Computational Algebra

4. Multivariate division

Now that we have at hand the concept of term order and of leading term, we can mimick the algorithm for division of univariate polynomials into the multivariate setting.

We will immediately consider the division of a polynomial by a *tuple* of polynomials since we have in mind the goal that, given $f \in k[x_1, \dots, x_n]$ and $(g_1, \dots, g_s) \in (k[x_1, \dots, x_n])^s$, we want to produce $(q_1, \dots, q_s) \in (k[x_1, \dots, x_n])^s$ and $p \in k[x_1, \dots, x_n]$ so that

$$f = q_1 \cdot g_1 + \dots + q_s \cdot g_s + p$$

Consider for example the term order Lex on $\mathbb{T}(x, y)$ and the polynomials

$$f = x_1^2 x_2 + x_1 x_2^2 + x_2^2 \qquad g_1 = x_1 x_2 - 1 \qquad g_2 = x_2^2 - 1$$

$$\begin{array}{l}
\underline{x_1^2 x_2} + x_1 x_2^2 + x_2^2 \\
x_1^2 x_2 - x_1 \\
\hline
\qquad \underline{x_1 x_2^2} + x_2^2 + x_1 \\
\qquad x_1 x_2^2 - x_2 \\
\hline
\underline{x_1} + x_2 + x_2^2 \\
\qquad x_2 + \underline{x_2^2} \\
\qquad\qquad x_2^2 - 1 \\
\hline
\qquad \underline{x_2} + 1 \\
\qquad\quad \underline{1}
\end{array}
\qquad
\left|\begin{array}{l}
g_1 \cdot (x_1 + x_2) \\
g_2 \cdot 1
\end{array}\right.
\qquad p = x_1 + x_2 + 1$$

We generalize the previous procedure into the following algorithm

<u>Algorithm</u> MultivariateDivision

Input: $f\in k[x_1,\dots,x_n]$, $(g_1,\dots,g_s)\in (k[x_1,\dots x_n])^s$, $\preccurlyeq$ on $\mathbb{T}^n$, $\{g_i\}$ nonzero

Output: $(q_1,\dots,q_s)\in (k[x_1,\dots,x_s])^s$, $p\in k[x_1,\dots x_n]$ such that

$$f = q_1 g_1 + \dots + q_s g_s + p$$

1. Set $q_1 := 0, \dots, q_s := 0$, $p := 0$, $v := f$
2. While $f \neq 0$:
3. While $LT_{\preccurlyeq}(f)$ is divisible by some of $LT_{\preccurlyeq}(g_1),\dots, LT_{\preccurlyeq}(g_s)$:
4. Find the smallest $i\in\{1,\dots,s\}$ such that $LT_{\preccurlyeq}(g_i)$ divides $LT_{\preccurlyeq}(f)$.
5. $q_i := q_i + \dfrac{LM_{\preccurlyeq}(v)}{LM_{\preccurlyeq}(g_i)}$, $v := v - \dfrac{LM_{\preccurlyeq}(v)}{LM_{\preccurlyeq}(g_i)}\cdot g_i$
6. $p := p + LM(v)$, $v := v - LM_{\preccurlyeq}(v)$
7. Return $(q_1,\dots,q_s)$ and p.

<u>Theorem</u>: MultivariateDivision stops after finitely many steps and delivers the output as in the specification.

Proof: notice that at each step in MultivariateDivision the following holds:

$$f = q_1 g_1 + \dots + q_s g_s + p + v \qquad (\square)$$

in fact, the two steps in which the quantities that are involved in $(\square)$ change are Step 5 and Step 6, which by a simple computation preserve ($\square$); hence, if the algorithm terminates, it gives the output that is specified; now, notice that in both steps 5 and 6 the leading term of v decreases, and since $\leqslant$ is a well-order, the variable v must reach 0 after finitely many steps.

Actually, we notice that the algorithm just returning $q_1 = \dots = q_s = 0$ and $p = f$ satisfies the same output requirement of MultivariateDivision. There must be more, and this is indeed the case.

<u>Theorem</u>: the output of MultivariateDivision satisfies the following properties:

a. no element of $\mathrm{supp}(p)$ is contained in $\langle LT_\leqslant(g_1), \dots, LT_\leqslant(g_s)\rangle$

b. if $q_i \neq 0$ for some $i \in \{1, \dots, s\}$, then $LT_\leqslant(q_i g_i) \leqslant LT_\leqslant(f)$

c. for all $i \in \{1, \dots, s\}$ and for all $t \in \mathrm{supp}(q_i)$, we have

$$t \cdot LT_\leqslant(g_i) \notin \langle LT_\leqslant(g_1), \dots, LT_\leqslant(g_{i-1})\rangle$$

moreover, the output $(q_1, \dots, q_s) \in (k[x_1, \dots, x_n])^s$ and $p \in k[x_1, \dots, x_n]$ is the unique pair determined by the properties a., b., c. starting from $f \in k[x_1, \dots, x_n]$ and $(g_1, \dots, g_s) \in (k[x_1, \dots, x_n])^s$

Proof. a. this condition is satisfied because in the algorithm, at step 6, a term is added to p precisely when it does not belong to $\langle LT_\leqslant(g_1), \dots, LT_\leqslant(g_s)\rangle$.

b. we prove this condition by induction, by showing that we always have $LT_\leq(v) \leq LT_\leq(f)$ and $LT_\leq(q_i \cdot g_i) \leq LT_\leq(f)$ whenever it holds $q_i \neq 0$; these conditions, in fact, are true at the beginning of the algorithm; when Step 5 or Step 6 is performed, then $LT_\leq(v)$ decreases, so the first condition still holds true; when Step 5 is performed and both the old and new values of q_i are not zero, we have

$$LT_\leq\left(\left(q_i + \frac{LM_\leq(v)}{LM_\leq(g_i)}\right)\cdot g_i\right) \leq \max\{LT_\leq(q_i g_i), LT_\leq(v)\} \leq LT_\leq(f)$$

moreover, also when the old value of q_i is zero the inequality holds true, hence b. is always satisfied

c. this condition is always true, because in the only step when q_i is modified, namely Step 5, we have that a monomial having a term $t \in \mathbb{T}^n$ is added to q_i only when $t \cdot LT_\leq(g_i)$ was not already eliminated in an earlier round of the loop, namely only when $t \cdot LT_\leq(g_i) \notin (LT_\leq(g_1), \dots, LT_\leq(g_{i-1}))$

to show uniqueness we suppose that

$$f = q_1 g_1 + \dots + q_s g_s + p = q_1' g_1 + \dots + q_s' g_s + p'$$

and that both these two representations satisfy conditions a., b., and c.; then, we have

$$(q_1-q_1')\,g_1+\dots+(q_s-q_s')\cdot g_s+(p-p')=0 \qquad (\Delta)$$

from condition a. we have that

$$LT_\leq(p-p')\notin(LT_\leq(g_1),\dots,LT_\leq(g_s))$$

and condition c. implies that for all $i\in\{1,\dots,s\}$,

$$LT_\leq((q_i-q_i')\cdot g_i)\notin(LT_\leq(g_1),\dots,LT_\leq(g_{i-1}))$$

hence the leading terms of all summands of (Δ) are distinct; however, looking at the rules for the leading terms of sums of polynomials, we see that this is only possible when

$$q_1=q_1',\ \dots,\ q_s=q_s',\qquad p=p'.$$

Good! So now we have an algorithm that mimics the one of univariate polynomials. Notice that in the specification of the input, the polynomials $g_1,\dots,g_s$ are given as a <u>tuple</u>, and not as a set. This is necessary, as the following example shows:

<u>Example</u>: let us consider the example we had at the beginning, but with the roles of g_1 and g_2 swapped, namely, we consider

$$f=x_1^2x_2+x_1x_2^2+x_2^2 \qquad g_1'=x_2^2-1 \qquad g_2'=x_1x_2-1 \qquad \leq\,=\text{Lex}$$

then we have

$$f=g_2'(x_1+1)+g_2'\cdot x_1+(2x_1+1)$$

hence we see that the algorithm strongly depends on the order of the $\{g_i\}_{i=1}^s$

At least, we know that if we are given f and $g_1, \dots, g_s$ and we perform MultivariateDivision and obtain $p=0$, then f belongs to the ideal generated by $g_1, \dots, g_s$. Unfortunately, the converse is not true.

Example: consider the situation

$$f = x_1x_2^2 - x_1 \qquad g_1 = x_1x_2 + 1 \qquad g_2 = x_2^2 - 1 \qquad \leq = Lex$$

then MultivariateDivision outputs

$$f = g_1 \cdot x_2 + g_2 \cdot 0 + (-x_1 - x_2)$$

on the other hand, $f = x_1 \cdot g_2$, so $f \in (g_1, g_2)$.

So we need to improve our tools if we want a procedure that, as in the univariate case, allows us to algorithmically decide whether a polynomial belongs to an ideal. We will see that Gröbner bases provide an answer, but for that we need a long detour. For the moment, let us close with one definition:

Def: let $f \in k[x_1, \dots, x_n]$ and let $(g_1, \dots, g_s) \in (k[x_1, \dots, x_n])^s$ be non-zero polynomials and let $p \in k[x_1, \dots, x_n]$ be as in the output of MultivariateDivision; let $G := (g_1, \dots, g_s)$, then we say that p is the normal remainder of f with respect to G, and we denote $p = NR_{\leq, G}(f)$