Publ. I.R.M.A. Strasbourg, 1984, 229/S–08
Actes 8[e] Séminaire Lotharingien, p. 59–62

ÜBER SYMMETRISCHE POLYNOME

VON

HEINZ LÜNEBURG

Es sei R ein kommutativer Ring mit 1 und $R[x_1, \ldots, x_n]$ sei der Polynomring in den Unbestimmten $x_1, \ldots, x_n$ über R. Ist Σ der Teilring aller symmetrischen Polynome aus $R[x_1, \ldots, x_n]$, so ist $R[x_1, \ldots, x_n]$ en freier Σ-Modul und die Elemente $x_2^{e_1} \ldots x_n^{e_{n-1}}$ mit $0 \le e_i \le i$ bilden eine freie Σ-basis dieses Moduls. Setzt man $E(n) = \{e \mid 0 \le e_i \le i$ für $i = 1, \ldots, n-1\}$, so gibt es also zu jedem $f \in R[x_1, \ldots, x_n]$ eindeutig bestimmte $S_e \in \Sigma$ mit

$$f = \sum_e S_e x_2^{e_1} \ldots x_n^{e_{n-1}} \qquad \big(e \in E(n)\big).$$

Wir geben hier einen Algorithmus an, der zu f die S_e berechnet. Für diesen Algorithmus benötigen wir die Prozedur naechst(e, initium, ekm, i), die alle $e \in E(i)$ erzeugt. Sie wird mit initium = true gestartet. Nach dem ersten Aufruf ist initium = false. Ist $i = 1$, so wird beim ersten Aufruf $e = \emptyset$ und ekm = false ausgegeben. Ist $i > 1$, so wird beim ersten Aufruf etwa $e = (0, \ldots, 0)$ ausgeben und bei jedem weiteren Aufruf der Nachfolger des bislang aktuellen e. Ferner ist ekm = true, solange noch nicht alle $e \neq E(i)$ erzeugt sind.

Algorithmus SYMKOEF

Berechnet zu $f \in R[x_1, \ldots, x_n]$ eine Familie $\big(S_e \mid e \in E(n)\big)$ von symmetrischen Polynomen S_e mit

$$f = \sum_e S_e x_2^{e_1} \ldots x_n^{e_{n-1}} \qquad \big(e \in E(n)\big).$$

begin $S_0 := f\,;\ i := 1\,;$
  while $i < n$ **do**
  begin *for* $k := 0$ *to* i **do**
    for $j := k$ *to* i **do**
    $p_{jk} = \sum_r x_{i+1}^{r_1} x_i^{r_2} \ldots x_{i-k+2}^{r_k} x_{i-k+1}^{j-k-r_1-\cdots-r_k}\,;$

```
{Die Summation ist über alle r mit 0 ≤ r_1 ≤ j − k,
0 ≤ r_2 ≤ j − k − r_1, ..., 0 ≤ r_k ≤ j − k − r_1 − ··· − r_{r−1}
zu erstrecken .}
initium := true; ekm := true;
while ekm do
begin naechst(e, initium, ekm, i);
    T_0 := S_e;
    if T_0 = 0 then for k := 0 to i do S_{e,k} := 0 else
    begin for k := 1 to i do
        T*_{k−1} := T_{k−1}(..., x_{i−k+2}, x_{i−k+1}, ...);
        T_k := (T_{k−1} − T*_{k−1})/(x_{i−k+2} − x_{i−k+1})
        S_{e,i} := T_i;
        for k := i − 1 down to 0 do
        S_{e,k} := T_k − Σ_{j=k+1}^{i} p_{jk} S_{e,j}
    end
  end
  i := i + 1
end
end
```

Will man dies wirklich programmieren, so wird man bei der Berechnung der T_k von der Formel

$$x^a y^b - x^b y^a = (y - x) \sum_{i=0}^{b-a-1} y^{a+i} x^{b-1-i},$$

die im Falle $a \leq b$ gilt, und von der Formel

$$x^a y^b - x^b y^a = (x - y) \sum_{i=0}^{a-b-1} x^{b+i} y^{a-1-i},$$

die für $a > b$ gilt, Gebrauch machen. Ferner wird man die p_{jk} nicht explizit ausrechnen and abspeichern. Sie werden ja nur einmal benötigt, so daß es genügt, sich bei der Berechnung von $S_{e,k}$ sukzessive die in p_{jk} vorkommenden Monome zu verschaffen. Dabei ist es nützlich zu wissen, daß durch

$$(r_1, \dots, r_n) \to (r_1, r_1 + r_2 + 1, \dots, r_1 + r_2 + \cdots + r_k + k - 1)$$

eine Bijektion der Menge der in $p_{j,k}$ vorkommenden r auf die Menge der k-Teilmengen von $\{0, \dots, n + k - 1\}$ definiert wird.

Ist $f \in \mathbf{Z}[x]$ und ist der Leitkoeffizient von f gleich 1, so gilt für die Diskriminante $\mathrm{Dis}(f)$ nach einem Satz von STICKELBERGER, daß $\mathrm{Dis}(f) = 0$ oder $1 \pmod 4$ ist. SCHUR bewies diesen Satz, indem er zeigte, daß es ganze Zahlen a und b gibt mit $\mathrm{Dis}(f) = a^2 - 4b$. Hierin sind a und b natürlich nicht eideutig bestimmt. Das von SCHUR benutzte a war nun nichts anderes als die Permanente der Vandermondematrix aus den Nullstellen von f. Nun ist diese Permanente eine symmetrische Funktion in den Nullstellen von f, so daß sie nach dem Waringschen Satz ein Polynom in den Koeffizienten von f ist. Es erhebt sich daher die Frage, ob man diese Permanente ohne explizite Kenntnis der Nullstellen fon f berechnen kann. Dies ist in der Tat möglich, wie man folgendermaßen einsehen kann.

Es seien $x_1, \ldots, x_n$ Überstimmte über $\mathbf{Z}$. Ferner sei

$$\mathrm{Vdm}(x_1, \ldots, x_n) = \begin{vmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-1} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^{n-1} \end{vmatrix}.$$

Dann ist

$$\mathrm{perm}\big(\mathrm{Vdm}(x_1, \ldots, x_n)\big) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_i^{\sigma(i)-1}.$$

Dies ist ein symmetrisches Polynom in den x_i. Dieses Polynom kann man nun mit Hilfe bekannter Algorithmen als Polynom in den elementarsymmetrischen Polynomen der x_i darstellen. Da die elementarsymmetrischen Funktionen in den Nullstellen von f nach dem Vietaschen Wurzelsatz bis aufs Vorzeichen, das man auch beherrscht, die Koeffizienten von f sind, ist unsere Frage bejahend beantwortet. Dieser Algorithmus ist praktisch nur für sehr kleine n durchführbar, wie im übrigen auch der nächste, der jedoch nicht ganz so aufwendig ist.

Wir definieren im Polynomring $\mathbf{Z}[y_1, \ldots, y_{n-1}]$ rekursiv Polynome $f_2, \ldots, f_n$ vermöge $f_2 = y_1$ und

$$f_{k+1} = y_k^k \sum_\alpha r_\alpha \,(y_1^{\alpha_1} - y_k^{\alpha_1}) \cdot \big(y_{k-1}^{\alpha_{k-1}} - y_k^{\alpha_{k-1}}\big).$$

falls

$$f_k = \sum_\alpha r_\alpha \, y_1^{\alpha_1} \ldots y_{k-1}^{\alpha_{k-1}}$$

ist. Ist dann

$$f_n = \sum_\beta r_\beta \, y_1^{\beta_1} \ldots y_{n-1}^{\beta_{n-1}},$$

so ist

$$\mathrm{perm}\big(\mathrm{Vdm}(x_1, \ldots, x_n)\big) = \sum_\beta r_\beta \, \tau_{\beta_1}^{(n)} \ldots \tau_{\beta_{n-1}}^{(n)}.$$

Dabei ist $\tau_0^{(n)} = 1$ und $\tau_i^{(n)} = \sum_{j=1}^{n} x_j^i$, falls $i \geq 1$ ist. Man beachte, daß die Definition von $\tau_0^{(n)}$ von der üblichen Definition abweicht. Üblich ist, $\tau_0^{(n)} = n$ su setzen. Die $\tau_i^{(n)}$ kann man nun mit Hilfe der Newtonschen Formeln durch die elementarsymmetrischen Polynome in den $x_1, \ldots, x_n$ ausdrücken, so daß wir eine zweite Lösung unserer Aufgabe gefunden haben.

Analysiert man die Entstehung der Monome in f_n, so folgt $\sum_\beta |r_\beta| = \sum_{k=1}^{n-1} a(n,k)$, wobei die $a(n,k)$ durch folgende Rekursion definiert sind : $a(m,0) = 0$ für alle m und $a(2,1) = 1$. Ferner

$$a(m+1, i+1) = \sum_{k=i}^{m-1} \binom{k}{i} a(m,k).$$

Meine Vermutung lautete, daß $\sum_{k=1}^{n-1} a(n,k) = (n-1)!$ ist. Die Herren Nicoletti und Comtet bemerkten an Hand einer kleinen Tabelle der $a(n,k)$, daß $a(n,k) = c(n-1,k)$ sein müsse, wobei $c(n-1,k)$ die vorzeichenlosen Stirlingzahlen sind. Dies konnte ich mittlerweile verifizieren, so daß in der Tat

$$\sum_\beta |r_\beta| = \sum_{k=1}^{n-1} a(n,k) = (n-1)!$$

ist. Einen anderen Beweis dieser Tatsache schickte mir Herr E. Triesch (Aachen). Der zweite Algorithmus ist also ebenfalls sehr aufwendig.

Eine Note mit detaillierten Beweisen ist in Vorbereitung.

Heinz Lüneburg,
Fachbereich Mathematik
der Universität,
Pfaffenbergstraße 95,
D-6750 Kaiserslautern.