**Opgave Computeralgebra, week 9: PolynomialReduce** *(zie boek, §21.2)*

Laat $\mathbb{N} := \{0, 1, 2, \ldots\}$. Neem op $\mathbb{N}^n$ een totale ordening (genoteerd $\prec$) zo dat:

(a) Als $\alpha, \beta, \gamma \in \mathbb{N}^n$ en $\alpha \prec \beta$ dan $\alpha + \gamma \prec \beta + \gamma$.

(b) Elke deelverzameling van $\mathbb{N}^n$ heeft een mininaal element t.o.v. $\prec$ .

Een voorbeeld is de *lexicografiche ordening*, d.w.z. dat $(\alpha_1, \alpha_2, \ldots, \alpha_n) \prec (\beta_1, \beta_2, \ldots, \beta_n)$ desda er is een $i$ met $\alpha_j = \beta_j$ als $j = 1, 2, \ldots, i-1$ en met $\alpha_i < \beta_i$.

Zij nu $F$ een lichaam (bijv. $\mathbb{Q}$, $\mathbb{R}$ of $\mathbb{C}$). Neem $f \in F[x_1, \ldots, x_n]$, dus $f$ is een polynoom in de variabelen $x_1, \ldots, x_n$, dus

$$f = \sum_{\alpha \in \mathbb{N}^n} c_\alpha x^\alpha = \sum_{(\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n} c_{\alpha_1, \ldots, \alpha_n} x_1^{\alpha_1} \ldots x_n^{\alpha_n}$$

met $c_\alpha \in F$ en met $c_\alpha \neq 0$ voor slechts eindig veel $\alpha$. Dan is er onder de $\alpha \in \mathbb{N}^n$ met $c_\alpha \neq 0$ een maximaal element t.o.v. de ordening $\prec$. Dit element van $\mathbb{N}^n$ noemen we de *maximale graad* van $f$, genoteerd $\mathrm{mdeg}(f)$. Dan spreken we verder van:

*leading coefficient*: $\mathrm{lc}(f) := c_{\mathrm{mdeg}(f)}$;

*leading monomial*: $\mathrm{lm}(f) := x^{\mathrm{mdeg}(f)}$;

*leading term*: $\mathrm{lt}(f) := c_{\mathrm{mdeg}(f)}\, x^{\mathrm{mdeg}(f)}$.

Het algoritme **PolynomialReduce** voor *deling met rest in meer variabelen* heeft als input polynomen $f, f_1, \ldots, f_s \in F[x_1, \ldots, x_n]$ en als output polynomen $q_1, \ldots, q_s, r \in F[x_1, \ldots, x_n]$ zo dat

(a) $f = q_1 f_1 + \cdots + q_s f_s + r$ ;

(b) Voor geen enkele $i$ is $\mathrm{lm}(f_i)$ een deler van een monomial in $r$.

In pseudocode luidt het algoritme:

$r := 0;\quad p := f;\quad q_1, q_2, \ldots, q_s := 0;$

`while` $p \neq 0$ `do`

`if` $\exists j\ \mathrm{lt}(f_j) \mid \mathrm{lt}(p)$ (d.w.z. dat $\mathrm{lt}(f_j)$ een deler is van $\mathrm{lt}(p)$)

`then` $i := \min\{j \colon \mathrm{lt}(f_j) \mid \mathrm{lt}(p)\}$;

$$q_i := q_i + \frac{\mathrm{lt}(p)}{\mathrm{lt}(f_i)}\,;$$

$$p := p - \frac{\mathrm{lt}(p)}{\mathrm{lt}(f_i)}\, f_i\,;$$

`else` $r := r + \mathrm{lt}(p);\quad p := p - \mathrm{lt}(p);$

`return`$(q_1, \ldots, q_n, r)$

**Opgave**

(i) Programmeer in Mathematica het bovenstaande algoritme voor $F := \mathbb{Q}$ en met $\prec$ de lexicografische ordening.

(ii) Test je programma met de inputs van Example 21.9 en Example 21.10. Vergelijk je testresultaten ook met de Mathematica-functie

$$\textbf{PolynomialReduce}[f, \{f_1, \ldots, f_s\}, \{x_1, \ldots, x_n\}, \\ \textbf{MonomialOrder} \rightarrow \textbf{Lexicographic}, \textbf{CoefficientDomain} \rightarrow \textbf{Rationals}]$$

(iii) Test je programma met random inputs voor $f, f_1, \ldots, f_s$, waarbij $n$, $s$, de graden van de polynomen en de (geheeltallig te nemen) coëfficiënten van de polynomen voldoende klein worden gehouden om binnen een redelijke tijd output te krijgen. Ga het effect na van verwisseling van $f_1, \ldots, f_s$. Laat Mathematica controleren of de output voldoet aan eisen (a) en (b).