Available online at www.sciencedirect.com
SCIENCE DIRECT®

Journal of Symbolic Computation 41 (2006) 475–511

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Bruno Buchberger’s PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal[☆]

Bruno Buchberger[*]

Johannes Kepler University of Linz (JKU), RISC Institute, Schloss Hagenberg, 4232 Hagenberg, Austria

Available online 15 December 2005

Dissertation to obtain the degree Doctor of Philosophy from Leopold-Franzens University, Innsbruck

Abstract

This is the English translation (by Michael P. Abramson) of the PhD thesis of Bruno Buchberger, in which he introduced the algorithmic theory of Gröbner bases. Some comments by Buchberger on the translation and the thesis are given in an additional short paper in this issue of the Journal of Symbolic Computation.

1. Introduction

The residue class ring of a zero dimensional polynomial ideal in $K[x_1, \ldots, x_n]$ has the structure of an algebra with finitely many basis elements. In the present work, an algorithm for enabling the computation of these basis elements from the generating polynomials of a polynomial ideal, which was provided by Professor Wolfgang Gröbner during his research seminar in the spring of 1964, will be studied more closely. The goal of studying this algorithm is to find a termination criterion for the algorithm (Sections 4 and 8), and to sufficiently systematize it so that it is suitable for implementation on an electronic computer (Sections 4, 6 and 9). Certain inherent properties will also be presented, which suggest an application to the calculation of the Hilbert function of an arbitrary polynomial ideal (Sections 5 and 7).

[☆] Translation by Michael P. Abramson.

[*] Tel.: +43 732 2468 9921; fax: +43 732 2468 9930.
E-mail address: bruno.buchberger@jku.at.

0747-7171/$ - see front matter © 2005 Published by Elsevier Ltd
doi:10.1016/j.jsc.2005.09.007

2. Abbreviations, symbols, concepts and theorems used

Abbreviations

P-ring	polynomial ring
P-ideal	polynomial ideal
LCM	least common multiple
PP	power product
PPR	residue class of a power product

Symbols

$a \in M$	a is an element of the set M
$N \subset M$	the set N is a subset of the set M
$\overset{\text{def}}{=}$	equal by definition
$\bar{x}$	residue class of x
$a \equiv b \quad (\mathcal{A})$	a is congruent to b modulo the ideal $\mathcal{A}$
$x \to u$	the element x of a ring maps to the element u in the residue class ring modulo an ideal

Algebraic symbols and concepts will be used in precisely the same sense defined in Gröbner (1949). For this reason, we will not state the definitions of the concepts group, ring, field, ideal, congruence modulo an ideal, dimension of a P-ideal, and so forth. We state only additional definitions.

(2.1) **Convention:** The field K of coefficients of the P-ring $K[x_1, x_2, \ldots, x_n]$ will be assumed to be commutative.

(2.2) **Definition of an algebra:** An algebra is a finite R-module (Van der Waerden, 1937, p. 46) which is also a ring. However, we give this definition also explicitly, because later we will refer to individual parts of it: A nonempty set G is called an *algebra* (or *hypercomplex system*) of rank m over R if the following conditions hold:

(2.2.1) G is an additive abelian group.

(2.2.2) R is a ring with identity.

(2.2.3) There is a multiplication defined for elements $\alpha, \beta, \gamma, \ldots$ in R with elements $u, v, w, \ldots$ in G having the properties:

(2.2.3.1) The product of an element α in R with an element u in G always belongs to G.

(2.2.3.2) $\alpha(u + v) = \alpha u + \alpha v$.

(2.2.3.3) $(\alpha + \beta)u = \alpha u + \beta u$.

(2.2.3.4) $(\alpha\beta)u = \alpha(\beta u)$.

(2.2.3.5) Every element of G is uniquely representable as a linear combination $\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_m u_m$ by means of m fixed elements $u_1, u_2, \ldots, u_m$ with $\alpha_i \in R, u_i \in G$ $(i = 1, 2, \ldots, m)$.

(2.2.4) There is a multiplication defined among elements $u, v, w, \ldots$ in G with the following properties:

(2.2.4.1) The product of two elements u and v in G lies again in G.

(2.2.4.2) $(uv)w = u(vw)$.

(2.2.4.3)
$$\begin{aligned}(u + v)w &= uw + vw,\\ u(v + w) &= uv + uw.\end{aligned}$$

(2.2.4.4) $(\alpha u)v = u(\alpha v) = \alpha(uv)$ for all $\alpha \in R$.

(2.2.5) **Definition:** The set of m elements $u_1, u_2, \ldots, u_m$ in (2.2.3.5) is called a *basis* for the algebra.

From (2.2.3.4) and (2.2.4.4), it follows that

(2.2.6) $(\alpha u)(\beta v) = (\alpha\beta)(uv)$, and

(2.2.7) $\left(\sum_{j=1}^{m} \alpha_j u_j\right)\left(\sum_{k=1}^{m} \beta_k u_k\right) = \sum_{j=1}^{m}\sum_{k=1}^{m}(\alpha_j\beta_k)(u_j u_k).$

Therefore every product uv is computable provided that the products $u_j u_k$ are known, which, as elements of G, can be written as linear combinations of the $u_1, u_2, \ldots, u_m$.

(2.2.8) $u_j u_k = \sum_{l=1}^{m} \gamma_{jk}^{l} u_l \quad (j = 1, 2, \ldots, m;\ k = 1, 2, \ldots, m;\ \gamma_{jk}^{l} \in K).$

(2.2.9) **Definition:** The m^3 elements γ_{jk}^{i} of R appearing in (2.2.8) are called *structure constants* of the algebra G.

(2.2.10) **Definition:** The set of all presentations of type (2.2.8) is called the *multiplication table* of the algebra G.

A generalization of algebras to algebras with infinitely many basis elements is also possible. To do this, axiom (2.2.3.5) is modified to:

(2.2.3.5a) Every element is uniquely representable as a linear combination $\alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_m u_m$ of finitely many of the infinitely many basis elements $u_1, u_2, \ldots, u_k, \ldots$.

3. The residue class ring of a zero dimensional ideal

The following theorem holds for the residue class ring $K[x_1, x_2, \ldots, x_n]/\mathcal{A} = \mathcal{O}$ modulo a zero dimensional P-ideal $\mathcal{A} \subset K[x_1, x_2, \ldots, x_n]$:

Theorem 3.1. *The residue class ring $\mathcal{O}$ modulo a zero dimensional P-ideal is an algebra over the ground field K, if we take the addition between residue classes already defined in $\mathcal{O}$ as the additive group operation, the multiplication*[1] *αu between elements $\alpha \in K$ and the residue class $u \in \mathcal{O}$ as the multiplicative operation* (2.2.3)*, and the multiplication between residue classes already defined as the multiplicative operation* (2.2.4).

Proof. We will show successively that axioms (2.2.1) through (2.2.4) are satisfied.

(2.2.1) is satisfied since $\mathcal{O}$ is an abelian group with respect to its addition as a ring.

(2.2.2): As a field, K is a ring with unity.

(2.2.3.1) to (2.2.3.4) are in fact properties of the multiplication between elements of the ground field K and the residue classes.

To prove that (2.2.3.5) is satisfied, we need two lemmas:

[1] First a multiplication $\bar{\alpha} \cdot u$ is defined as multiplication between residue classes. But since the set of the $\bar{\alpha}$ is isomorphic to the ground field K, a multiplication $\alpha \cdot u$ is also immediately definable: $\alpha \cdot u \overset{\text{def}}{=} \bar{\alpha} \cdot u$

Lemma 3.2. *Let $u_1, u_2, \ldots, u_m$ be elements of an algebra G with the property that every $u \in G$ can be represented as*

$$u = \sum_{j=1}^{m} \alpha_j u_j \qquad (\alpha_j \in K;\ j = 1, 2, \ldots, m). \tag{3.2.1}$$

Then the following holds: If $u_1, u_2, \ldots, u_m$ are linearly independent over R, then the representation (3.2.1) *is unique, and conversely.*

Proof of 3.2. We assume that the representation (3.2.1) is not unique, i.e. there exists u such that on one hand

$$u = \sum_{j=1}^{m} \alpha_j u_j, \tag{3.2.2a}$$

and on the other hand

$$u = \sum_{j=1}^{m} \beta_j u_j \qquad (\alpha_j \neq \beta_j \text{ for at least one } j). \tag{3.2.2b}$$

Then,

$$0 = \sum_{j=1}^{m} (\alpha_j - \beta_j) u_j \qquad (\alpha_j - \beta_j \neq 0 \text{ for at least one } j). \tag{3.2.3}$$

However, (3.2.3) expresses the linear dependence of $u_1, u_2, \ldots, u_m$.

Suppose now that $u_1, u_2, \ldots, u_m$ are linearly dependent, so for example

$$u_1 = \sum_{j=2}^{m} \gamma_j u_j, \tag{3.2.4}$$

then some $u \in G$ having a representation (3.2.1) with $\alpha_1 \neq 0$ also has the representation

$$u = \sum_{j=1}^{m} \alpha_j u_j = \alpha_1 u_1 + \sum_{j=2}^{m} \alpha_j u_j = \sum_{j=2}^{m} (\alpha_1 \gamma_j + \alpha_j) u_j, \tag{3.2.5}$$

which contradicts uniqueness.

Lemma 3.3. *If a P-ideal has dimension* 0*, then it contains polynomials $p_i(x_i)$ $(i = 1, 2, \ldots, k)$ each of which depends only on a single variable x_i.*

Proof of 3.3. By Gröbner (1949, p. 98), the dimension of a P-ideal $\mathcal{A}$ is the maximal number of independent variables relative to $\mathcal{A}$. This implies that a zero dimensional P-ideal has no independent variables relative to $\mathcal{A}$, or every variable is dependent relative to $\mathcal{A}$. Hence, by the definition of dependence relative to a P-ideal (Gröbner, 1949, p. 97), for every variable x_i there exists a polynomial $p_i(x_i)$ in $\mathcal{A}$ that is dependent only on this variable $(i = 1, 2, \ldots, n)$.

It can now be proved that (2.2.3.5) holds for $\mathcal{O}$ by showing that there exist finitely many residue classes $u_1, u_2, \ldots, u_p$ in $\mathcal{O}$ by which all others can be represented. From these p residue classes, m linearly independent ones can always be chosen, by which all residue classes can be uniquely represented because of 3.2.

First, we represent each residue class from $\mathcal{O}$ by a linear combination of the residue classes of the PP in n variables $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$. Since

$$p_i(x_i) \overset{\text{def}}{=} x_i^{k_i} + c_{i,1} x_i^{k_i - 1} + \cdots + c_{i,k_i} \in \mathcal{A} \qquad (3.4a)$$
$$(i = 1, 2, \ldots, n; \quad c_{i,j} \in K; \quad j = 1, 2, \ldots, k_i),$$

we have

$$x_i^{k_i} \equiv -c_{i,1} x_i^{k_i-1} - \cdots - c_{i,k_i} = -\sum_{l=1}^{k_i} c_{i,l} x_i^{k_i-l} \qquad (\mathcal{A}), \qquad (3.4b)$$

and for the PP $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ of degree $\sigma \geq \tau$ $(\tau = k_1 + k_2 + \cdots + k_n)$

$$\begin{aligned} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} &= (x_1^{i_1} x_2^{i_2} \cdots x_j^{i_j - k_j} \cdots x_n^{i_n})\, x_j^{k_j} \qquad (3.4c) \\ &= -\sum_{l=1}^{k_j} c_{j,l} \;\; x_j^{k_j - l} \cdot x_1^{i_1} x_2^{i_2} \cdots x_j^{i_j - k_j} \cdots x_n^{i_n} \\ &\equiv -\sum_{l=1}^{k_j} c_{j,l} \;\; x_1^{i_1} x_2^{i_2} \cdots x_j^{i_j - l} \cdots x_n^{i_n} \qquad (\mathcal{A}), \end{aligned}$$

if $i_j \geq k_j$, which for power products of degree $\sigma \geq \tau$ must certainly be the case for some j.

Now the PPs $x_1^{i_1} x_2^{i_2} \cdots x_j^{i_j - l} \cdots x_n^{i_n}$ can themselves be further processed in the manner of (3.4c) provided that they have degree $\sigma \geq \tau$, until (3.4c) is transformed into

$$x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \equiv \sum c_{j_1, j_2, \ldots, j_n} x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n} \qquad (\mathcal{A}) \qquad (3.4d)$$

where only PPs $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ of degree $< \tau$ appear in the sum. A major task of the algorithm described in Section 4 is to find m linearly independent PPs $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ modulo $\mathcal{A}$ effectively.

(2.2.4.1) through (2.2.4.4) are precisely the properties of multiplication between residue classes.

In the case of a positive dimensional P-ideal, all considerations in the proof of 3.1 hold, with the exception of 3.3 and its consequences. Consequently:

Theorem 3.5. *The residue class ring modulo a P-ideal of dimension $d > 0$ is an algebra with infinitely many basis elements.*

The converse of 3.1 is also true, which we write in the following form:

Theorem 3.6. *If there are only finitely many linearly independent residue classes in $K[x_1, x_2, \ldots, x_n]/\mathcal{A} = \mathcal{O}$, then $\mathcal{A}$ is zero dimensional.*

Proof. Suppose there exist m linearly independent residue classes, and $m + 1$ residue classes are already linearly dependent. Certainly the PPs $1, x_i, \ldots, x_i^m$ $(i = 1, 2, \ldots, n)$ are also linearly dependent modulo $\mathcal{A}$, so there exists a relation

$$p_i(x_i) \overset{\text{def}}{=} \sum_{j=0}^{m} c_{i,j} \;\; x_i^{m-j} \equiv 0 \quad (\mathcal{A}) \qquad (i = 1, 2, \ldots, n). \qquad (3.6.1a)$$

However, this means

$$p_i(x_i) \in \mathcal{A}, \tag{3.6.1b}$$

and hence no variable is independent relative to $\mathcal{O}$, i.e. $\mathcal{A}$ is zero dimensional.

4. An algorithm for finding a basis of the algebra in 3.1

4.1. Preparatory considerations

For the purposes of the algorithm, we first establish a unique ordering on the power products $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ in n variables $x_1, x_2, \ldots, x_n$, namely the lexicographic order:

Definition 4.1. A PP $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ *precedes* a PP $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ (*has lower index than* the PP $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$) if:

1. $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ has lower degree than $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ or
2. the two degrees are identical, and the first non-vanishing difference $i_j - k_j$ is positive.

Now let $\mathcal{A} \in K[x_1, x_2, \ldots, x_n]$ be a given zero dimensional P-ideal with a generating basis

$$\mathcal{A} = (f_1, f_2, \ldots, f_s), \qquad \text{where} \tag{4.2}$$

$$f_j \overset{\text{def}}{=} \sum a^{(j)}_{i_1 i_2\cdots i_n}\ x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n} \qquad (j = 1, 2, \ldots, s;\ a^{(j)}_{i_1 i_2\cdots i_n} \in K). \tag{4.3}$$

(The summation is taken over all index combinations $(i_1, i_2, \ldots, i_n)$ up to a combination $(k_1^{(j)}, k_2^{(j)}, \ldots, k_n^{(j)})$, where $x_1^{k_1^{(j)}}x_2^{k_2^{(j)}}\cdots x_n^{k_n^{(j)}}$ has the highest index in the order 4.1 among the PPs of f_j with nonzero coefficients. Without loss of generality, we may assume $a^{(j)}_{k_1^{(j)},k_2^{(j)},\ldots,k_n^{(j)}} = 1$ since K is a field.)

This implies

$$\sum a^{(j)}_{i_1 i_2\cdots i_n}\ x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n} \equiv 0 \quad (\mathcal{A}), \qquad (j = 1, 2, \ldots, s) \tag{4.4a}$$

or

$$x_1^{k_1^{(j)}}x_2^{k_2^{(j)}}\cdots x_n^{k_n^{(j)}} \equiv -\sum a^{(j)}_{i_1 i_2\cdots i_n}\ x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n} \quad (\mathcal{A}) \qquad (j = 1, 2, \ldots, s) \tag{4.4b}$$

(where the summation is taken over all index combinations $(i_1, i_2, \ldots, i_n) \neq (k_1^{(j)}, k_2^{(j)}, \ldots, k_n^{(j)})$) and

$$x_1^{k_1^{(j)}+l_1}x_2^{k_2^{(j)}+l_2}\cdots x_n^{k_n^{(j)}+l_n} \equiv -\sum a^{(j)}_{i_1 i_2\cdots i_n}\ x_1^{i_1+l_1}x_2^{i_2+l_2}\cdots x_n^{i_n+l_n} \quad (\mathcal{A}) \tag{4.4c}$$
$$(j = 1, 2, \ldots, s;\ \ l_i = 0, 1, 2, \ldots;\ \text{for } i = 1, 2, \ldots, n).$$

If we consider the set of all polynomials $f \in \mathcal{A}$, which are the polynomials of the form

$$f = \sum_{j=1}^{s} d_j(x_1, x_2, \ldots, x_n) f_j = \sum a_{i_1 i_2\cdots i_n}\ x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}, \tag{4.5}$$
$$d_j(x_1, x_2, \ldots, x_n) \in K[x_1, x_2, \ldots, x_n] \qquad (j = 1, 2, \ldots, s),$$

then we obtain all possible relations between the PPRs in $\mathcal{O}$:

$$\sum a_{i_1 i_2 \cdots i_n} \quad x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \equiv 0 \qquad (\mathcal{A}). \tag{4.6}$$

(4.6) is a linear equation between PPRs. From every such congruence (4.6), we can now express, for example, the PP with the highest index among those PPs with nonzero coefficients in terms of PPs having lower index. There remain (in the case of a zero dimensional P-ideal, finitely many) PPs, which do not occur in any relation (4.6) as PPs with highest index. Their residue classes form a linearly independent basis of $\mathcal{O}$.

In order to arrive, step by step, at an algorithm which determines such a basis, we make one more observation. We assume we have found specific PPRs $u_1, u_2, \ldots, u_m$ coming from the relations (4.4b) already discussed, such that the residue classes of all PPs $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$ can be expressed as linear combinations of them:

$$x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n} = \sum_{i=1}^{m} \alpha_i^{(k_1,k_2,\ldots,k_n)} \ u_i \quad (\mathcal{A}); \qquad (\alpha_i^{(k_1,k_2,\ldots,k_n)} \in K) \tag{4.7}$$

(including the special case $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n} \equiv u_i$ $(\mathcal{A})$ for a specific i).

Furthermore, assume that it can be shown for every PP $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$, that by decomposing $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ into t factors

$$x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} = x_1^{i_1^{(1)}} x_2^{i_2^{(1)}} \cdots x_n^{i_n^{(1)}} x_1^{i_1^{(2)}} x_2^{i_2^{(2)}} \cdots x_n^{i_n^{(2)}} \cdots x_1^{i_1^{(t)}} x_2^{i_2^{(t)}} \cdots x_n^{i_n^{(t)}} \tag{4.8}$$
$$\left(\sum_{l=1}^{t} i_j^{(l)} = i_j \quad \text{for } j = 1, 2, \ldots, n; \quad 1 \le t \le \sum_{j=1}^{n} i_j \right),$$

substituting the representation (4.7) for the partial products in (4.8), multiplying out and further reducing the results (4.8a) of the multiplication

$$x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \equiv \sum b_{k_1 k_2 \cdots k_n} x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n} \qquad (\mathcal{A}), \tag{4.8a}$$

by applying the representation (4.7) for the $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$, we always come to the same representation (4.7) of $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$, independent of the division (4.8) into partial products. Then we can be certain that the $u_1, u_2, \ldots, u_m$ are linearly independent, and hence form a basis of $\mathcal{O}$ in the sense of (2.2.5).

Namely, if we can show the independence of the representation (4.7) of every PP from the decomposition in (4.8), then the residue classes of the polynomials

$$\begin{aligned} x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n} f_j \ = {} & x_1^{l_1+k_1^{(j)}} x_2^{l_2+k_2^{(j)}} \cdots x_n^{l_n+k_n^{(j)}} \\ & + \sum a_{i_1 i_2 \cdots i_n}^{(j)} \ x_1^{l_1+i_1} x_2^{l_2+i_2} \cdots x_n^{l_n+i_n} \in \mathcal{A} \\ & (j = 1, 2, \ldots, s; \quad l_i = 0, 1, 2, \ldots \quad \text{for } i = 1, 2, \ldots, n), \end{aligned} \tag{4.9}$$

which by (4.4c) possess a representation

$$x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n} f_j \equiv 0 \cdot u_1 + 0 \cdot u_2 + \cdots + 0 \cdot u_m \equiv 0 \tag{4.9a}$$
$$(j = 1, 2, \ldots, s; \quad l_i = 0, 1, 2, \ldots \quad \text{for } i = 1, 2, \ldots, n)$$

($\equiv$ is the identity symbol here!), possess only this representation, independent of the order in which we perform the necessary multiplications and additions in the computation of the residue classes of $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} f_j$. If there were still a relation

$$\sum_{i=1}^{m} c_i u_i = 0 \qquad (c_i \neq 0 \text{ for at least one } i), \tag{4.10}$$

which expressed the linear dependence of $u_1, u_2, \ldots, u_m$, then a polynomial $f \in \mathcal{A}$ would correspond to this relation, which would possess a representation (4.5) that can also be written as:

$$\begin{aligned} f &= \sum b^{(1)}_{l_1 i_2 \cdots l_n}\, x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n} f_1 + \sum b^{(2)}_{l_1 l_2 \cdots l_n}\, x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n} f_2 \\ &\quad + \cdots + \sum b^{(s)}_{l_1 l_2 \cdots l_n}\, x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n} f_s, \end{aligned} \tag{4.11}$$

if

$$\begin{aligned} &d_j(x_1, x_2, \ldots, x_n) \overset{\text{def}}{=} \sum b^{(j)}_{l_1 l_2 \cdots l_n}\, x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n} \\ &\quad (j = 1, 2, \ldots, s; \quad b^{(j)}_{l_1 l_2 \cdots l_n} \in K). \end{aligned} \tag{4.12}$$

The polynomials $x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n} f_j$ $(j = 1, 2, \ldots, s)$ which appear here are precisely of the type (4.9), for which we know that their residue classes have only the identically zero representation. Hence, $\bar{f} \equiv \sum_{i=1}^{m} c_i u_i$ also has only the identically zero representation relative to the residue classes $u_1, u_2, \ldots, u_m$ in $\mathcal{O}$. So there cannot be a relation (4.10).

Now the algorithm which follows proceeds just by taking the existing relations (4.4b) and computing the representation of all the PPRs in the manner described by multiplying out the representation of partial products and comparing them with each other. From two distinct representations of the very same PPR, one of the PPRs occurring in both representations (e.g. the one with the highest index) can then be eliminated. This means we can compute a representation of this PPR using other PPRs (with lower index). Now we must continue checking whether all the different ways to compute the PPR from the partial products lead to the same result until it is the case that there is in fact one way for all the PPRs. Then we know by the previous remarks that the remaining PPRs which are not a linear combination of other PPRs form a linearly independent basis of $\mathcal{O}$. Of course we cannot perform this check of the representation for infinitely many PPRs. So following the description of this algorithm, we must provide criteria which allow us to infer the uniqueness of representations of all PPRs from the uniqueness of the representations of finitely many PPRs.

4.2. Description of the algorithm

First just a convention about terminology: the representation of the residue class of a PP $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ as a linear combination of other PPRs with lower index, which cannot be represented as a linear combination of other PPRs in the current step of the algorithm, is called a *Σ-representation* of the PPR (sometimes, imprecisely, a Σ-representation of the PP under consideration).

We now describe the algorithm for the ideal $\mathcal{A}$ in (4.2) in a form from which later we could easily derive a rough flowchart for calculation with an electronic computer. However, we will not really do this, since another variant of the algorithm will be used for programming. We now explain the algorithm in more detail.

(A) We put the relations (4.4b) in a list which we call S. We consider the residue class of 1. If, because of a relation in the list S, this already could be replaced by a residue class of another constant, then $\mathcal{A}$ would possess only one residue class, and therefore would have dimension -1. (The same would be true if we encountered a relation $1 = 0$ later in the computation.) In general this will not be the case, and we go to (B).

(B) We take the next PP according to the ordering (4.1) and consider its residue class.

(BA) This class may already have one or several Σ-representations because of the relations in the list S. If so, then we write this representation in the row next to the PP under consideration and go to (BB). If not, then we go immediately to (BB).

(BB) We decompose the PP under consideration into two partial products in all possible ways and compute from this, if possible, the type of Σ-representation of the PP described in (4.7) ff. in terms of the Σ-representation of the PPs known up to this point. The Σ-representations obtained in this way, as well as the decompositions of the PP into two partial products, which do not lead to any Σ-representation, are written in the row next to the PP under consideration.

(C) In the row next to the PP, we can now have:

(CA) No Σ-representation of the PP, rather only decompositions into partial products. We remark that up to this point this PPR is not yet representable by other PPRs with lower indices and go to (B).

(CB) A single Σ-representation or several copies of the same Σ-representation of the PP under consideration. We go immediately to (B).

(CC) Several Σ-representations of the PP under consideration among which at least two are different. Using methods from linear algebra, we eliminate from these as many of the existing PPRs as possible, beginning with those with the largest index. Thus we obtain Σ-representations of PPs which did not possess such representations so far. We write all of these Σ-representations again into the list S. Then we begin again by considering the residue class of 1 and resume at (BA) (we say we *begin a new round*).

If we combine the preparatory considerations, the following holds about termination of the algorithm:

(4.13) The algorithm can be terminated if

1. for the round just run, every PP for which a Σ-representation is recorded in the list S has occurred as the PP under consideration in (B), and
2. it is certain that, from the decomposition into two partial products, a PPR may not obtain different Σ-representations. (By the criteria 4.14 and 4.19, this can be claimed already if no more different Σ-representations appear for specific finite degrees.)

Since we are storing both the Σ-representation of the PPRs and the decomposition into two partial products, it is easy to read off the multiplication table of the basis elements from the rows of the algorithm. Because of the systematic flow of the algorithm from one PP to the next by the order (4.1), it suffices to consider all decompositions into *two* partial products. Different decompositions of one partial product into further factors cannot change the resulting Σ-representations any more, since the decomposition of the partial products into further factors was just carried out in earlier steps of the algorithm, and it was guaranteed that these partial products have at most a single Σ-representation.

The logical flow of the algorithm will now be illustrated with an example:

Example 1. Let the given zero dimensional P-ideal be

$$\mathcal{A} = (x_1^2 - 2x_2 + x_1,\quad x_1x_3 - x_3,\quad x_3^2 - 2x_3 + x_2) \subset K[x_1, x_2, x_3].$$

First, we write down the individual rows of the algorithm and then describe each step explicitly.

$$\begin{array}{lll}
\text{List } S: & x_1^2 \equiv 2x_2 - x_1 & (\mathcal{A}) \\
& x_1x_3 \equiv x_3 & (\mathcal{A}) \\
& x_3^2 \equiv 2x_3 - x_2 & (\mathcal{A}) \\
& (x_2x_3 \longrightarrow)\ u_6 = u_3 & \\
& (x_1x_2 \longrightarrow)\ u_4 = u_2 & \\
& (x_2^2 \longrightarrow)\ u_5 = u_2 &
\end{array}$$

$$\begin{array}{lll}
1 & & \\
x_1 & \longrightarrow & u_1\circ \\
x_2 & \longrightarrow & u_2\circ \\
x_3 & \longrightarrow & u_3\circ \\
x_1^2 & \longrightarrow & 2u_2 - u_1 = u_1^2 \\
x_1x_2 & \longrightarrow & u_1u_2 = \cancel{u_4\circ} = u_2 \\
x_1x_3 & \longrightarrow & u_3 = u_1u_3 \\
x_2^2 & \longrightarrow & u_2^2 = \cancel{u_5\circ} = u_2 \\
x_2x_3 & \longrightarrow & u_2u_3 = \cancel{u_6\circ} = u_3 \\
x_3^2 & \longrightarrow & 2u_3 - u_2 = u_3^2 \\
x_1^3 & \longrightarrow & \cancel{2u_4} - \cancel{2u_2} + u_1 \\
x_1^2x_2 & \longrightarrow & \cancel{u_1u_4} = \cancel{2u_5} - \cancel{u_4} = u_2 \\
x_1^2x_3 & \longrightarrow & u_3 = \cancel{2u_6} - \cancel{u_3} \\
x_1x_2^2 & \longrightarrow & u_1u_5 = \cancel{u_2u_4} = \cancel{u_7\circ} = u_2 \\
x_1x_2x_3 & \longrightarrow & u_3 = \cancel{u_3} = \cancel{u_3u_4} \\
x_1x_3^2 & \longrightarrow & \cancel{2u_3} - \cancel{u_4} = 2u_3 - u_2 \\
x_2^3 & \longrightarrow & u_2 \\
x_2^2x_3 & \longrightarrow & \cancel{u_3} = u_3 \\
x_2x_3^2 & \longrightarrow & \cancel{2u_3} - \cancel{u_2} = 2u_3 - u_2 \\
x_3^3 & \longrightarrow & 3u_3 - 2u_2 \\
x_1^4 & \longrightarrow & \cdots \\
\vdots & &
\end{array}$$

1. First, using (A), we have written down the basis elements of $\mathcal{A}$ as residue class relations and put them in the first three rows of the list S. The residue class of 1 is not representable by these relations, so we go to (B).
2. Using (B), we consider the residue class of x_1. x_1 has neither a Σ-representation by (BA) nor a Σ-representation by (BB). So x_1 falls under (CA). We set $x_1 \to u_1\circ$, which should indicate that x_1 still has no Σ-representation. We go to (B).
3. The steps described in 2 must now be carried out for x_2 and x_3 according to the instructions in the algorithm. We arrive again at (B).
4. The next PP is x_1^2. Its residue class has a Σ-representation $x_1^2 \to 2u_2 - u_1$ by (BA), but not by (BB), nevertheless we record the decomposition $x_1^2 \to u_1 \cdot u_1$. Because of (CB), we go immediately back to (B).

5. For x_1x_2, only by (BB), we have a decomposition $u_1 \cdot u_2$. By (CA), we indicate by $x_1x_2 \to u_4\circ$ that x_1x_2 is a PP without any Σ-representation.
6. x_1x_3 is handled like x_1^2, x_2^2 and x_2x_3 like x_1x_2, and x_3^2 like x_1^2.
7. By (BB), x_1^3 has a decomposition $x_1^3 = x_1^2 \cdot x_1$, x_1^2 has a Σ-representation, which we can take from the list S or the row x_1^2. So x_1^3 is computed as follows:

$$x_1^3 = x_1^2x_1 \to (2u_2 - u_1)u_1 = 2u_1u_2 - u_1^2 = 2u_4 - 2u_2 + u_1.$$

   We write this representation down and, by (CB), go immediately to (B).
8. $x_1^2x_2$ has a Σ-representation, which is calculated as in 7, but also possesses an additional decomposition

$$x_1^2x_2 = x_1(x_1x_2) = u_1u_4,$$

   which does not lead to a Σ-representation.
9. Now $x_1^2x_3$ possesses two distinct Σ-representations, which can be computed from

$$x_1^2x_3 = x_1(x_1x_3) \qquad \text{and} \qquad x_1^2x_3 = (x_1^2) \cdot x_3,$$

   as in 7. So we are in case (CC). We eliminate u_6 from both representations ($u_6 = \overline{x_2x_3}$ has a higher index than $u_3 = \overline{x_3}$!). We record the relation $u_6 = u_3$ in the list S and begin the second round.
10. We see immediately that in the second round of the algorithm, nothing changes from the first round until the row for x_2x_3. For x_2x_3, we can replace u_6 by u_3 and we cancel u_6. Again nothing changes until the row $x_1^2x_3$, and we get two identical Σ-representations for x_2x_3 from the substitution $u_6 = u_3$. We let one stay and cancel the other.
11. $x_1x_2^2$ has two decompositions into partial products, neither of which leads to Σ-representations. By (CA), we set $x_1x_2^2 \to u_7\circ$.
12. $x_1x_2x_3$ has three decompositions into partial products, two of which lead to Σ-representations which, however, are identical. By (CB), we go again immediately to (B).
13. $x_1x_3^2$ is handled like $x_1^2x_3$. We obtain a new relation for the list S: $u_4 = u_2$, begin with a new round, and take the corresponding steps as in 10, whereby we obtain $x_1x_2 \to u_2$ and $x_1^3 \to u_1$. However, by using all existing relations, we arrive at two distinct Σ-representations for $x_1^2x_2$ from which we can eliminate $u_5 = u_2$. We again store this relation in the list S.
14. If we now begin again with a new round, we obtain $x_2^2 \to u_2$, and then $x_1^2x_2 \to u_2$ as the only Σ-representation. The two decompositions of $x_1x_2^2$ produce Σ-representations which, however, are identical. The decomposition of $\overline{x_1x_2x_3}$ in $u_3 \cdot u_4$ produces u_3 once again. $2u_3 - u_2$ is the only Σ-representation of $x_1x_3^2$. The remaining PPs up to x_3^3 have only a single Σ-representation.

At this stage, the algorithm can be terminated: First of all, the PPs on the left side of the congruences in the list S have all occurred in the last round, and secondly, by applying the subsequent Theorem 4.14, no more relations can appear from different decompositions of a PP into partial products.

The PPRs with Σ-representations

$$1, u_1, u_2, u_3$$

remain as basis elements. Their multiplication table can be read off from the rows of the algorithm:

	u_1	u_2	u_3
u_1	$2u_2 - u_1$	u_2	u_3
u_2		u_2	u_3
u_3			$2u_3 - u_2$

The multiplications

$$\begin{aligned} 1 \cdot 1 &= 1 \\ 1 \cdot u_1 &= u_1 \\ 1 \cdot u_2 &= u_2 \\ 1 \cdot u_3 &= u_3 \end{aligned}$$

are trivial and, therefore, are not shown in the multiplication table.

In practical computation, the rows of the algorithm and the list S are combined into a single diagram.

4.3. Termination criteria for the algorithm

Theorem 4.14. *Let $u_1, u_2, \ldots, u_m$ be finitely many PPRs from which all others can be linearly combined. Let u_m have the highest index in the order* (4.1) *and let it have degree k. (In view of observation* (4.13)*, let it further be guaranteed that the PPs on the left side of the list S, whose degrees can be at most a finite value p, in steps (BA) and (BB) of the algorithm, only get a single Σ-representation. At degree p this is verified!) Then we have: If we have checked that the PPs up to degree* $2k + 1$ *produce only a single Σ-representation, then we can be certain that the decompositions of additional PPs always lead to only one Σ-representation.*

Proof. At degree $2k + 1$ it is checked that the following identities hold:

$$u_j(u_i u_k) = (u_j u_i)u_k \quad \begin{pmatrix} j = 1, 2, \ldots, m-1, & k = 1, 2, \ldots, m, \\ i = 1, 2, \ldots, l, & l \leq m \end{pmatrix}, \tag{4.15}$$

(where $u_1, u_2, \ldots, u_l$ are the residue classes of those variables $x_{i_1}, x_{i_2}, \ldots, x_{i_l}$ with the property that x_{i_j} has no Σ-representation ($j = 1, 2, \ldots, l$)). For the PPs with degree $> 2k + 1$, every decomposition into two partial products produces a Σ-representation since one of the two factors must have degree larger than k and hence possesses a Σ-representation from which a Σ-representation of the PP under consideration results. Two arbitrary decompositions of such a PP $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ into two factors can be converted into one another by finitely many steps of the form:

$$x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} = (x_1^{i_1 - i'_1} \cdots x_p^{i_p - i'_p} \cdots x_n^{i_n - i'_n})[(x_p) x_1^{i'_1} \cdots x_p^{i'_p - 1} \cdots x_n^{i'_n}], \tag{4.16}$$

$$= [x_1^{i_1 - i'_1} \cdots x_p^{i_p - i'_p} \cdots x_n^{i_n - i'_n} (x_p)](x_1^{i'_1} \cdots x_p^{i'_p - 1} \cdots x_n^{i'_n}). \tag{4.17}$$

Therefore, as soon as we know that (4.16) and (4.17) produce the same Σ-representation under the hypotheses of Theorem 4.14, we also know that only a single Σ-representation can be computed from the different decompositions of $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ into two factors. We have:

$$x_1^{i_1 - i'_1} \cdots x_p^{i_p - i'_p} \cdots x_n^{i_n - i'_n} \longrightarrow \sum_{j=1}^{m} \alpha_j u_j, \tag{4.18a}$$

$$x_1^{i'_1} \cdots x_p^{i'_p - 1} \cdots x_n^{i'_n} \longrightarrow \sum_{k=1}^{m} \beta_k u_k, \qquad \text{and} \tag{4.18b}$$

$$x_p \longrightarrow \sum_{i=1}^{l} \gamma_i u_i \tag{4.18c}$$

$$(\alpha_j, \beta_k, \gamma_i \in K; \quad j, k = 1, 2, \ldots, m; \quad i = 1, 2, \ldots, l).$$

Using (4.18a)–(4.18c), we compute (4.16) and (4.17) further:

$$(x_1^{i_1-i_1'} \cdots x_p^{i_p-i_p'} \cdots x_n^{i_n-i_n'})[(x_p)x_1^{i_1'} \cdots x_p^{i_p'-1} \cdots x_n^{i_n'}] \longrightarrow \tag{4.16a}$$

$$\left(\sum_{j=1}^{m} \alpha_j u_j\right)\left(\sum_{i=1}^{l} \gamma_i u_i \sum_{k=1}^{m} \beta_k u_k\right) = \sum_{j=1}^{m}\sum_{i=1}^{l}\sum_{k=1}^{m} \alpha_j \gamma_i \beta_k u_j (u_i u_k),$$

$$[x_1^{i_1-i_1'} \cdots x_p^{i_p-i_p'} \cdots x_n^{i_n-i_n'}(x_p)](x_1^{i_1'} \cdots x_p^{i_p'-1} \cdots x_n^{i_n'}) \longrightarrow \tag{4.17a}$$

$$\left(\sum_{j=1}^{m} \alpha_j u_j \sum_{i=1}^{l} \gamma_i u_i\right)\left(\sum_{k=1}^{m} \beta_k u_k\right) = \sum_{j=1}^{m}\sum_{i=1}^{l}\sum_{k=1}^{m} \alpha_j \gamma_i \beta_k (u_j u_i) u_k.$$

But by (4.15), the final expressions of (4.16a) and (4.17a) produce the same Σ-representation under the hypotheses of the theorem.

Because of this theorem, we can terminate the algorithm at step 14 in Example 1, since u_1, u_2, u_3 satisfy the hypotheses of Theorem 4.14, and hence all additional PPRs have only a single Σ-representation. Theorem 4.14 can sometimes be sharpened:

Theorem 4.19. *If every PP of degree $k + 1$ already possesses a Σ-representation because each is a multiple of PPs having a Σ-representation (the basis element with the highest index has again degree at most k, and it will be assumed again that the relations present in the list S have all been used), then we have: If we have checked that the PPs up to degree $2k - 1$ produce at most a single Σ-representation, then we can be certain that the decompositions of additional PPs always lead to only one Σ-representation.*

Proof. The proof of this theorem will be given in Section 6 in connection with results from Section 5.

In Example 1, this theorem cannot be profitably used, since for degree $k + 1 = 2$, every PP possesses a Σ-representation, not because they are multiples of PPs having a Σ-representation, but rather because of the relations recorded in the list S. If we take $k+1 = 3$, then the hypothesis of Theorem 4.19 holds: Every Σ-representation of a PP of degree 3 results from the PP being a multiple of a PP of degree 2 having a Σ-representation. Since $2k - 1 = 3$, this theorem is no more advantageous than Theorem 4.14.

The following Conjectures 4.20 and 4.21 on the termination of the algorithm turn out to be wrong:

Conjecture 4.20. *Assume that all products $u_i u_k$ ($i = 1, 2, \ldots, p$; $k = i, i + 1, \ldots, p$) of all residue classes $u_l \circ$ that appeared in (CA) are already treated according to the instructions of the algorithm ($l = 1, 2, \ldots, p$, p is the highest index that occurred so far in the PPRs $u_l \circ$; in Example 1, $p = 7$). Assume also that all PPs that occur on the left side of the list S are treated in the algorithm. Then no more different Σ-representations can appear for any PPR.*

In other words, this is the question of whether it suffices to compute only the multiplication table of the $u_l \circ$.

Counterexample. $\mathcal{A} = (x_1^2 - 2x_2,\ \ x_2^2 - 2x_2,\ \ x_1x_2 - x_2) \subset K[x_1, x_2]$.

$$\begin{array}{rcl@{\quad}ll}
1 & & & & \\
x_1 & \longrightarrow & u_1\circ & & \\
x_2 & \longrightarrow & u_2\circ & & \\
x_1^2 & \longrightarrow & 2u_2 = u_1^2 & & \\
x_1x_2 & \longrightarrow & u_2 = u_1u_2 & & \\
x_2^2 & \longrightarrow & 2u_2 = u_2^2 & \longleftarrow & \text{We could stop here by the} \\
 & & & & \text{hypotheses in the conjecture,} \\
x_1^3 & \longrightarrow & 2u_2 & & \\
x_1^2x_2 & \longrightarrow & u_2 = 4u_2 & \longleftarrow & \text{but two different } \Sigma\text{-representations} \\
 & & & & \text{of a PPR appear here.}
\end{array}$$

Conjecture 4.21. *If the associativities,*

$$u_i(u_ju_k) = (u_iu_j)u_k, \tag{4.21.1}$$

are proved ($i, j, k = 1, 2, \ldots, l$, where $u_1, \ldots, u_l$ are again the residue classes of those variables $x_{i_1}, \ldots, x_{i_l}$ with the property that x_{i_j} has no Σ-representation ($j = 1, 2, \ldots, l$)), which is the case for degree 3*, and if the PPs on the left side of the list S have already been processed by the instructions of the algorithm, then no more different Σ-representations can appear in a PPR.*

Counterexample. $\mathcal{A} = (x_1^2x_2 - x_1^2,\ \ x_1x_2^2 - x_2) \subset K[x_1, x_2]$.

$$\begin{array}{rcl@{\quad}ll}
1 & & & & \\
x_1 & \longrightarrow & u_1\circ & & \\
x_2 & \longrightarrow & u_2\circ & & \\
x_1^2 & \longrightarrow & u_3\circ & & \\
x_1x_2 & \longrightarrow & u_4\circ & & \\
x_2^2 & \longrightarrow & u_5\circ & & \\
x_1^3 & \longrightarrow & u_6\circ = u_1u_3 & & \\
x_1^2x_2 & \longrightarrow & u_3 = u_2u_3 = u_1u_4 & & \\
x_1x_2^2 & \longrightarrow & u_1u_5 = u_2u_4 = u_2 & & \\
x_2^3 & \longrightarrow & u_2u_5 = u_7\circ & \longleftarrow & \text{We could stop here by the} \\
 & & & & \text{hypotheses in Conjecture 4.21} \\
x_1^4 & \longrightarrow & u_3^2 = u_1u_6 = u_8\circ & & \\
x_1^3x_2 & \longrightarrow & u_6 = u_2u_6 = u_3u_4 & & \\
x_1^2x_2^2 & \longrightarrow & u_4 = u_3 = u_3u_5 = u_4^2 & \longleftarrow & \text{but another new relation between} \\
 & & & & \text{residue classes appears here.}
\end{array}$$

By examining the proof of Theorem 4.14, we see further that the assumption (4.21.1) would not suffice to prove the claim 4.21 in a similar manner. Only the validity of the associativities (4.15) (which requires more than (4.21.1)) will make the proof possible.

5. The appearance of different Σ-representations in one step of the algorithm

In this section, we would like to derive a rule which tells us in which row of the algorithm the possibility exists that we will come to different Σ-representations for the very same PPR. To do this, we will prove four lemmas with which we can then bring the algorithm into a somewhat modified form in Section 6.

Lemma 5.1. *If we use the algorithm for ideals of the form*

$$\mathcal{A} = (f_1) \in K[x_1, x_2, \ldots, x_n] \qquad \textit{(principal ideals)},$$

where

$$f_1 \overset{\text{def}}{=} x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n} + \ldots$$

($x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$ has the highest index of all the PPs of f_1), then the different decompositions (4.8) *of an arbitrary PP into partial products can never lead to different Σ-representations.*

Proof. First, it should be noted that we could apply the algorithm in its present form purely formally to P-ideals for which we do not know the dimension. However, then is it possible that more and more PPRs appear with no Σ-representation. The hypotheses of Theorems 4.14 and 4.19 are then never satisfied, so we never know when we could terminate the algorithm. Hence, we can apply the algorithm to the ideal $\mathcal{A} = (f_1)$ as well, which for $n > 1$ is certainly not zero dimensional (Gröbner, 1949, p. 123).

To prove Lemma 5.1, we establish first that for $f \in \mathcal{A}$, where

$$f \overset{\text{def}}{=} x_1^{K_1} x_2^{K_2} \cdots x_n^{K_n} + \ldots \tag{5.2}$$

($x_1^{K_1} x_2^{K_2} \cdots x_n^{K_n}$ has the highest index of all the PPs of f), it follows that $x_1^{K_1} x_2^{K_2} \cdots x_n^{K_n}$ is a multiple of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$. PPs which are multiples of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$ obtain Σ-representations by step (BB). PPs $x_1^{L_1} x_2^{L_2} \cdots x_n^{L_n}$ which are not multiples of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$ cannot have any Σ-representation, because if

$$x_1^{L_1} x_2^{L_2} \cdots x_n^{L_n} \equiv \sum a_{i_1 i_2 \cdots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \qquad (\mathcal{A}) \tag{5.3}$$

($x_1^{L_1} x_2^{L_2} \cdots x_n^{L_n}$ has a larger index than every $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$), then

$$f \overset{\text{def}}{=} x_1^{L_1} x_2^{L_2} \cdots x_n^{L_n} - \sum a_{i_1 i_2 \cdots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} \in \mathcal{A}, \tag{5.4}$$

in contradiction to $x_1^{L_1} x_2^{L_2} \cdots x_n^{L_n}$ not being a multiple of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$. Now if in the course of the algorithm, a PPR contained two different Σ-representations, then from it, we could compute a Σ-representation for a PP $x_1^{L_1} x_2^{L_2} \cdots x_n^{L_n}$, where $x_1^{L_1} x_2^{L_2} \cdots x_n^{L_n}$ is not a multiple of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$, because the residue classes of the other PPs (the multiples of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$) do not occur in a Σ-representation, but will themselves be represented as linear combinations of PPRs by steps (BA) or (BB).

Because of later applications, we will now prove Lemma 5.1 also in a second more complicated way: We look successively at PPs with smaller index than $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$, then at $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$, and finally at PPs with larger index than $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$, and show that all PPs of each group obtain at most a single Σ-representation by the method of the algorithm.

We begin with first group, the PPs with smaller index than $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$. For these, absolutely no Σ-representation can be derived, since they are not multiples of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$.

$x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$ has a single Σ-representation because of $f_1 \equiv 0 \quad (\mathcal{A})$. Decompositions of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$ into partial products cannot lead to a Σ-representation because, as PPs of the first group, the partial products possess no Σ-representation.

Inside the third group, which consists of the PPs with larger index than $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$, there are two different types of PPs which we will denote by types 3A and 3B.

The group 3A comprises the PPs $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ which are not multiples of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$. The first such is the one immediately following $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$. This one has as partial products only PPs of the first group and has therefore no Σ-representation. We make the induction hypothesis: Up to the PP $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$, no PP of group 3A has a Σ-representation. Then $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$ itself cannot have a Σ-representation as well. Indeed in a decomposition of $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$ into partial products, no factor is a multiple of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$. The partial products are therefore PPs of the first group or of group 3A with smaller index than $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$. In both cases, they have no Σ-representation, so $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$ has none also.

The group 3B comprises the PPs $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ which are multiples of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$. Again we use induction. For $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$ itself, it has been shown already that it has only one Σ-representation. The induction hypothesis is: Every PP of group 3B up to, but not including, $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$ has only one Σ-representation. Now assume $x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n}$ has at least two decompositions which lead to a Σ-representation (otherwise there is nothing to show). These decompositions are such that at least one of the two factors is a multiple of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$:

$$x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n} = x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n} \cdot x_1^{k_1-j_1} x_2^{k_2-j_2} \cdots x_n^{k_n-j_n} = U, \tag{5.5}$$

$$x_1^{k_1} x_2^{k_2} \cdots x_n^{k_n} = x_1^{j_1'} x_2^{j_2'} \cdots x_n^{j_n'} \cdot x_1^{k_1-j_1'} x_2^{k_2-j_2'} \cdots x_n^{k_n-j_n'} = V. \tag{5.6}$$

W.l.o.g. let $x_1^{k_1-j_1} x_2^{k_2-j_2} \cdots x_n^{k_n-j_n}$ and $x_1^{k_1-j_1'} x_2^{k_2-j_2'} \cdots x_n^{k_n-j_n'}$ be the multiples of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$. The two representations can now be written down as follows:

$$\begin{aligned} U &= (x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n})[(x_1^{k_1-j_1-I_1} x_2^{k_2-j_2-I_2} \\ &\quad \cdots x_n^{k_n-j_n-I_n})(x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n})] \qquad (5.5a) \\ &\equiv (x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n})\Bigg[(x_1^{k_1-j_1-I_1} x_2^{k_2-j_2-I_2} \\ &\quad \cdots x_n^{k_n-j_n-I_n})\left(\sum_{j=1}^{p} \alpha_j u_j\right)\Bigg] \\ &\equiv \sum_{j=1}^{p} \alpha_j (x_1^{j_1} x_2^{j_2} \cdots x_n^{j_n})[(x_1^{k_1-j_1-I_1} x_2^{k_2-j_2-I_2} \\ &\quad \cdots x_n^{k_n-j_n-I_n})(u_j)] \ \ (\mathcal{A}), \qquad (5.5b) \end{aligned}$$

if $\sum\limits_{j=1}^{p} \alpha_j u_j$ is the Σ-representation of $x_1^{I_1} x_2^{I_2} \cdots x_n^{I_n}$, and

$$\begin{aligned} V &\equiv \sum_{j=1}^{p} \alpha_j (x_1^{j_1'} x_2^{j_2'} \cdots x_n^{j_n'})[(x_1^{k_1-j_1'-I_1} x_2^{k_2-j_2'-I_2} \\ &\quad \cdots x_n^{k_n-j_n'-I_n})(u_j)] \quad (\mathcal{A}). \end{aligned} \tag{5.6b}$$

The PPRs

$$\begin{aligned}&\overline{x_1^{j_1}x_2^{j_2}\cdots x_n^{j_n}\cdot x_1^{k_1-j_1-I_1}x_2^{k_2-j_2-I_2}\cdots x_n^{k_n-j_n-I_n}}\;u_j\\ &\quad=\overline{x_1^{j_1'}x_2^{j_2'}\cdots x_n^{j_n'}\cdot x_1^{k_1-j_1'-I_1}x_2^{k_2-j_2'-I_2}\cdots x_n^{k_n-j_n'-I_n}}\;u_j\\ &\quad=\overline{x_1^{k_1-I_1}x_2^{k_2-I_2}\cdots x_n^{k_n-I_n}}\;u_j\end{aligned}\tag{5.7}$$

already appear in the algorithm before the residue class $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ and have therefore only one Σ-representation because of the remarks thus far and the induction hypothesis. Hence, $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ possesses only one as well.

Lemma 5.8. *If we apply the algorithm to an ideal of the form*

$$\mathcal{A}=(f_1,\ f_2)\subset K[x_1,x_2,\ldots,x_n]$$

($f_1\overset{\text{def}}{=}x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}+\ldots,\ f_2\overset{\text{def}}{=}x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}+\ldots;\ x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ ($x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$) is the PP with the highest index among the PPs appearing in f_1 (f_2)), then we have:

1. *The residue class of $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ ($G_j=\max\,(I_j,K_j);\ j=1,2,\ldots,n$), i.e. the LCM of $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ and $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$, is the first PPR, for which different Σ-representations can appear in the algorithm.*
2. *If no distinct Σ-representations appear at $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$, then the algorithm produces no more Σ-representations for any PPR.*

Proof. By the steps of the algorithm, a PP which precedes $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ cannot have two different Σ-representations. A PP $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ can only have a Σ-representation if it is a multiple of $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ or $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$. It cannot be a multiple of $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ and $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$ simultaneously if it precedes $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$, since otherwise it would contain $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ also. By the same considerations as in the proof of Lemma 5.1, group 3B, it is clear that this PP can only have a single Σ-representation.

The situation is different for $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$. Namely,

$$\begin{aligned}x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}&=x_1^{G_1-I_1}x_2^{G_2-I_2}\cdots x_n^{G_n-I_n}\cdot x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}\\ &\equiv x_1^{G_1-I_1}x_2^{G_2-I_2}\cdots x_n^{G_n-I_n}\left(\sum_{j=1}^{p}\alpha_j u_j\right)\quad(\mathcal{A}),\end{aligned}\tag{5.9}$$

and

$$\begin{aligned}x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}&=x_1^{G_1-K_1}x_2^{G_2-K_2}\cdots x_n^{G_n-K_n}\cdot x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}\\ &\equiv x_1^{G_1-K_1}x_2^{G_2-K_2}\cdots x_n^{G_n-K_n}\left(\sum_{j=1}^{p}\beta_j u_j\right)\quad(\mathcal{A}),\end{aligned}\tag{5.10}$$

where $\sum_{j=1}^{p}\alpha_j u_j$ and $\sum_{j=1}^{p}\beta_j u_j$ are the Σ-representations of $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ and $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$, respectively. However, nothing here allows us to conclude that (5.9) and (5.10) are equal if we expand them further.

We now prove the second part of Lemma 5.8 by induction. Suppose that for $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$, only one Σ-representation appears. Then certainly for the PP immediately following $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$, only one Σ-representation appears as well, since this cannot be divisible by both $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ and $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$. (If it were divisible by both, it would be divisible by the LCM $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ also. But this could only happen if it has larger degree than $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ has or is identical with it. If it were to have larger degree, then $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ would be of the form x_n^l, but the next PP would then be x_1^{l+1}, which certainly is not divisible by x_n^l.)

The induction hypothesis reads: every PP following $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ up to, but not including, PP $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ has only one Σ-representation. Now $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ itself can obtain Σ-representations from its decompositions if

1. $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ divides a partial product, or
2. $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$ divides a partial product.

The Σ-representations from a decomposition of the first type are equal to each other, similarly for those from decompositions of the second type, by exactly the same reasoning as in the proof of 5.1, group 3B. If decompositions appear of types 1 and 2, then a decomposition also appears where a partial product is divisible by $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$. This decomposition belongs now to group 1 and group 2. The Σ-representation computed from this is identical to the Σ-representations resulting from 1 and 2, so they are also equal to each other.

Lemma 5.11. *If the PPs $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ and $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$ appearing in Lemma 5.8 have the property that $G_j = I_j + K_j$ $(j = 1, 2, \ldots, n)$ (that therefore the LCM $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ of $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ and $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$ is equal to the product of the two), then certainly $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ has only one Σ-representation.*

Proof. $x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n}$ can have decompositions where

1. $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ divides a partial product, or
2. $x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n}$ divides a partial product.

By reason of the corresponding remarks as in the proof of Lemma 5.1, group 3B, the Σ-representations that arise from decompositions of the first type are equal to each other. The same is true for the Σ-representations of the second type. The decomposition

$$x_1^{G_1}x_2^{G_2}\cdots x_n^{G_n} = x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}\cdot x_1^{K_1}x_2^{K_2}\cdots x_n^{K_n} \tag{5.12}$$

is simultaneously one of type 1 and 2. From this, it follows again that all the Σ-representations are equal to each other.

Lemma 5.13. *Let*

$$\mathcal{A} = (f_1, f_2, \ldots, f_s) \tag{5.13.1}$$

be the P-ideal, where $x_1^{I_1^{(l)}}x_2^{I_2^{(l)}}\cdots x_n^{I_n^{(l)}}$ is the PP of f_l with the largest index $(l = 1, 2, \ldots, s)$. Now if the PP

$$x_1^{G_1^{(k,l)}}x_2^{G_2^{(k,l)}}\cdots x_n^{G_n^{(k,l)}} \qquad G_j^{(k,l)} = \max(I_j^{(k)}, I_j^{(l)}) \tag{5.13.2}$$
$$(j = 1, 2, \ldots, n; \quad k, l = 1, 2, \ldots s)$$

(the LCM of $x_1^{I_1^{(k)}}x_2^{I_2^{(k)}}\cdots x_n^{I_n^{(k)}}$ and $x_1^{I_1^{(l)}}x_2^{I_2^{(l)}}\cdots x_n^{I_n^{(l)}}$) obtains only one Σ-representation by the steps of the algorithm, then every PPR has only one Σ-representation.

Proof. Lemma 5.13 is certainly true for the residue class of 1. We state the induction hypothesis: Lemma 5.13 holds up to, but not including, $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$. Now $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ can be divisible by all possible $x_1^{I_1^{(l)}}\,x_2^{I_2^{(l)}}\cdots x_n^{I_n^{(l)}}$. But with $x_1^{I_1^{(l_1)}}\,x_2^{I_2^{(l_1)}}\cdots x_n^{I_n^{(l_1)}}$ and $x_1^{I_1^{(l_2)}}\,x_2^{I_2^{(l_2)}}\cdots x_n^{I_n^{(l_2)}}$, it is also divisible by the LCM of both. By the remarks in the proof of Lemma 5.1, group 3B, the Σ-representations which arise from decompositions where one partial product is divisible by $x_1^{I_1^{(l_1)}}\cdots x_n^{I_n^{(l_1)}}$ are equal to each other, and similarly the Σ-representations from decompositions where a partial product is divisible by $x_1^{I_1^{(l_2)}}\cdots x_n^{I_n^{(l_2)}}$ are equal to each other. Both Σ-representations are equal to the Σ-representation from a decomposition where a partial product is divisible by $x_1^{G_1^{(l_1,l_2)}}\,x_2^{G_2^{(l_1,l_2)}}\cdots x_n^{G_n^{(l_1,l_2)}}$. Therefore, they are also equal to each other. In the same way, the equality of all possible Σ-representations of $x_1^{k_1}x_2^{k_2}\cdots x_n^{k_n}$ can also be shown.

6. Applying Lemmas 5.1, 5.8, 5.11 and 5.13 to simplify the algorithm

We start again with the ideal $\mathcal{A}$ of the form (5.13.1) with the additional definition (5.13.2). Among all of the $x_1^{G_1^{(k,l)}}\,x_2^{G_2^{(k,l)}}\cdots x_n^{G_n^{(k,l)}}$, let $x_1^{G_1^{(p,q)}}\,x_2^{G_2^{(p,q)}}\cdots x_n^{G_n^{(p,q)}}$ be the one with the lowest index ($1 \le p \le s$, $1 \le q \le s$).

If we proceed according to the instructions of the algorithm, the first PP that can obtain two different Σ-representations is $x_1^{G_1^{(p,q)}}\,x_2^{G_2^{(p,q)}}\cdots x_n^{G_n^{(p,q)}}$. For this reason, we will skip all the steps of the algorithm up to that point and immediately compute two Σ-representations of $x_1^{G_1^{(p,q)}}\,x_2^{G_2^{(p,q)}}\cdots x_n^{G_n^{(p,q)}}$ in two essentially different ways: We decompose this power product once so that one partial product is a multiple of $x_1^{I_1^{(p)}}\,x_2^{I_2^{(p)}}\cdots x_n^{I_n^{(p)}}$, and once so that one partial product is a multiple of $x_1^{I_1^{(q)}}\,x_2^{I_2^{(q)}}\cdots x_n^{I_n^{(q)}}$. If we obtain two different Σ-representations in this manner, then we eliminate the PPR with the highest index and obtain from this the Σ-representation of a PPR that has possessed none so far. This representation corresponds to a polynomial $f_{s+1} \in \mathcal{A}$, which we put into the basis of $\mathcal{A}$. If however we did not obtain different Σ-representations, then we jump to the LCM with the next highest index, from which we compute again Σ-representations of two essentially different types.

Every time we have found a new Σ-representation in this manner, we add the corresponding polynomial to the basis (this corresponds in the old algorithm to storing a Σ-representation in the list S) and jump to the LCM with the next highest index (this corresponds in the old algorithm to beginning a new round; but now we know precisely for each new round where the first time two different Σ-representations for a PPR can appear, and we proceed there with the computation right away).

If for an LCM $x_1^{G_1^{(k,l)}}\,x_2^{G_2^{(k,l)}}\cdots x_n^{G_n^{(k,l)}}$, it follows that $G_j^{(k,l)} = I_j^{(k)}$ (or $G_j^{(k,l)} = I_j^{(l)}$) for $j = 1, 2, \ldots, n$ (i.e. if the LCM is equal to one of the two PPs), then we compute the two Σ-representations, perhaps add a new resulting polynomial to the basis, but can delete f_k (resp. f_l) from the basis because the relation which exists between residue classes because of $f_k \equiv 0 \quad (\mathcal{A})$ ($f_l \equiv 0 \quad (\mathcal{A})$), now exists in any case because of $f_l \equiv 0 \quad (\mathcal{A})$ ($f_k \equiv 0 \quad (\mathcal{A})$).

As soon as the hypothesis of Lemma 5.13 is satisfied for the current basis $\mathcal{A} = (f_1', f_2', \ldots, f_{s'}')$ of the ideal, we can terminate the algorithm. Those PPRs which neither have a Σ-representation because of $f_j' \equiv 0 \quad (\mathcal{A})$, ($j = 1, 2, \ldots, s'$) nor because of $x_1^{l_1}x_2^{l_2}\cdots x_n^{l_n}f_j' \equiv 0 \quad (\mathcal{A})$,

$(l_i = 0, 1, 2, \ldots; i = 1, 2, \ldots, n)$, are basis elements of the polynomial ring modulo the ideal $\mathcal{A}$, whose multiplication table must now be computed using all available Σ-representations.

The calculation of two different Σ-representations of an LCM is the main work in practical computation. To do this, many Σ-representations of PPs of lower index must be prepared. Of course, one will record the Σ-representations of these auxiliary PPs as in the earlier algorithm, so that we need only compute them once. In this manner many elements of the multiplication table are computed as a by-product.

For programming, the algorithm was used in the form just discussed. A "recording" of the Σ-representations of the auxiliary PPs was not immediately possible because of the small memory of the machine used, and would also immediately overwhelm the capacity of larger computers on somewhat larger examples. Thus, we must tolerate a longer computation time for the benefit of huge memory savings.

For calculating with electronic computer, we will need one more consideration: It does not matter in which order we take the LCMs $x_1^{G_1^{(k,l)}} x_2^{G_2^{(k,l)}} \cdots x_n^{G_n^{(k,l)}}$ in order to compute two essentially different types of Σ-representations for them. Namely, if one LCM has only one Σ-representation when the basis of the ideal in this stage of the algorithm is precisely $\mathcal{A} = (f'_1, f'_2, \ldots, f'_{s'})$, then this LCM cannot obtain two different Σ-representations in later stages where all the relations $f'_j \equiv 0 \quad (\mathcal{A})$ $(j = 1, 2, \ldots, s')$, and perhaps even more, hold.

Therefore, if we were to take the LCMs in an order other than the one described previously, which we want to call the *normal order*, and if we were to thereby obtain a basis representation $\mathcal{A} = (f''_1, f''_2, \ldots, f''_{s''})$, then we could apply the algorithm again, this time using the normal order of the LCMs. However, all of these LCMs were already computed during the computation of the two essentially different types in the other order, and yielded only a single Σ-representation. Hence, they cannot now obtain two different ones, where certainly no fewer relations exist than before. Thus the basis $\mathcal{A} = (f''_1, f''_2, \ldots, f''_{s''})$ must already be what we would have obtained by computing relative to the normal order.

In what follows, we compute Example 1 again, this time with the algorithm in the new form.

Example 1.

$$\mathcal{A} = (x_1^2 - 2x_2 + x_1, x_1x_3 - x_3, x_3^2 - 2x_3 + x_2).$$

We transform the basis polynomials into the corresponding relations (1), (2), (3) in $\mathcal{O}$:

$$\begin{array}{llllllll}
(1) & x_1^2 & \equiv 2x_2 - x_1 & (\mathcal{A}) & (4) & x_2x_3 & \equiv x_3 & (\mathcal{A}) \\
(2) & x_1x_3 & \equiv x_3 & (\mathcal{A}) & (5) & x_1x_2 & \equiv x_2 & (\mathcal{A}) \\
(3) & x_3^2 & \equiv 2x_3 - x_2 & (\mathcal{A}) & (6) & x_2^2 & \equiv x_2 & (\mathcal{A}).
\end{array}$$

Furthermore, we make a list of the $x_1^{G_1^{(k,l)}} x_2^{G_2^{(k,l)}} \cdots x_n^{G_n^{(k,l)}}$ in order to determine their order, where we give (k, l) and its associated PP $x_1^{G_1^{(k,l)}} x_2^{G_2^{(k,l)}} \cdots x_n^{G_n^{(k,l)}}$. In the initial stage, these are (2,1), (3,1), (3,2):

(2, 1) $x_1^2x_3$	~~(3, 1) $x_1^2x_3^2$~~	~~(4, 1) $x_1^2x_2x_3$~~	(5, 1) $x_1^2x_2$	~~(6, 1) $x_1^2x_2^2$~~
	(3, 2) $x_1x_3^2$	(4, 2) $x_1x_2x_3$	(5, 2) $x_1x_2x_3$	~~(6, 2) $x_1x_2^2x_3$~~
		(4, 3) $x_2x_3^2$	~~(5, 3) $x_1x_2x_3^2$~~	~~(6, 3) $x_2^2x_3^2$~~
			(5, 4) $x_1x_2x_3$	(6, 4) $x_2^2x_3$
				(6, 5) $x_1x_2^2$.

We immediately strike through the $x_1^{G_1^{(k,l)}} x_2^{G_2^{(k,l)}} \cdots x_n^{G_n^{(k,l)}}$ for which, by 5.11, we need not calculate any Σ-representation. Finally, we prepare another diagram in order to record the auxiliary relations. To begin, we can already enter (1), (2), and (3) here.

$$
\begin{array}{lll}
1 & & \\
x_1 & & \\
x_2 & & \\
x_3 & & \\
x_1^2 & \equiv 2x_2 - x_1 & (\mathcal{A}) \\
x_1x_2 & \equiv x_2 & (\mathcal{A}) \\
x_1x_3 & \equiv x_3 & (\mathcal{A}) \\
x_2^2 & \equiv x_2 & (\mathcal{A}) \\
x_2x_3 & \equiv x_3 & (\mathcal{A}) \\
x_3^2 & \equiv 2x_3 - x_2 & (\mathcal{A}) \\
x_1^3 & & \\
x_1^2x_2 & \equiv x_2 & (\mathcal{A}) \\
x_1^2x_3 & \equiv x_3 & (\mathcal{A}) \\
x_1x_2^2 & & \\
x_1x_2x_3 & & \\
x_1x_3^2 & \equiv 2x_3 - x_2 & (\mathcal{A}) \\
x_2^3 & & \\
x_2^2x_3 & & \\
x_2x_3^2 & & \\
x_3^3. & &
\end{array}
$$

Now we compute, in two ways, representations for $x_1^2x_3$, the LCM with the lowest index.

$$x_1^2x_3 \equiv (x_1^2)x_3 \equiv (2x_2 - x_1)x_3 \equiv 2x_2x_3 - \cancel{x_1x_3} - x_3 \qquad (\mathcal{A}), \tag{6.1a}$$

$$x_1^2x_3 \equiv x_1(x_1x_3) \equiv x_1x_3 \equiv x_3 \qquad (\mathcal{A}). \tag{6.1b}$$

(The reduction of a representation of a PP as a linear combination of other PPs (mod $\mathcal{A}$!) to a Σ-representation can be carried out with minimal effort if we also strike through representable PPs of the representation and add their representations to the end of the expression.)

From (6.1a) and (6.1b), we get $x_2x_3 \equiv x_3 \quad (\mathcal{A})$. We write this as (4) under (3) (we take $x_2x_3 - x_3$ into the basis). Similarly, we complete the list of auxiliary relations. By doing this, we obtain new LCMs as well, namely (4,1), (4,2), and (4,3), which we write down at the end of (3,2). Also, we record in the list of auxiliary relations the representation $x_1^2x_3 \equiv x_3 \quad (\mathcal{A})$ calculated in (6.1). To indicate that (2,1) was already used, we equip it with a check mark.

Now we go to the calculation of Σ-representations for the next LCM, namely (4,2). It yields no relations, so we immediately take (3,2):

$$x_1x_3^2 \equiv (x_1x_3)x_3 \equiv x_3 \cdot x_3 \equiv \cancel{x_3^2} + 2x_3 - x_2 \qquad (\mathcal{A}), \tag{6.2a}$$

$$x_1x_3^2 \equiv x_1(x_3^2) \equiv x_1(2x_3 - x_2) \equiv \cancel{2x_1x_3} - x_2x_2 + 2x_3 \qquad (\mathcal{A}). \tag{6.2b}$$

This yields (5). As earlier, the list of auxiliary relations and LCMs will also be completed.

Now we consider (5.1):

$$x_1^2x_2 \equiv (x_1^2)x_2 \equiv (2x_2 - x_1)x_2 \equiv 2x_2^2 - \cancel{x_1x_2} - x_2 \qquad (\mathcal{A}), \tag{6.3a}$$

$$x_1^2x_2 \equiv (x_1x_2)x_1 \equiv x_2x_1 \equiv x_2 \qquad (\mathcal{A}). \tag{6.3b}$$

This yields (6). Again we make the necessary entries in the auxiliary list.

We continue in this manner. The later LCMs have only a single Σ-representation, as can be easily checked.

The saving in work which comes from the simplification of the algorithm is not evident from this example, but is considerable for more complicated ideals. Also we combine the different lists into a single diagram.

The proof of Theorem 4.19 can now be easily furnished. Suppose that while computing with the algorithm in the earlier form, the condition arises described in the hypotheses of Theorem 4.19; Let $f'_1, f'_2, \ldots, f'_{s'}$ be in the list S ($f'_j \overset{\text{def}}{=} x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}} + \ldots$ ($j = 1, 2, \ldots, s'$), $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$ is the PP of f'_j with the highest index). By assumption, those PPs $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$ with degree $\geq k+1$ are themselves multiples of some other $x_1^{I_1^{(p)}} x_2^{I_2^{(p)}} \cdots x_n^{I_n^{(p)}}$, so

$$x_1^{G_1^{(p,j)}} x_2^{G_2^{(p,j)}} \cdots x_n^{G_n^{(p,j)}} = x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}} \qquad (1 \leq p \leq s', \quad p \neq j).$$

For these PPs, we must compute two essentially different types of Σ-representations, then we can release them from the basis. (This was just assumed to have happened in Theorem 4.19.) Hence, it still remains to calculate the LCM $x_1^{G_1^{(j_1,j_2)}} x_2^{G_2^{(j_1,j_2)}} \cdots x_n^{G_n^{(j_1,j_2)}}$ for all the pairs of PPs $x_1^{I_1^{(j_1)}} x_2^{I_2^{(j_1)}} \cdots x_n^{I_n^{(j_1)}}$ and $x_1^{I_1^{(j_2)}} x_2^{I_2^{(j_3)}} \cdots x_n^{I_n^{(j_2)}}$, with degree $\leq k$. But these $x_1^{G_1^{(j_1,j_2)}} x_2^{G_2^{(j_1,j_2)}} \cdots x_n^{G_n^{(j_1,j_2)}}$ appear at degree $2k-1$ at the latest (with the help of Lemma 5.11!).

In the new form, the algorithm can be used for every arbitrary ideal, even when we do not know its dimension beforehand. It can be decided in every stage whether we must continue to compute or whether all of the important relations have already been found. By applying the algorithm to an arbitrary ideal $\mathcal{A} = (f_1, f_2, \ldots, f_s)$, we obtain a basis representation of the ideal:

$$\mathcal{A} = (f'_1, f'_2, \ldots, f'_{s'}) \qquad \text{with} \tag{6.a}$$

$$f'_j \overset{\text{def}}{=} x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}} + \ldots \tag{6.b}$$

(where $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$ is the PP of f'_j with the highest index), which has the property that the highest indexed PP of an arbitrary polynomial $f \in \mathcal{A}$ is a multiple of at least one of the PPs $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$. (If there were a polynomial $f \in \mathcal{A}$ not having this property, then a PPR $u_l \circ$ which was found to be a basis element by the algorithm would have a Σ-representation.)

It should also be noted that relative to the order (4.1) of the PPs, only one basis of the ideal can be found having this property. Indeed, if there were two distinct such bases

$$\mathcal{A} = (f'_1, f'_2, \ldots, f'_{s'}) \qquad \text{and} \tag{6.4}$$

$$\mathcal{A} = (f''_1, f''_2, \ldots, f''_{s''}) \tag{6.5}$$

(where $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$ $(x_1^{K_1^{(l)}} x_2^{K_2^{(l)}} \cdots x_n^{K_n^{(l)}})$ is the PP of f'_j (f''_l), $j = 1, 2, \ldots, s'$ $(l = 1, 2, \ldots, s'')$ with the highest index), then either one PP would have a Σ-representation relative to the one basis representation of the ideal but none relative to the other (if one PP occurs among the $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$ that does not occur among the $x_1^{K_1^{(l)}} x_2^{K_2^{(l)}} \cdots x_n^{K_n^{(l)}}$, or vice versa), or the Σ-representation of at least one PP would be different relative to the two basis representations. Both stand in contradiction to the result, which the application of the algorithm has to an ideal.

If among the $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$ $(j = 1, 2, \ldots, s')$ PPs of the form $x_i^{I_i}$ $(i = 1, 2, \ldots, n)$ occur, then certainly every PP starting from degree $\sum\limits_{i=1}^{n} I_i$ has a Σ-representation, this by reason of the same considerations as in the proof of Theorem 3.1. Thus there are only finitely many linearly independent PPRs in $\mathcal{O}$. Therefore by Theorem 3.6, $\mathcal{A}$ is zero dimensional. However, if among the $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$, PPs of the form $x_i^{I_i}$ do not occur for all i (e.g. no $x_1^{I_1^{(j)}} x_2^{I_2^{(j)}} \cdots x_n^{I_n^{(j)}}$ has the form $x_k^{I_k}$), then no PP of the form x_k^p $(p = 0, 1, 2, \ldots)$ has a Σ-representation, so $\mathcal{O}$ has infinitely many linearly independent elements, and is an algebra with infinitely many basis elements in the sense of 3.5. Therefore, $\mathcal{A}$ has higher dimension.

If in the course of the algorithm, $f \equiv 1$ must be taken into the basis, then $\mathcal{A}$ is the whole P-ring, so $\mathcal{A}$ has dimension -1. Thus with the help of the algorithm, we can also make certain statements about the dimension of an arbitrary ideal.

If we apply the algorithm in its second form to the ideal $\mathcal{A} = (f_1(x), f_2(x), \ldots, f_s(x)) \subset K[x]$, it produces for us the greatest common divisor of the basis polynomials $f_j(x)$ $(j = 1, 2, \ldots, s)$, thus replacing the Euclidean algorithm. An example of this:

Example 2. $\mathcal{A} = (x^3 - 7x^2 + 11x - 5, x^5 - 28x^3 + 16x^2 - 3x - 10, x^4 - 2x^3 - 19x^2 + 15x + 25)$.

~~(1) $x^3 \equiv 7x^2 - 11x + 5$ $(\mathcal{A})$~~
~~(2) $x^5 \equiv 28x^3 - 16x^2 + 3x + 10$ $(\mathcal{A})$~~
~~(3) $x^4 \equiv 2x^3 + 19x^2 - 15x - 25$ $(\mathcal{A})$~~
~~(4) $x^2 \equiv 7x - 10$ $(\mathcal{A})$~~
(5) $x \equiv 5$ $(\mathcal{A})$.

We compute first two Σ-representations of x^4. One is already there: (3). It only needs to be simplified by applying (1):

$$x^4 \equiv \cancel{2x^3} + \cancel{19x^2} - \cancel{15x} - \cancel{25} + \cancel{14x^2} - \cancel{22x} + \cancel{10} + 33x^2 - 37x - 15 \quad (\mathcal{A}).$$

We compute the other from (1):

$$x^4 \equiv \cancel{7x^3} - \cancel{11x^2} + \cancel{5x} + \cancel{49x^2} - \cancel{77x} + \cancel{35} + 38x^2 - 72x + 35 \qquad (\mathcal{A}).$$

This yields a new relation (4): $x^2 \equiv 7x - 10$ $(\mathcal{A})$. (3) can be deleted from the basis. We consider x^3:

From (1): $x^3 \equiv \cancel{7x^2} - \cancel{11x} + \cancel{5} + \cancel{49x} - \cancel{70} + 38x - 65$ $(\mathcal{A})$.
From (4): $x^3 \equiv \cancel{7x^2} - \cancel{10x} + \cancel{49x} - 70 + 39x$ $(\mathcal{A})$.

This yields (5): $x \equiv 5$ $(\mathcal{A})$. (1) can be deleted from the basis. We consider x^2:

From (4): $x^2 \equiv 25$ $(\mathcal{A})$.
From (5): $x^2 \equiv 5x \equiv 25$ $(\mathcal{A})$.

This yields no new relation, so (4) can be deleted from the basis. Also the computation of x^5 from (2) and from (5) produces identical Σ-representations, so (2) can be deleted from the basis.

(5) remains as the only basis polynomial left and is the greatest common divisor of the original three basis polynomials (Gröbner, 1949, p. 39).

7. The calculation of the Hilbert function of an ideal from a basis of the form (6.4)

In this section, the symbols and definitions from Gröbner (1949, p. 154ff), will be used. If we have found a basis for an ideal $\mathcal{A}$ of the form (6.4) using the method of the algorithm, then two lemmas hold for this basis.

Lemma 7.1. *The number of linearly independent polynomials in $\mathcal{A}$ of degree $\leq t$ is equal to the number of PPs of degree $\leq t$ which are multiples of at least one $x_1^{I_1^{(l)}} x_2^{I_2^{(l)}} \cdots x_n^{I_n^{(l)}}$, if $\mathcal{A} = (f'_1, f'_2, \ldots, f'_{s'})$ is a basis (6.4) of $\mathcal{A}$, and $f'_L \overset{\text{def}}{=} x_1^{I_1^{(l)}} x_2^{I_2^{(l)}} \cdots x_n^{I_n^{(l)}} + \ldots$, $l = 1, 2, \ldots, s'$ ($x_1^{I_1^{(l)}} x_2^{I_2^{(l)}} \cdots x_n^{I_n^{(l)}}$ is the PP of f_l with the highest index).*

Lemma 7.2. *The number of PPs mentioned in Lemma 7.1 is*

$$\begin{aligned} N(t; (f'_1, f'_2, \ldots, f'_{s'})) = {} & H(t - t_1; n-1) + H(t - t_2; n-1) \\ & \quad + \cdots + H(t - t_{s'}; n-1) \\ & - H(t - t_{1,2}; n-1) - H(t - t_{1,3}; n-1) \\ & \quad - \cdots - H(t - t_{s'-1,s'}; n-1) \\ & + \quad \cdots \\ & \vdots \\ & + (-1)^{s'-1} H(t - t_{1,2,\ldots,s'}; n-1), \end{aligned}$$

where t_l is the degree of f'_l, $t_{i_1,i_2,\ldots,i_k}$ is the degree of $x_1^{G_1^{i_1,i_2,\ldots,i_k}} x_2^{G_2^{i_1,i_2,\ldots,i_k}} \cdots x_n^{G_n^{i_1,i_2,\ldots,i_k}}$, and $G_i^{i_1,i_2,\ldots,i_k} = \max(I_i^{(i_1)}, I_i^{(i_2)}, \ldots, I_i^{(i_k)})$, for $i = 1, 2, \ldots, n$ and $1 \leq i_1 < i_2 < \cdots < i_k \leq s'$, $2 \leq s'$.

From these two lemmas, the following formula for the calculation of the Hilbert function of $\mathcal{A}$ follows immediately:

$$H(t; (f'_1, f'_2, \ldots, f'_{s'})) = H(t; n-1) - N(t; (f'_1, f'_2, \ldots, f'_{s'})). \tag{7.3}$$

For $t < 0$, $H(t; n)$ is defined here by $H(t; n) = 0$.

Proof of 7.1. First of all, we know that there must be at least as many linearly independent polynomials $f \in \mathcal{A}$ of degree $\leq t$ as PPs of degree $\leq t$ that are multiples of at least one $x_1^{I_1^{(l)}} x_2^{I_2^{(l)}} \cdots x_n^{I_n^{(l)}}$ ($l = 1, 2, \ldots, s'$) because the Σ-representations of the residue classes of these PPs are the residue classes of polynomials

$$f_{i_1,i_2,\ldots,i_n} \overset{\text{def}}{=} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n} + \cdots \in \mathcal{A}, \tag{7.4}$$

where every $f_{i_1,i_2,\ldots,i_n}$ has the PP $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ as the PP with the highest index. But these PPs are linearly independent, so there can only be a relation

$$\sum a_{i_1,i_2,\ldots,i_n} f_{i_1,i_2,\ldots,i_n} \equiv 0 \qquad (a_{i_1,i_2,\ldots,i_n} \in K), \tag{7.5}$$

if $a_{i_1,i_2,\ldots,i_n} = 0$.

Now if a polynomial $f^* \in \mathcal{A}$ with degree $\leq t$ were still to exist, which is not a linear combination of the $f_{i_1,i_2,\ldots,i_n}$, then we could form a polynomial

$$g = f^* - \sum a_{i_1,i_2,\ldots,i_n} f_{i_1,i_2,\ldots,i_n} \equiv 0 \qquad (a_{i_1,i_2,\ldots,i_n} \in K), \tag{7.6}$$

and thereby choose the coefficients $a_{i_1,i_2,\ldots,i_n}$ so that g contains none of the PPs $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$. Among the remaining PPs in g with nonzero coefficients, we seek now the one with the highest index, and call it $x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n}$. Its residue class was found to be a basis element by the algorithm. But (7.6) becomes

$$g \equiv 0 \qquad (\mathcal{A}) \tag{7.7}$$

in $\mathcal{O}$, whereby $x_1^{l_1} x_2^{l_2} \cdots x_n^{l_n}$ obtains a Σ-representation, contradicting its property as a basis element.

Proof of 7.2. We furnish the proof of 7.2 by induction on s'. For the case $s' = 1$, the formula becomes $N(t; (f_1)) = H(t - t_1; n - 1)$ in accordance with the fact that we must multiply $x_1^{I_1^{(1)}} x_2^{I_2^{(1)}} \cdots x_n^{I_n^{(1)}}$ by all PPs of degree $t - t_1$ in order to obtain the PPs of degree t which are multiples of $x_1^{I_1^{(1)}} x_2^{I_2^{(1)}} \cdots x_n^{I_n^{(1)}}$. Now suppose the formula holds for s' basis polynomials. The number of PPs which are multiples of at least one of the PPs

$$x_1^{I_1^{(l)}} x_2^{I_2^{(l)}} \cdots x_n^{I_n^{(l)}} \qquad (l = 1, 2, \ldots, s' + 1) \tag{7.8}$$

can be compiled in the following manner: We take all PPs which are multiples of the first s' PPs of (7.8) (their total can be computed by the induction hypothesis) and add to that all PPs which are multiples of $x_1^{I_1^{(s'+1)}} x_2^{I_2^{(s'+1)}} \cdots x_n^{I_n^{(s'+1)}}$, whose total has also already been computed. However, in this way, we have counted some PPs twice, namely those which are multiples of an $x_1^{G_1^{(j,s'+1)}} x_2^{G_2^{(j,s'+1)}} \cdots x_n^{G_n^{(j,s'+1)}}$ $(j = 1, 2, \ldots, s', G_i^{(j,s'+1)} = \max\ (I_i^{(j)}, I_i^{(s'+1)})$ for $i = 1, 2, \ldots, n)$, and were therefore already counted with the first group. This number can already be computed by the induction hypothesis. Hence, we must subtract these PPs from the earlier number. This yields:

$$\begin{aligned}
N(t; (f_1, f_2, \ldots, f_{s'+1})) = {} & H(t - t_1; n - 1) + \cdots + H(t - t_{s'}; n - 1) \\
& - H(t - t_{1,2}; n - 1) - \cdots - H(t - t_{s'-1,s'}; n - 1) \\
& + \cdots + (-1)^{s'-1} H(t - t_{1,2,\ldots,s'}; n - 1) \\
& + H(t - t_{s'+1}; n - 1) - \Big[H(t - t_{1,s'+1}; n - 1) + \cdots \\
& + H(t - t_{s',s'+1}; n - 1) - H(t - t_{1,2,s'+1}; n - 1) \\
& - \cdots - H(t - t_{s'-1,s',s'+1}; n - 1) \\
& + \cdots + (-1)^{s'-1} H(t - t_{1,2,\ldots,s',s'+1}; n - 1) \Big] \\
= {} & H(t - t_1; n - 1) + H(t - t_2; n - 1) \\
& \quad + \cdots + H(t - t_{s'+1}; n - 1) \\
& - H(t - t_{1,2}; n - 1) + H(t - t_{1,3}; n - 1) \\
& \quad + \cdots + H(t - t_{s',s'+1}; n - 1)
\end{aligned}$$

$$\begin{aligned} &+\cdots \\ &\vdots \\ &+\ (-1)^{s'}\ H(t-t_{1,2,\ldots,s'+1};n-1). \end{aligned}$$

Here, we used the fact that the LCM of

$$x_1^{G_1^{(i_1,s'+1)}}\cdots x_n^{G_n^{(i_1,s'+1)}},\quad x_1^{G_1^{(i_2,s'+1)}}\cdots x_n^{G_n^{(i_2,s'+1)}},\ldots,\quad x_1^{G_1^{(i_k,s'+1)}}\cdots x_n^{G_n^{(i_k,s'+1)}}$$

is equal to the LCM of

$$x_1^{I_1^{(i_1)}}\cdots x_n^{I_n^{(i_1)}},\quad x_1^{I_1^{(i_2)}}\cdots x_n^{I_n^{(i_2)}},\ldots,\quad x_1^{I_1^{(i_k)}}\cdots x_n^{I_n^{(i_k)}},\quad x_1^{I_1^{(s_1+1)}}\cdots x_n^{I_n^{(s_1+1)}},$$

where $1\le i_1<i_2<\cdots<i_k\le s'$; $1\le k\le s'$.

With this, we can compute the Hilbert function of an arbitrary ideal, after its basis is first brought into the required form (6.4) with the help of the algorithm.

8. Determination of a bound for the termination of the algorithm from the basis polynomials of the ideal

By the considerations in Section 6, we can now attempt to calculate a bound from the basis polynomials $f_1, f_2, \ldots, f_s$ of a P-ideal $\mathcal{A}=(f_1, f_2, \ldots, f_s)$ for how high a degree we must compute at the most, so that all steps of the algorithm are carried out (i.e. the hypothesis of Lemma 5.13 is satisfied).

Here a bound will be found for the case

$$\begin{aligned} \mathcal{A}\ &=(f_1, f_2, \ldots, f_s)\subset K[x_1,x_2], && (8.1)\\ f_j\ &=x_1^{I_1^{(j)}}x_2^{I_2^{(j)}}+\cdots \end{aligned}$$

($j=1,2,\ldots,s$; $x_1^{I_1^{(j)}}x_2^{I_2^{(j)}}$ is the PP of f_j which has the highest index). We will also need the following quantities:

$$I_j^{(l,k)}=\max(I_j^{(l)},I_j^{(k)});\qquad j=1,2;\ \ l=1,2,\ldots,s-1;\ \ k=l+1,\ldots,s. \tag{8.2a}$$

$$K^{(l,k)}=I_1^{(l,k)}+I_2^{(l,k)}. \tag{8.2b}$$

$$K=\max(K^{(l,k)});\qquad l=1,2,\ldots,s-1;\ \ k=l+1,\ldots,s. \tag{8.2c}$$

$$I_1=\min(I_1^{(l)})=I_1^{(l_1)};\qquad l=1,2,\ldots,s;\ \ 1\le l_1\le s. \tag{8.2d}$$

$$I_2=\min(I_2^{(l)})=I_2^{(l_2)};\qquad l=1,2,\ldots,s;\ \ 1\le l_2\le s. \tag{8.2e}$$

$$I=I_1+I_2. \tag{8.2f}$$

First of all, it is true that there are at most I PPs of degree K without a Σ-representation, namely $x_1^K, x_1^{K-1}x_2,\ldots,x_1^{K-I_2+1}x_2^{I_2-1}$ and $x_1^{I_1-1}x_2^{K-I_1+1}$, $x_1^{I_1-2}x_2^{K-I_1+2},\ldots,x_2^K$. $x_1^{K-I_2}x_2^{I_2}$ up to $x_1^{I_1^{(l_2)}}x_2^{K-I_1^{(l_2)}}$ are multiples of $x_1^{I_1^{(l_2)}}x_2^{I_2^{(l_2)}}$, and $x_1^{I_1^{(l_2)}-1}x_2^{K-I_1^{(l_2)}+1}$ up to $x_1^{I_1}x_2^{K-I_1}$ are multiples of $x_1^{I_1^{(l_1)}}x_2^{I_2^{(l_1)}}$. For this it need only be shown that

$$K-I_1^{(l_2)}+1\ge I_2^{(l_1)},\qquad \text{or} \tag{8.3a}$$

$$K+1\ge I_1^{(l_2)}+I_2^{(l_1)}. \tag{8.3b}$$

This is true because

$$\begin{aligned} K &= \max(K^{(l,k)}) = \max[\max(I_1^{(l)}, I_1^{(k)}) + \max(I_2^{(l)}, I_2^{(k)})] \\ &\geq \max(I_1^{(l_2)}, I_1^{(l_1)}) + \max(I_2^{(l_2)}, I_2^{(l_1)}) = I_1^{(l_2)} + I_2^{(l_1)}. \end{aligned} \tag{8.3c}$$

By the same considerations, there exist at most I PPs of degree $t > K$ without a Σ-representation. If a PP of degree t has two different Σ-representations from which the Σ-representation of a yet non-representable PP $x_1^{I_1^{(s+1)}} x_2^{I_2^{(s+1)}}$ can be obtained, then again $x_1^{I_1^{(s+1)}} x_2^{I_2^{(s+1)}}$ has at most degree K. Now if $I_1^{(s+1)} \geq I_1$ and $I_2^{(s+1)} \geq I_2$, then

$$\begin{aligned} &K \geq I_1^{(s+1,k)} + I_2^{(s+1,k)}, \qquad k = 1, 2, \dots, s; \\ &\quad I_j^{(s+1,k)} = \max(I_j^{(s+1)}, I_j^{(k)}), \qquad j = 1, 2. \end{aligned} \tag{8.4}$$

This means therefore that the new LCMs all have degree $\leq K$. We prove this as follows: Since $x_1^{I_1^{(s+1)}} x_2^{I_2^{(s+1)}}$ does not yet possess a Σ-representation, it must be true that $I_2^{(s+1)} < I_2^{(l_1)}$ and $I_1^{(s+1)} < I_1^{(l_2)}$. Then also

$$I_1^{(s+1,k)} + I_2^{(s+1,k)} \leq I_1^{(s+1,l_2)} + I_2^{(s+1,l_1)} = I_1^{(l_2)} + I_2^{(l_1)} \leq K, \quad (k = 1, 2, \dots, s), \tag{8.5}$$

if we take into account that up to degree K every polynomial of the basis, whose PP with the highest index is a multiple of another $x_1^{I_1^{(l)}} x_2^{I_2^{(l)}}$, was already eliminated, and therefore

$$\begin{aligned} I_1^{(k)} &\geq I_1; \qquad I_2^{(k)} < I_2^{(l_1)}; \qquad I_2^{(k)} \geq I_2; \qquad I_1^{(k)} < I_1^{(l_2)} \\ &\quad (k = 1, 2, \dots, s; \quad k \neq l_1, \quad k \neq l_2) \\ I_2^{(k)} &= I_2^{(l_1)} \text{ for } k = l_1, \qquad I_1^{(k)} = I_1^{(l_2)} \text{ for } k = l_2. \end{aligned} \tag{8.6}$$

Thus if $x_1^{I_1^{(s+1)}} x_2^{I_2^{(s+1)}}$ and one of the $x_1^{I_1^{(l)}} x_2^{I_2^{(l)}}$ have LCM with degree $> K$, then it must be that either $I_1^{(s+1)} < I_1$ or $I_2^{(s+1)} < I_2$. If we now redefine I_1 and I_2 as

$$I_1 = \min(I_1^{(l)}), \qquad l = 1, 2, \dots, s+1, \quad \text{and} \tag{8.7a}$$

$$I_2 = \min(I_2^{(l)}), \qquad l = 1, 2, \dots, s+1, \tag{8.7b}$$

then we can say that either I_1 or I_2 it must have been decreased. But this also means that for degree $t \geq K$ there are now fewer PPRs without Σ-representation than earlier. The new LCMs of $x_1^{I_1^{(s+1)}} x_2^{I_2^{(s+1)}}$ and $x_1^{I_1^{(l)}} x_2^{I_2^{(l)}}$ $(l = 1, 2, \dots, s)$ must have all appeared by degree $2K$. For degree $2K$, we can apply the same reasoning: Either, through a newly appearing relation, I_1 or I_2 will decrease, or every new LCM has already appeared before degree $2K$. In the first case, the new LCMs appear before degree $K+2K = 3K$. We can continue in this manner. However, $I = I_1+I_2$ can only decrease I times and thereby increase the bound.

Thus if $I = 1$, all of the LCMs ever computed appear before degree $K^{(1)} = 2K$; if $I = 2$, then before degree $K^{(2)} = K + K^{(1)} = 3K$; if $I = 3$, then before degree $K^{(3)} = K^{(1)} + K^{(2)} = 5K$. In this way, we continue recursively: If $I = l$, then the LCMs appear before degree $K^{(l)} = K^{(l-2)} + K^{(l-1)}$.

In order to obtain an explicit formula, we must make the estimate somewhat coarser, and say:

For $I = 1$, the LCMs appear before $K^{(1)} = 2K$.
For $I = 2$, the LCMs appear before $K^{(2)} = 2K^{(1)} = 2^2K$.
$\vdots$
For $I = l$, the LCMs appear before

$$K^{(l)} = 2^l K, \tag{8.8}$$

as an easy induction shows.

Of course in most cases the algorithm will have already terminated by much smaller degrees. Thus (8.8) has only theoretical value and states that for an arbitrary P-ideal $\subset K[x_1, x_2]$, the algorithm can certainly be terminated for specific predetermined degrees.

9. Programming the algorithm

In order to adapt the algorithm for electronic computers, we must first consider how polynomials can be computed in such devices. Then we will provide a rough flowchart for the algorithm as well as for the most important subroutine (the computation of a Σ-representation for a PPR), which is formulated as much as possible without regard to the special properties of a specific computer. In addition to this, we describe the two programs which we wrote for the ZUSE Z 23 V, in that we first provide the peculiarities of these two programs and then describe how the data must be input and in what form the results appear. In the flow charts a new symbol will be used: $a \leftarrow b$. This will mean: the new value for a results from the previous value for b, with the frequent use $a \leftarrow a + 1$, which means in words: the new value for a results from the previous value for a by addition of 1.

9.1. Computing with polynomials on a computer

Here we deal with first representing PPs and then with combining PPs to form polynomials. In order to represent PPs, two different ways were pursued. One orders each PP by the index arising from the order (4.1), and computes with this index. In this way, we only need a single cell to represent a PP, and we can, within certain limits, compute with arbitrarily many variables in arbitrarily high degree. But for the individual operations with PPs (such as multiplication, forming LCMs, etc.), we need time-consuming subroutines. The other way represents the exponents of a PP $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ as a single number. How this is done can be shown most quickly with an example. Suppose $n = 3$ and $i_1 = 2, i_2 = 1, i_3 = 4$, then the corresponding number would read 20104. In this way, exponents ≤ 99 can be handled, we reserve again only one cell for a PP, and the subroutines for the operations with PPs are substantially simplified (especially for multiplication of PPs: this happens now simply by adding the two corresponding numbers). Of course, we can compute with just a limited number of variables (in our case with five), because in one cell, only numbers with a limited number of digits can be represented.

For the first way, a basic formula will be given here, namely the one which, given the number n of variables and the exponents $i_1, i_2, \ldots, i_n$, computes the index $N(i_1, i_2, \ldots, i_n)$ coming from the PPs $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ in the order (4.1). It reads:

$$N(i_1, i_2, \ldots, i_n) = \sum_{\tau=0}^{t-1} H(\tau; n-1) + H(t; n-1) \tag{9.1}$$

$$-\sum_{j=1}^{n-1}\left(\sum_{\tau=t+2-\sum_{l=1}^{j} i_l}^{t+1-\sum_{l=1}^{j-1} i_l} H(\tau-1; n-1-j)\right),$$

$$\text{where } t=\sum_{j=1}^{n} i_j$$

with the definition

$$\sum_{\tau=k}^{l} m_\tau = 0 \qquad \text{for } l < k. \tag{9.1a}$$

The partial expression

$$\sum_{\tau=0}^{t-1} H(\tau; n-1) \tag{9.1b}$$

from (9.1) gives the number of PPs of degree $< t$ in n variables, so the rest yields the index of the PP within the considered degree t. We prove by induction on t and n.

First the formula is true for $t = 0$ and all n. For this case, the formula evaluates to

$$\begin{aligned} N(0,0,\dots,0) &= H(0; n-1) - \sum_{j=1}^{n-1} \sum_{\tau=0+2-0}^{0+1-0} H(\tau-1; n-1-j) \\ &= H(0; n-1) = 1, \end{aligned} \tag{9.2}$$

agreeing with fact that there is only one PP of degree 0 for all n and this has index 1.

Formula (9.1) is also true for $n = 1$ and all t. In this case, we have

$$N(i_1) = \sum_{\tau=0}^{t} H(\tau; 0) - 0 = \sum_{\tau=0}^{t} 1 = t+1, \tag{9.3}$$

agreeing with the fact that for $n = 1$ only one PP appears for each degree, and hence this PP obtains the index $t + 1$.

Now we suppose (9.1) holds for n and all τ as well as for $n + 1$ and all degrees τ up to degree t. We show that the formula holds also for $n + 1$ variables and degree $t + 1$. Let $i_1, i_2, \dots, i_{n+1}$ be given with $\sum_{j=1}^{n+1} i_j = t + 1$. The PPs of degree $t + 1$ consist of those with $i_1 \neq 0$, which are simply the PPs of degree t in $n + 1$ variables multiplied by x_1, and those with $i_1 = 0$, whose order is determined by the order of the PPs of degree $t + 1$ in the n variables $x_2, x_3, \dots, x_n$. In both cases, the formula applies by the induction hypothesis.

For the case $i_1 = 0$, we have

$$N(0, i_2, \dots, i_{n+1}) = N_1 + N_2 + N_3, \tag{9.4}$$

where

$N_1 = \sum_{\tau=0}^{t} H(\tau; n)$ is the number of PPs of degree $\tau \le t$ in $n + 1$ variables,

$N_2 = H(t, n)$ is the number of PPs of degree $t + 1$ in the $n + 1$ variables $x_1, x_2, \dots, x_{n+1}$ with $i_1 \neq 0$, and

$$N_3 = H(t+1; n-1) - \sum_{j=1}^{n-1} \left(\sum_{\tau=(t+1)+2-\sum_{l=2}^{j+1} i_l}^{(t+1)+1-\sum_{l=2}^{j} i_l} H(\tau-1; n-1-j) \right)$$

is the number of the PP within the PPs of degree $t+1$
in the $n+1$ variables $x_1, x_2, \ldots, x_{n+1}$ with $i_1 = 0$
(computed with (9.1)).

By the index substitution $k-1 = j$ and the subsequent replacement of k by j again, N_3 becomes

$$N_3 = H(t+1; n-1) - \sum_{j=2}^{n} \left(\sum_{\tau=(t+1)+2-\sum_{l=1}^{j} i_l}^{(t+1)+1-\sum_{l=1}^{j-1} i_l} H(\tau-1; n-j) \right).$$

Hence,

$$N(0, i_2, \ldots, i_{n+1}) = \sum_{\tau=0}^{t+1} H(\tau; n) - \sum_{j=1}^{n} \left(\sum_{\tau=(t+1)+2-\sum_{l=1}^{j} i_l}^{(t+1)+1-\sum_{l=1}^{j-1} i_l} H(\tau-1; n-j) \right),$$

because $i_1 = 0$ implies $\sum_{l=2}^{k} i_l = \sum_{l=1}^{k} i_l$, and

$$\sum_{\tau=(t+1)+2-\sum_{l=1}^{1} i_l}^{(t+1)+1-\sum_{l=1}^{0} i_l} H(\tau-1; n-1) = 0,$$

because of (9.1a).

In the case $i_1 \neq 0$, we have

$$N(i_1, i_2, \ldots, i_{n+1}) = M_1 + M_2, \tag{9.5}$$

where

$$M_1 = \sum_{\tau=0}^{t} H(\tau; n) \qquad \text{is the number of PPs of degree } \tau \leq t \text{ in } n+1 \text{ variables,} \tag{9.5a}$$

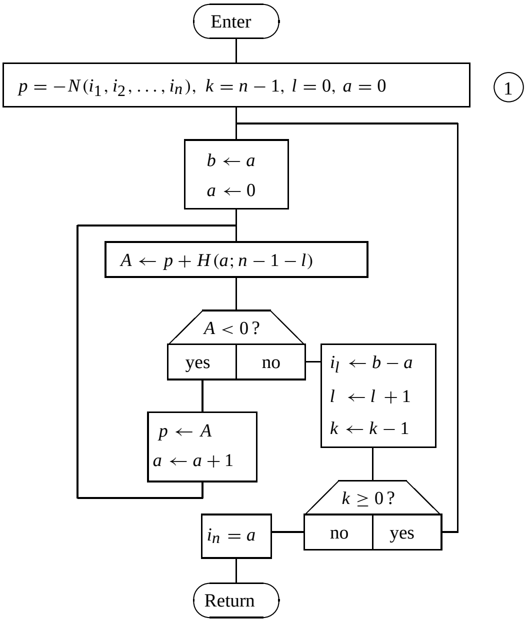
and the index of the PP $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ within degree $t+1$ is computed (with $i_1' = i_1 - 1$, $i_j' = i_j$ $(j = 2, 3, \ldots, n+1)$) as follows:

$$M_2 = N(i_1', i_2', \ldots, i_{n+1}') - \sum_{\tau=0}^{t-1} H(\tau; n) \tag{9.5b}$$

$$
\begin{aligned}
&= H(t;n) - \sum_{j=2}^{n} \left(\sum_{\tau=t+2-\sum_{l=1}^{j} i'_l}^{t+1-\sum_{l=1}^{j-1} i'_l} H(\tau-1;n-j) \right) \\
&\quad - \sum_{\tau=t+2-i'_1}^{t+1-0} H(\tau-1;n-1) \\
&= H(t+1;n) - \cancel{H(t+1;n-1)} \\
&\quad - \sum_{j=1}^{n} \left(\sum_{\tau=(t+1)+2-\sum_{l=1}^{j} i_l}^{(t+1)+1-\sum_{l=1}^{j-1} i_l} H(\tau-1;n-j) \right) \\
&\quad + \cancel{H((t+1)+1-1;n-1).}
\end{aligned}
$$

Therefore, combining (9.5a) and (9.5b) yields the correctness of (9.1) for the case $i_1 \neq 0$ as well.

The transformation of the indices back into $i_1, i_2, \ldots, i_n$ for given n is done algorithmically according to the following flowchart (here $-t$ will also be computed as i_0):



(9.6)

This flowchart holds for $n = 1$. Suppose that it holds for n variables and consider $p = -N(i_1, i_2, \ldots, i_{n+1})$. In the case of n variables, $i_1, i_2, \ldots, i_n$ will be computed starting from ①, after the computation of i_0 by the flowchart has set the variables to the following values: p is the negative of the index of $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ among the PPs of degree t, $l = 1$, $k = n-2$, $b = t$, $a = 0$. In the case of $n+1$ variables, t and i_1 are computed correctly, as a step-by-step precise execution according to the flowchart instructions will confirm. After that, the variables used at the mark ① have the following values: p is the negative of the index of $x_2^{i_2}x_3^{i_3}\cdots x_{n+1}^{i_{n+1}}$ among the PPs in the n variables $x_2, x_3, \ldots, x_{n+1}$ of degree $t - i_1$, $l = 2$, $k = n - 2$, $b =$ degree of the PP $x_2^{i_2}x_3^{i_3}\cdots x_{n+1}^{i_{n+1}}$, $a = 0$. By the induction hypothesis, the flowchart computes precisely $i_2, i_3, \ldots, i_{n+1}$, if we start with these values at ①. $l = 2$ just has the effect that the exponents still to compute obtain the indices $2, 3, \ldots, n + 1$, and the second argument $(n + 1) - 1 - l$ of $H(a; (n+1) - 1 - l)$ takes on the correct values in the case of the n variables $x_2, x_3, \ldots, x_{n+1}$. But this is precisely the effect required here.

Polynomial residue classes

$$f \overset{\text{def}}{=} x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n} + \sum a_{i_1i_2\cdots i_n}x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n} \equiv 0 \qquad (\mathcal{A})$$

(where $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ is the PP of f with the highest index; $a_{i_1i_2\cdots i_n} \in K$), are represented in the machine in the form

$$x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n} \equiv -\sum a_{i_1i_2\cdots i_n}x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n} \qquad (\mathcal{A}),$$

so that the number of PPs $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ with nonzero coefficients is stored in the first of several successive cells, $x_1^{I_1}x_2^{I_2}\cdots x_n^{I_n}$ (as index or number) in the next one, the first nonzero coefficient in the next, the associated PP (as index or number) in the following one, and so forth, up to the last nonzero coefficient and its associated PP. A specific order of the PPs $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ within the polynomial is not considered important, but for every operation with polynomial residue classes, the resulting residue class has stored again only PPs with nonzero coefficients, and so is "compactly expressed". K was taken to be the field of rational numbers. Numerators and denominators of a coefficient were stored in two separate cells so that every coefficient actually uses two cells. Some subroutines must manage the arithmetic operations between rational numbers that are represented in this way. The individual polynomials were ordered in rows of variable length. Indicators must be built in, which report excesses of this length (as well as excesses of every other limit that must be adhered to during computation, for example: the highest number of polynomials to be processed, the highest allowed range of numbers, the maximal number of stored basis elements, etc.), in order to avoid false results which arise from computing further. (These indicators are not shown in the flowcharts so that clarity is not impaired.)

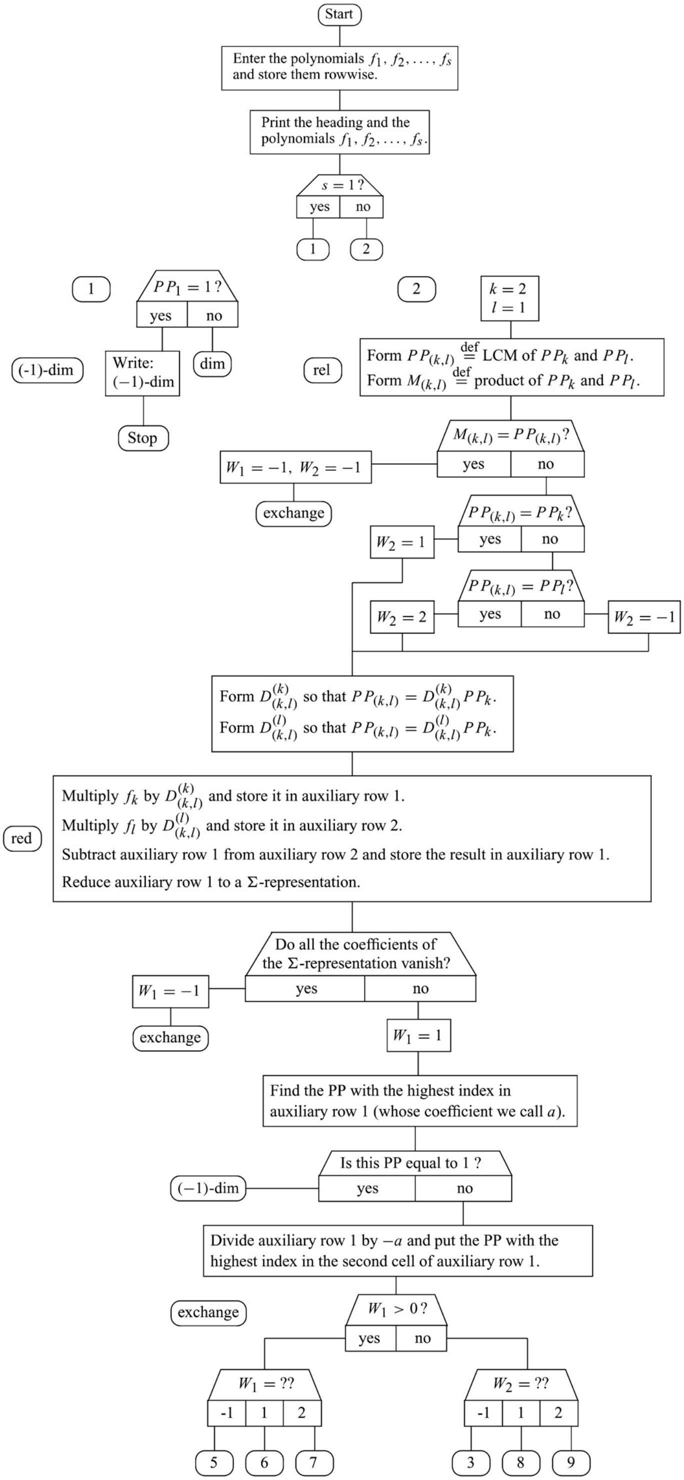
9.2. Flowchart of the algorithm

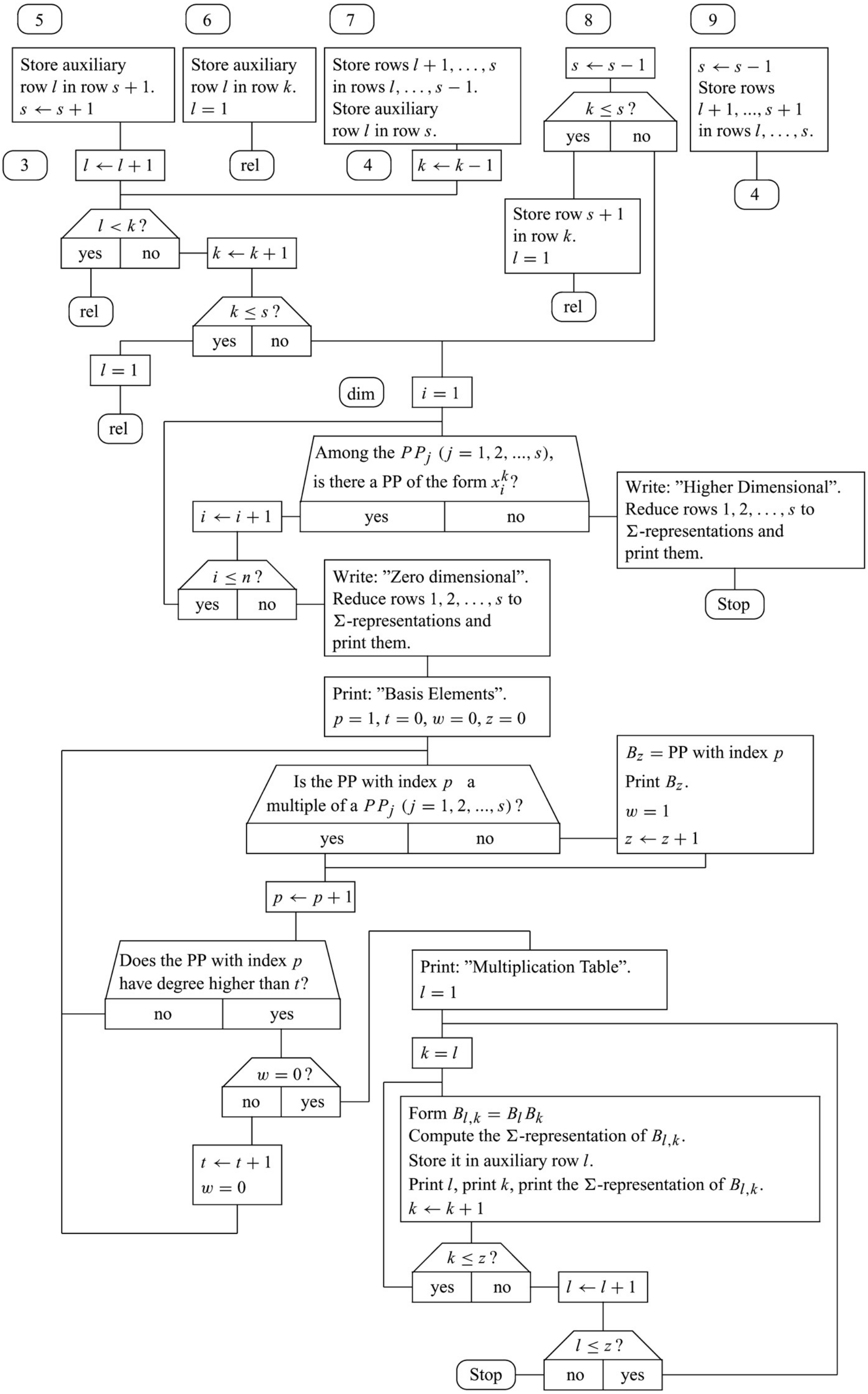
Let the ideal

$$\mathcal{A} = (f_1, f_2, \ldots, f_s) \subset K[x_1, x_2, \ldots, x_n],$$
$$f_j = x_1^{I_1^{(j)}}x_2^{I_2^{(j)}}\cdots x_n^{I_n^{(j)}} + \sum a^{(j)}_{i_1i_2\cdots i_n}x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n} = PP_j + \sum_{i=1}^{\tau_j} a_i^{(j)}PP_i^{(j)},$$

be given ($x_1^{I_1^{(j)}}x_2^{I_2^{(j)}}\cdots x_n^{I_n^{(j)}} = PP_j$ is the PP of f_j with the highest index; $a^{(j)}_{i_1i_2\cdots i_n} \in K$; $PP_i^{(j)}$ are the $x_1^{i_1}x_2^{i_2}\cdots x_n^{i_n}$ which have coefficients $a^{(j)}_{i_1i_2\cdots i_n} \neq 0$ written in any order).

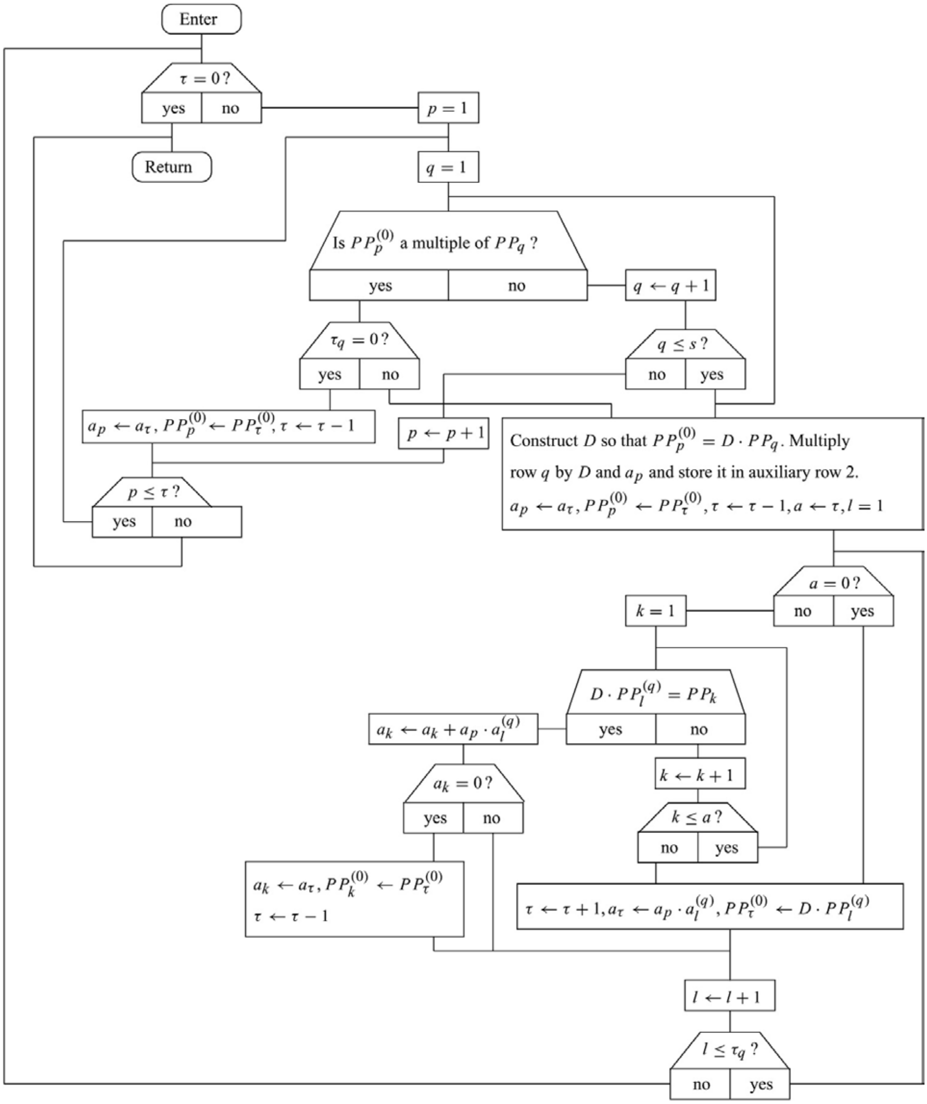
A rough flowchart for the algorithm in the form described in Section 6 is then as follows (auxiliary relations are not stored, the LCMs are used in the following order: the LCM of PP_k and PP_l precedes the LCM of PP_p and PP_q if $k < p$ or $k = p$ and $l < q$!):


Start
Enter the polynomials $f_1, f_2, \dots, f_s$ and store them rowwise.
Print the heading and the polynomials $f_1, f_2, \dots, f_s$.
$s = 1$?
yes
no
1
2
1
$PP_1 = 1$?
yes
no
(-1)-dim
Write: (-1)-dim
dim
Stop
2
$k = 2$
$l = 1$
rel
Form $PP_{(k,l)} \overset{\text{def}}{=}$ LCM of PP_k and PP_l.
Form $M_{(k,l)} \overset{\text{def}}{=}$ product of PP_k and PP_l.
$M_{(k,l)} = PP_{(k,l)}$?
yes
no
$W_1 = -1,\ W_2 = -1$
exchange
$PP_{(k,l)} = PP_k$?
yes
no
$W_2 = 1$
$PP_{(k,l)} = PP_l$?
yes
no
$W_2 = 2$
$W_2 = -1$
Form $D^{(k)}_{(k,l)}$ so that $PP_{(k,l)} = D^{(k)}_{(k,l)} PP_k$.
Form $D^{(l)}_{(k,l)}$ so that $PP_{(k,l)} = D^{(l)}_{(k,l)} PP_k$.
red
Multiply f_k by $D^{(k)}_{(k,l)}$ and store it in auxiliary row 1.
Multiply f_l by $D^{(l)}_{(k,l)}$ and store it in auxiliary row 2.
Subtract auxiliary row 1 from auxiliary row 2 and store the result in auxiliary row 1.
Reduce auxiliary row 1 to a Σ-representation.
Do all the coefficients of the Σ-representation vanish?
yes
no
$W_1 = -1$
exchange
$W_1 = 1$
Find the PP with the highest index in auxiliary row 1 (whose coefficient we call a).
Is this PP equal to 1 ?
yes
no
(-1)-dim
Divide auxiliary row 1 by $-a$ and put the PP with the highest index in the second cell of auxiliary row 1.
exchange
$W_1 > 0$?
yes
no
$W_1 = ??$
-1
1
2
5
6
7
$W_2 = ??$
-1
1
2
3
8
9


5
Store auxiliary row l in row $s+1$. $s \leftarrow s+1$
6
Store auxiliary row l in row k. $l=1$
7
Store rows $l+1, \ldots, s$ in rows $l, \ldots, s-1$. Store auxiliary row l in row s.
8
$s \leftarrow s-1$
$k \leq s$?
yes
no
9
$s \leftarrow s-1$ Store rows $l+1, ..., s+1$ in rows $l, \ldots, s$.
3
$l \leftarrow l+1$
rel
4
$k \leftarrow k-1$
4
Store row $s+1$ in row k. $l=1$
$l < k$?
yes
no
$k \leftarrow k+1$
rel
$k \leq s$?
yes
no
rel
$l=1$
rel
dim
$i=1$
Among the PP_j $(j=1,2,...,s)$, is there a PP of the form x_i^k?
yes
no
$i \leftarrow i+1$
Write: ”Higher Dimensional”. Reduce rows $1, 2, \ldots, s$ to Σ-representations and print them.
Stop
$i \leq n$?
yes
no
Write: ”Zero dimensional”. Reduce rows $1, 2, \ldots, s$ to Σ-representations and print them.
Print: ”Basis Elements”. $p=1, t=0, w=0, z=0$
Is the PP with index p a multiple of a PP_j $(j=1,2,...,s)$?
yes
no
B_z = PP with index p Print B_z. $w=1$ $z \leftarrow z+1$
$p \leftarrow p+1$
Does the PP with index p have degree higher than t?
no
yes
Print: ”Multiplication Table”. $l=1$
$k=l$
$w=0$?
no
yes
Form $B_{l,k} = B_l B_k$ Compute the Σ-representation of $B_{l,k}$. Store it in auxiliary row l. Print l, print k, print the Σ-representation of $B_{l,k}$. $k \leftarrow k+1$
$t \leftarrow t+1$ $w=0$
$k \leq z$?
yes
no
$l \leftarrow l+1$
$l \leq z$?
Stop
no
yes

9.3. Flowchart for the reduction of a representation of a PPR by other PPRS of lower index to a Σ-representation

In the auxiliary row 1, let the relation $PP_0 \equiv \sum_{i=1}^{\tau} a_i PP_i^{(0)}$ $(\mathcal{A})$ (PP_0 and $PP_i^{(0)}$ are power products, $a_i \neq 0$, $a_i \in K$ $(i = 1, 2, \ldots, \tau)$) be stored in the following form: $\tau, PP_0, a_1, PP_1^{(0)}, \ldots, a_\tau, PP_\tau^{(0)}$. Similarly in the rows $1, 2, \ldots, s$, the basis relations $PP_j \equiv \sum_{i=1}^{\tau_j} a_i^{(j)} PP_i^{(j)}$ $(\mathcal{A})$ (PP_j and $PP_i^{(j)}$ are power products, PP_j has a higher index than $PP_i^{(j)}$ $(i = 1, 2, \ldots, \tau_j)$, $a_i^{(j)} \neq 0$, $a_i^{(j)} \in K$ $(i = 1, 2, \ldots, \tau_j)$) are stored. Then the transformation of $\sum_{i=1}^{\tau} a_i PP_i^{(0)}$ to a Σ-representation is done by the following process:


Enter
τ = 0 ?
yes
no
Return
p = 1
q = 1
Is $PP_p^{(0)}$ a multiple of PP_q ?
yes
no
q ← q + 1
$\tau_q = 0$?
yes
no
q ≤ s ?
no
yes
$a_p \leftarrow a_\tau, PP_p^{(0)} \leftarrow PP_\tau^{(0)}, \tau \leftarrow \tau - 1$
p ← p + 1
Construct D so that $PP_p^{(0)} = D \cdot PP_q$. Multiply row q by D and a_p and store it in auxiliary row 2.
$a_p \leftarrow a_\tau, PP_p^{(0)} \leftarrow PP_\tau^{(0)}, \tau \leftarrow \tau - 1, a \leftarrow \tau, l = 1$
p ≤ τ ?
yes
no
a = 0 ?
no
yes
k = 1
$D \cdot PP_l^{(q)} = PP_k$
yes
no
$a_k \leftarrow a_k + a_p \cdot a_l^{(q)}$
k ← k + 1
$a_k = 0$?
yes
no
k ≤ a ?
no
yes
$a_k \leftarrow a_\tau, PP_k^{(0)} \leftarrow PP_\tau^{(0)}$
τ ← τ − 1
$\tau \leftarrow \tau + 1, a_\tau \leftarrow a_p \cdot a_l^{(q)}, PP_\tau^{(0)} \leftarrow D \cdot PP_l^{(q)}$
l ← l + 1
$l \leq \tau_q$?
no
yes

9.4. Description of the first program[2]

(See Appendix 1a for the program listing)

The first program for the ZUSE Z 23 V is, for the most part, written in the Formelübersetzer, an algorithmic programming language which corresponds approximately to ALGOL. However, the parts which are repeated often (rational arithmetic and operations with PPs) are programmed in the Freiburger code, which is similar to the internal language of the machine. The representation of PPs in this program is done by assigning natural numbers to the individual PPs using the formula (9.1) and algorithm (9.6). The use of the Formelübersetzer and this representation of the PPs proved to be too time consuming, given the speed of the machine used. Thus the program is practically worthless. Nevertheless, it is presented here because it treats the general case of *n* variables, and a program for faster machines, whose internal language is not accessible, should probably be written in a similar way. This program differs from the flowchart given above in two essential ways:

1. For every new run of the part $\boxed{\text{rel}}$, the representations of the PP_j $(j = 1, 2, \ldots, s)$ will be reduced each time to a Σ-representation.
2. In the part $\boxed{\text{red}}$, the auxiliary rows 1 and 2 will be reduced separately to Σ-representations and only then are subtracted from each other.

The second change, for the computing time, is a disadvantage. The first change can result in an advantage for complicated examples. In order to describe the usage of the program, it will be shown how the data tape and the output page look for the ideal:

$$\mathcal{A} = \left(x_3^2 - \frac{1}{2}x_1^2 - \frac{1}{2}x_2^2,\ \ x_1x_3 - 2x_3 + x_1x_2,\ \ x_1^2 - x_2\right) \tag{9.9}$$

(see Appendix 2).

9.5. Description of the second program

(See Appendix 1b for the program listing)

The second program was written entirely in the Freiburger code with the help of a subroutine that enables symbolic addressing (which, unlike the use of Formelübersetzer, has no negative influence on the computing time). Special care was taken to avoid waiting times on the slow magnetic drum. (In this way the computing time can be reduced by up to sixteen times in general.) In addition, the PPs were represented in the second form, namely as integers. With these three, as well as a couple of smaller improvements, we were able to reduce the computing time by a factor of 20–25 (and at the same time the usable memory was increased) so that computation with an electronic computer is quite profitable. Again we give the data tape and ouput page for the ideal (9.9) (see Appendix 3). The representation of the PPs as integers is slightly changed here, so that as the first part, the degree of the given PP is shown: Thus 2020000 is the PP x_1^2.

In both of these programs, the possibility exists, by adjusting a control key at the console to the value 17, of allowing the polynomials which are newly adjoined to the basis to be printed out

[2] In this section, references are made to five appendices in the thesis. These appendices are either program listings (in machine code or machine-oriented code for the ZUSE Z 23 V computer) or computer output of examples. We cannot print these appendices in this translation and refer the reader to the original thesis, which is also downloadable from the author's home page www.risc.uni-linz.ac.at/people/buchberger.

with an indication of the two power products PP_k and PP_l from which the $PP_{(k,l)}$ that produces the two different Σ-representations was formed. For the case of the ideal (9.9), this yields the following picture: Appendix 4.

In the second program, it is very easy to incorporate changes, in order to observe theoretical conjectures and considerations about the behavior of the algorithm in practical computation. Furthermore, it is possible to organize the memory differently, this means to store either many short or fewer long polynomials. For the ideal (9.9) the program needs 2 min, 46 s to find the basis (6.4) and another 59 s to compute the multiplication table. An additional 3 min 13 s are needed to output the results.

To conclude, we give another example of an ideal in $K[x_1, x_2, x_3, x_4, x_5]$ which will be found to be higher dimensional (see Appendix 5).

10. Conclusion

After showing in Section 3 that the residue class ring of a zero dimensional P-ideal has the structure of a hypercomplex system, in Section 4, step by step we introduced an algorithm, about which one can prove that it does, indeed, construct a basis for the algebra. 4.14 and 4.19 were derived as termination criteria for the algorithm in this form. Using four lemmas from Section 5, which make assertions about the occurrence of new relations between residue classes during the execution of the algorithm, the algorithm can be simplified in Section 6 and put into a somewhat different form (so that it specializes in the case of a single variable x to the Euclidean algorithm for determining the greatest common divisor of several polynomials). Results from Section 5 were applied in Section 7 for calculating the Hilbert function of an arbitrary P-ideal, and in Section 8 for determining a bound for the termination of the algorithm for the basis polynomials $f_1, f_2, \ldots, f_s$ of an arbitrary P-ideal $\subset K[x_1, x_2]$. Finally in Section 9, preparations for programming the algorithm were done, the most important flowcharts were provided, and finally examples were computed with the programs which we wrote for the ZUSE Z 23.

Acknowledgements

My sincere thanks go to Professor Wolfgang Gröbner for guiding this work. I also thank my colleagues at the computing center of the University of Innsbruck, Dr. H. Knapp and G. Margreiter, for several valuable programming suggestions.

References

Gröbner, W., 1949. Moderne Algebraische Geometrie. Springer-Verlag.
Van der Waerden, B.L., 1937. Moderne Algebra I, 2nd ed. Springer-Verlag.

Bruno Buchberger was born on 22 October 1942 in Innsbruck as the son of the policeman Joseph Buchberger and his wife Katherina. I attended elementary school and one year at the Hauptschule in Innsbruck, and then transferred in the second year to the I. Bundes-Realgymnasiums in Innsbruck where I graduated on 2 June 1960. Since the winter semester of 1960–61, I have been enrolled at the local Leopold-Franzens University in the fields of mathematics (major) and experimental physics (minor). I owe my education to Professor Gröbner for courses and guidance in scientific work, to Professors Schatz and Lochs for courses in mathematics, and to Professors Steinmaurer and Kolb for courses in physics. Since 1 July 1964, I have been employed as a research assistant at the computing center (Institute for Theoretical Physics).