by a straight-line program of size $\Delta(n)L + 2$*. This is enough to prove Proposition 1.

If $L = 0$, then the construction of the above paragraph concludes. Suppose now that $\Gamma$ has size $L+1$, and let $\Gamma'$ be the straight-line program made by keeping only the first $L$ operations of $\Gamma$. Then by the induction assumption, there exists a straight-line program of size $\Delta(n)L + 2$ that computes the coefficients $G_{i,\alpha}$ for $-n + 1 \leq i \leq L$. By definition, the polynomial $G_{L+1}$ takes one of the following forms:

1. $G_{L+1} = \lambda$, with $\lambda \in \mathbb{Q}$;
2. $G_{L+1} = \lambda + G_{a_i}$, $G_{L+1} = \lambda - G_{a_i}$ or $G_{L+1} = \lambda G_{a_i}$, with $-n + 1 \leq a_{L+1} \leq L$ and $\lambda \in \mathbb{Q}$;
3. $G_{L+1} = G_{a_{L+1}} + G_{b_{L+1}}$, $G_{L+1} = G_{a_{L+1}} - G_{b_{L+1}}$ or $G_{L+1} = G_{a_{L+1}} G_{b_{L+1}}$, with $-n+1 \leq a_{L+1}, b_{L+1} \leq L$.

The non-trivial case of the multiplication $G_{L+1} = G_{a_{L+1}} G_{b_{L+1}}$ is handled by Lemma 2; all others are immediate.

## 3.4 Proof of Theorem 2

To complete the proof of Theorem 2, we rely on Proposition 1 as an intermediate result; then it suffices to estimate the overhead induced by the base changes.

Let $\mathfrak{b} = (b_1, \ldots, b_n)$ be $\mathbb{Q}$-algebra generators of $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$, and let $\mathfrak{c} = (c_s)_{s \in \mathfrak{S}_n}$ be a basis of the free $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-module $\mathbb{Q}[X_1, \ldots, X_n]$. Then any polynomial $P$ can be written uniquely as

$$P = \sum_{s \in \mathfrak{S}_n} P_s(b_1, \ldots, b_n) c_s; \tag{2}$$

we now relate the complexity of evaluation of $P$ to that of the polynomials $P_s$. To this effect, we introduce two alternative representations of $P$, by means of some polynomials $\overline{P_\alpha}$ and $\widetilde{P_s}$:

$$P = \sum_{\alpha \in E} \overline{P_\alpha}(\overline{\sigma}_1, \ldots, \overline{\sigma}_n) X_1^{\alpha_1} \cdots X_n^{\alpha_n}, \quad P = \sum_{s \in \mathfrak{S}_n} \widetilde{P_s}(\overline{\sigma}_1, \ldots, \overline{\sigma}_n) c_s. \tag{3}$$

Proposition 1 relates the complexity of evaluation of $P$ to that of the polynomials $\overline{P_\alpha}$. Using this result, we will first estimate the complexity of the polynomials $\widetilde{P_s}$, and then deduce that of the polynomials $P_s$.

**Change of $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-module basis.** Let $M$ be the $n! \times n!$ matrix of change of basis from the standard monomial basis into $\mathfrak{c}$; the coefficients of this matrix are symmetric polynomials, which we choose to represent as polynomials in $\overline{\sigma}_1, \ldots, \overline{\sigma}_n$. Then the polynomials $\widetilde{P_s}$ are obtained from the polynomials $\overline{P_\alpha}$ by matrix-vector multiplication with the matrix $M$.

Let $\ell(\mathfrak{c})$ be the size of a straight-line program that evaluates all entries of $M$. Proposition 1 states that if $P$ can be computed in time $L$, then all polynomials $\overline{P_\alpha}$ can be computed in time $\Delta(n)L + 2$. Thus, the polynomials $\widetilde{P_s}$ can be computed in time $\Delta(n)L + 2 + \ell(\mathfrak{c}) + n!(2n! - 1)$, where the last term is the cost for matrix-vector product in size $n! \times n!$. Thus, the quantity $L(\mathfrak{c}) = \ell(\mathfrak{c}) + n!(2n! - 1)$ satisfies the claim of Theorem 2 relative to $\mathfrak{c}$.

Note that if $M$ has entries in $\mathbb{Q}$, only the term $n!(2n! - 1)$ remains; if $\mathfrak{c}$ is the standard monomial basis, then $M$ is the identity matrix, and we can completely dispense with the term $n!(2n! - 1)$, so that $L(\mathfrak{c})$ can be taken 0 in this case.

As an example, we evaluate $L(\mathfrak{c})$ in the case of the Schubert basis, introduced in [14]. Schubert polynomials are naturally indexed by $\mathfrak{S}_n$. We first define the permutation $\delta = [n, n-1, \ldots, 1]$, and the associated Schubert polynomial $\mathbb{X}_\delta = X_1^{n-1} X_2^{n-2} \cdots X_{n-1}$. Next, to any elementary transposition $\tau_i = (i, i+1)$ of $\mathfrak{S}_n$, we associate the divided difference operator $\partial_i$ which maps $f \in \mathbb{Q}[X_1, \ldots, X_n]$ to $(f - \tau_i f)/(x_i - x_{i+1})$. Let now $s$ be any permutation, and consider a decomposition of $s^{-1}\delta$ as a product $\partial_{i_1} \cdots \partial_{i_r}$ of elementary transpositions. The *Schubert polynomial* $\mathbb{X}_s$ is defined as $\partial_{i_1} \cdots \partial_{i_r}(\mathbb{X}_\delta)$; it happens to be independent of the choice of the factorization.

The Schubert polynomials form a basis of the $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-module $\mathbb{Q}[X_1, \ldots, X_n]$; their definition shows that the matrix that expresses the Schubert polynomials in terms of the standard monomial basis has integer entries; it is triangular, with $\{0, 1\}$ entries, and its inverse matrix $M$ also has integer entries. Thus, $L((\mathbb{X}_s)_{s \in \mathfrak{S}_n})$ is at most $n!(2n! - 1)$. However, $M$ being a very sparse matrix, we expect that a much better bound could be found.

**Change of algebra generators.** We now conclude the proof of Theorem 2. Since $\mathfrak{b}$ forms a basis of the algebra of symmetric polynomials, there exist unique polynomials $\mathcal{S}_1, \ldots, \mathcal{S}_n$ in $\mathbb{Q}[B_1, \ldots, B_n]$ such that $\overline{\sigma}_i = \mathcal{S}_i(b_1, \ldots, b_n)$ for $i = 1, \ldots, n$. Then comparing Equations (2) and (3), we deduce that $P_s = \widetilde{P_s}(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ for all $s$.

Let then $L(\mathfrak{b})$ be the size of a straight-line program $\Gamma$ that evaluates $\mathcal{S}_1, \ldots, \mathcal{S}_n$. By the previous paragraphs, we know that the polynomials $\widetilde{P_s}$ can be computed in time $\Delta(n) + L(\mathfrak{c}) + 2$. By composition with $\Gamma$, we deduce that the polynomials $P_s$ can be computed in time $\Delta(n) + L(\mathfrak{c}) + L(\mathfrak{b}) + 2$. This concludes the proof of Theorem 2.

We illustrate this construction with two well-known examples, the *complete symmetric polynomials* $\mathfrak{h} = (h_1, \ldots, h_n)$ and the *symmetric power sums* $\mathfrak{p} = (p_1, \ldots, p_n)$, which are respectively defined by:

$$h_i = \sum_{\alpha_1 + \cdots + \alpha_n = i} X_1^{\alpha_1} \cdots X_n^{\alpha_n} \quad \text{and} \quad p_i = \sum_{j=1}^{n} X_j^i;$$

note that $h_i$ and $p_i$ are actually defined for any $i \geq 0$.

We now give bounds on $L(\mathfrak{h})$ and $L(\mathfrak{p})$. To this effect, one could use the Newton relations; however, better can be done. We let $\mathcal{M} : \mathbb{N} \to \mathbb{N}$ denote the complexity of multiplying univariate polynomials (see Subsection 3.2).

**Lemma 4.** *We have $L(\mathfrak{h}) \in O(\mathcal{M}(n))$ and $L(\mathfrak{p}) \in O(\mathcal{M}(n))$.*

*Proof.* Recall that the symmetric polynomials $\overline{\sigma}$, $\mathfrak{h}$, and $\mathfrak{p}$ can be encoded via their respective generating series in $\mathbb{Q}[X_1, \ldots, X_n][[z]]$ (where we write $\overline{\sigma}_0 = 1$):

$$S(z) = \sum_{i=0}^{n} \overline{\sigma}_i z^i = \prod_{i=1}^{n} (1 - X_i z).$$

$$H(z) = \sum_{i \geq 0} h_i z^i = \prod_{i=1}^{n} \frac{1}{1 - X_i z}$$

$$P(z) = \sum_{i \geq 1} \frac{p_i}{i} z^i.$$

These generating series satisfy the relations:

$$S(z) = \frac{1}{C(z)}; \quad S(z) = \exp(-P(z)).$$

Hence, using Newton iteration for inverse and exponential of power series [2, 16, 18], one can compute the first $n + 1$ coefficients of $S(z)$ from those of $H(z)$ or $P(z)$ by a straight-line program of size $O(\mathcal{M}(n))$. □

## 3.5 Reynolds operator

The Reynolds operator is a $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-linear projection $\mathbb{Q}[X_1, \ldots, X_n] \to \mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$, and as such, is a quite important tool in the study of symmetric polynomials. For $P$ in $\mathbb{Q}[X_1, \ldots, X_n]$, $R(P)$ is given by the formula

$$R(P) = \frac{1}{n!} \sum_{s \in \mathfrak{S}_n} s \cdot P = \frac{1}{n!} \sum_{s \in \mathfrak{S}_n} P\left(X_{s(1)}, \ldots, X_{s(n)}\right).$$

Since $R(P)$ is symmetric, there exists a unique polynomial $Q \in \mathbb{Q}[S_1, \ldots, S_n]$ such that $R(P) = Q(\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$: our goal is now to relate the complexity of evaluation of $Q$ to that of $P$.

A brute-force use of the definition would consist in applying Proposition 1 to all conjugates of $P$; this would induce a loss of a factor $n!$ in complexity. Luckily enough, one can essentially read off a straight-line program for $Q$ from the straight-line program giving the coefficients $P_\alpha$ of $P$ on the standard monomial basis. Indeed, since $R$ is a $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$-module morphism, we have, writing for short $X^\alpha = X_1^{\alpha_1} \cdots X_n^{\alpha_n}$:

$$R(P) = R\left(\sum_{\alpha \in E} P_\alpha(\overline{\sigma}_1, \ldots, \overline{\sigma}_n) X^\alpha\right) = \sum_{\alpha \in E} P_\alpha(\overline{\sigma}_1, \ldots, \overline{\sigma}_n) R(X^\alpha).$$

Furthermore, $R(X^\alpha)$ does not depend on the order of the exponents in $\alpha = (\alpha_1, \ldots, \alpha_n)$. Let then $F$ denote the set of all partitions in $E$, that is, those elements that form weakly decreasing sequences. The previous sum can be rewritten as

$$R(P) = \sum_{\mu \in F} \left( \left( \sum_{\alpha \in E, \alpha \text{ permutation of } \mu} P_\alpha(\overline{\sigma}_1, \ldots, \overline{\sigma}_n) \right) R(X^\mu) \right).$$

Now, let us suppose that $P$ can be computed in time $L$, so that all $P_\alpha$ can be computed in time $\Delta(n)L+2$ by Proposition 1. Let next $D(n)$ be the size of a straight-line program that computes the polynomials $R(X^\mu)$ in terms of $\overline{\sigma}_1, \ldots, \overline{\sigma}_n$. Still denoting $Q$ the polynomial such that $R(P) = Q(\overline{\sigma}_1, \ldots, \overline{\sigma}_n)$, the above formula shows that, knowing all $P_\alpha$ and $R(X^\mu)$, $Q$ can be computed for $n!$ additional operations, that is, in total time $\Delta(n)L + n! + D(n) + 2$. Thus, to conclude the proof of Theorem 3, it suffices to give an upper bound on $D(n)$. This is the object of the upcoming lemma.

**Lemma 5.** *For $n \geq 1$, the inequality $D(n) \leq 2(8^n - 1)$ holds.*

*Proof.* To a partition $\mu \in F$, we associate the monomial symmetric function $m_\mu$; these functions form a vector-space basis of $\mathbb{Q}[X_1, \ldots, X_n]^{\mathfrak{S}_n}$. Now, $R(X^\mu)$ coincides with $m_\mu$ up to a non-zero constant factor; we will start by considering the $m_\mu$ functions. We use induction, following the standard (SAGBI) rewriting process of monomial symmetric functions in terms of elementary symmetric functions w.r.t. the degree lexicographic term order.

Let $\mu = (\mu_1, \ldots, \mu_{n-1}) \in F$ be fixed, let $k$ be the number of non-zero parts in it, and $\nu$ the unique partition such that $X^\mu$ factors as $X^\nu X_1 \cdots X_k$. Consider the expansion of the product $m_\nu \sigma_k$ in the monomial basis, where $\sigma_k$ is the $k$th elementary symmetric polynomial:

$$m_\nu \sigma_k = \sum_\eta c_\eta m_\eta,$$

for some $c_\eta$ in $\mathbb{N}$. The leading term of this product is the product of the leading terms of its operands, namely $X^\nu X_1 \cdots X_k = X^\mu$. Let now $\eta$ be a partition appearing in the right hand side: it follows from the previous remark that $\eta \leq_{\text{deglex}} \mu$. Furthermore, $\eta_i \leq \mu_i$, whenever $\mu_i > 0$, and otherwise, $\eta_i \in \{0, 1\}$. It is then straightforward to check that:

(*i*) either $\eta$ is right away in $F$,

$(ii)$ or $m_\eta$ is of the form $m_\eta = X_1 \cdots X_n m_{\eta'}$ with $\eta'$ in $F$.

Define $c'_\eta = c_\eta$ for all $\eta$ in case $(i)$, and $c'_\eta = 0$ otherwise. Define also $d_{\eta'} = c_\eta$, for all $\eta, \eta'$ as in case $(ii)$, and $d_{\eta'} = 0$ otherwise. Then, finally, $c_\mu = 1$ and altogether, $m_\mu$ can be written as

$$m_\mu = m_\nu \sigma_k - \sum_{\eta \in F,\, \eta <_{\text{deglex}} \mu} c'_\eta m_\eta - \sigma_n \sum_{\eta' \in F,\, \eta' <_{\text{deglex}} \mu} d_{\eta'} m_{\eta'},$$

where the two sums

Furthermore, by considering the maximal possible number of monomials in $\sigma_k$, we see that the total number $N_\mu$ of terms appearing with a non-zero coefficient in the two sums can be bounded by $2^n$. We now switch to the $R(X^\mu)$ themselves, and re-introduce our symmetric functions $\overline{\sigma}_k$; this yields:

$$R(X^\mu) = \gamma_\nu R(X^\nu)\overline{\sigma}_k - \sum_{\eta \in F, \eta <_{\text{deglex}} \mu} c''_\eta R(X^\eta) - \overline{\sigma}_n \sum_{\eta' \in F\, \eta' <_{\text{deglex}} \mu} d'_{\eta'} R(X^{\eta'}),$$

for some constants $\gamma_\nu$, $c''_\eta$ and $d'_{\eta'}$. Counting the operations appearing in the right-hand side expression, we see that a straight-line program computing all polynomials $\{R(X^\eta), \eta \in F, \eta <_{\text{deglex}} \mu\}$ can be extended to further compute $R(X^\mu)$, for an additional cost of $2N_\mu + 3$ operations. So, there exists a straight line program of size $\sum_{\mu \in F}(2N_\mu + 3)$, which computes $R(X^\mu)$ for all $\mu \in F$.

Now, $F$ is in bijection with the Dyck paths of length $2n$, so that the cardinality of $F$ is given by the $n$-th Catalan number $C(n) = \frac{1}{n+1}\binom{2n}{n}$. The estimate $(2 \cdot 2^n + 3)C(n) \leq 2(8^n - 1)$ proves the lemma. □

# References


[1] N. Bourbaki. *Éléments de mathématique. Algèbre. Chapitres 1 à 3.* Hermann, Paris, 1970.

[2] R. P. Brent. Multiple-precision zero-finding methods and the complexity of elementary function evaluation. In *Analytic computational complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1975)*, pages 151–176. Academic Press, New York, 1976.

[3] P. Bürgisser, M. Clausen, and A. M. Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1997.

[4] A. Colin. Solving a system of algebraic equations with symmetries. *Journal of Pure and Applied Algebra*, 117/118:195–215, 1997.

[5] D. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, 1997.

[6] L. Ducos and C. Quitté. Algèbre de décomposition universelle. Technical report, Université de Poitiers, 1996.

[7] W. Feit. A method for computing symmetric and related polynomials. *Journal of Algebra*, 234:540–544, 2000.

[8] P. Gaudry and É. Schost. Construction of secure random curves of genus 2 over prime fields. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 239–256. Springer, 2004.

[9] M. Giusti, K. Hägele, J. Heintz, J.-E. Morais, J.-L. Montaña, and L.-M. Pardo. Lower bounds for Diophantine approximation. *Journal of Pure and Applied Algebra*, 117/118:277–317, 1997.

[10] M. Giusti, J. Heintz, J.-E. Morais, J. Morgenstern, and L.-M. Pardo. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124:101–146, 1998.

[11] M. Giusti, J. Heintz, J.-E. Morais, and L.-M. Pardo. When polynomial equation systems can be solved fast? In *Proceedings of AAECC 11*, volume 948 of *Lecture Notes in Computer Science*, pages 205–231. Springer-Verlag, 1995.

[12] A. Lascoux. *Symmetric functions and combinatorial operators on polynomials*, volume 99 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC, 2003.

[13] A. Lascoux and P. Pragacz. *S*-function series. *Journal of Physics. A. Mathematical and General*, 21(22):4105–4114, 1988.

[14] A. Lascoux and M.-P. Schützenberger. Polynômes de Schubert. *Comptes Rendus des Séances de l'Académie des Sciences. Série I. Mathématique*, 294(13):447–450, 1982.

[15] N. Rennert and A. Valibouze. Calcul de résolvantes avec les modules de Cauchy. *Experimental Mathematics*, 8(4):351–366, 1999.

[16] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Technical report, University of Tübingen, 1982.

[17] B. Sturmfels. *Algorithms in invariant theory*. Texts and Monographs in Symbolic Computation. Springer-Verlag, 1993.

[18] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 1999.

[19] J. von zur Gathen and V. Shoup. Computing Frobenius maps and factoring polynomials. *Computational Complexity*, 2(3):187–224, 1992.