Maple-Praktikum für Lehramt 2017 - Blatt 12

Dieses Blatt wird in Kalenderwoche 30 (ab 24. Juli) testiert.

Aufgaben: 4

```
> restart;
  with(LinearAlgebra);
```

Diese Woche beschäftigen wir uns mit der Smith-Form und de Hauptsatz über endlich erzeugte abelsche Gruppen.

ÜBUNG [01]:

Geben Sie die folgenden Definitonen an:
1.) Basis eines $\mathbb{Z}$-Moduls,
2.) Äquivalenz von Matrizen und
3.) Smith-Form.

Zunächst berechnen wir einige Smith-Formen.

ÜBUNG [02]:

1.) Bestimmen Sie die Smith-Formen von $A = \begin{bmatrix} 0 & -1 & 1 & 1 \\ 6 & -4 & 6 & 4 \\ 4 & -5 & 3 & 3 \\ 6 & -5 & 5 & 3 \end{bmatrix}$ und $B = \begin{bmatrix} 12 & 20 & 30 \\ 0 & 4 & 8 \\ 0 & 16 & 4 \end{bmatrix}$ per Hand und mit dem Befehl SmithForm.
2.) Schreiben Sie ein Programm, das eine Matrix $M \in \mathbb{Z}^{n \times n}$ als Eingabe erhält und die Smith-Form von M berechnet. Dabei können Sie die Hilfsfunktionen findMin, dividesAll und dividesFirst benutzen. Testen Sie Ihr Programm mit den Matrizen aus 1.)
3.) Wie kann man mithilfe der Smith-Form erkennen, ob zwei gegebene Matrizen äquivalent sind?

```
> findMin:=proc(M::Matrix,n::posint,o::posint)
    # gibt die Indizes des betragsmäßig kleinsten Eintrages der
  Matrix (ab Zeile und Spalte o) zurück, der nicht 0 ist
    # M: Matrix
    # n: Zeilen- und Spaltenzahl der Matrix
```

```
  # o: Zeile und Spalte, ab der die Suche beginnt
 local i,j,minI,minJ;
 minI := 0;  # Default-Werte, wenn sie nicht geändert werden,
gibt es nur noch 0en
 minJ := 0;

 for i from o to n do
   for j from o to n do
     if (M[i,j] <> 0) then  # 0-Einträge ignorieren
       if (minI=0 and minJ=0) or (abs(M[i,j]) < abs(M[minI,minJ]
)) then
         minI := i;
         minJ := j;
       end if:
     end if:
   end do:
 end do:

 return minI,minJ;
end proc:

dividesAll:=proc(M::Matrix,n::posint,o::posint)
  # prüft, ob Eintrag (o,o) alle Einträge ab Zeile/Spalte o
teilt
  # falls ein Eintrag nicht geteilt wird, weden dessen Indizes
zurückgegeben, ansonsten true
  # M: Matrix
  # n: Zeilen- und Spaltenzahl der Matrix
  # o: Zeile und Spalte, ab der die Suche beginnt
 local i,j;

 for i from o to n do
   for j from o to n do
     if irem(M[i,j],M[o,o]) <> 0 then
       return i,j;
     end if;
   end do:
 end do:

 return true;
end proc:

dividesFirst:=proc(M::Matrix,n::posint,o::posint)
  # prüft, ob Eintrag (o,o) alle Einträge in der o-ten Zeile
und o-ten Spalte teilt
  # falls ein Eintrag nicht geteilt wird, weden dessen Indizes
zurückgegeben, ansonsten true
  # M: Matrix
```

```
    # n: Zeilen- und Spaltenzahl der Matrix
    # o: Zeile und Spalte, ab der die Suche beginnt
   local i;

   for i from o to n do
    if irem(M[i,o],M[o,o]) <> 0 then  # Spalte durchgehen
       return i,o;
     end if;
    if irem(M[o,i],M[o,o]) <> 0 then  # Zeile durchgehen
       return o,i;
     end if;
   end do:

   return true;
 end proc:
```

Um die endlich erzeugten abelschen Gruppen klassifizieren zu können, untersuchen wir ihren Zusammenhand mit den endlich erzeugten $\mathbb{Z}$-Moduln.

ÜBUNG [03]:

1.) Sei $(G,+)$ eine abelsche Gruppe. Für $g \in G$ und $z \in \mathbb{Z}$ definieren wir

$z \cdot g := \begin{cases} \sum_{i=1}^{z} g & z \geq 0 \\ \sum_{i=1}^{-z} -g & z < 0 \end{cases}$. Zeigen Sie, dass G mit dieser Multiplikation ein $\mathbb{Z}$-Modul ist.

2.) Sei nun G zusätzlich endlich erzeugt. Zeigen Sie: Es gibt ein $n \in \mathbb{N}$ und einen Teilmodul $M \leq \mathbb{Z}^n$, so dass $G \cong \mathbb{Z}^n / M$ als $\mathbb{Z}$-Moduln gilt. *Hinweis: Da G endlich erzeugt ist, gibt es einen Homomorphismus $\phi : \mathbb{Z}^n \rightarrow G$ für ein $n \in \mathbb{N}$. Verwenden Sie den Homomorphiesatz.*

Indem man die Erzeuger des Moduls $M \leq \mathbb{Z}^n$ aus Übung [03] als Spalten nebeneinander schreibt, erhält man eine Matrix $A \in \mathbb{Z}^{n \times m}$ mit $M = A\mathbb{Z}^m$. Damit ist $G \cong \mathbb{Z}^n / A\mathbb{Z}^m$. Auch für zu A äquivalente Matrizen B gilt $G \cong \mathbb{Z}^n / B\mathbb{Z}^m$. Insbesondere können wir als B die Smith-Form $\mathrm{diag}(\alpha_1, \ldots, \alpha_n)$ von A nehmen. Sei nun l maximal mit $\alpha_l \neq 0$. Dann ist $\mathbb{Z} / \alpha_1 \mathbb{Z} \times \ldots \times \mathbb{Z} / \alpha_n \mathbb{Z} \rightarrow \mathbb{Z}^n / B\mathbb{Z}^m$, $(x_1 + \alpha_1 \mathbb{Z}, \ldots, x_l + \alpha_l \mathbb{Z}, \alpha_{l+1}, \ldots, \alpha_n) \mapsto (x_1, \ldots, x_n)^{tr} + B\mathbb{Z}^m$ ein Isomorphismus. Damit haben wir den folgenden Satz gezeigt:

Hauptsatz über endlich erzeugte abelsche Gruppen: Sei G eine endlich erzeugte abelsche Gruppe. Dann gibt es ein $k \in \mathbb{N}_0$ und $\alpha_1, \ldots, \alpha_k \in \mathbb{Z}$ mit α_i teilt α_j für alle $1 \leq i \leq j \leq k$, so dass $G \cong \mathbb{Z} / \alpha_1 \mathbb{Z} \times \ldots \times \mathbb{Z} / \alpha_k \mathbb{Z}$.

In der folgenden Übung könnte der Befehl `divisors` aus dem Paket `numtheory` nützlich sein. Dieser gibt die Menge der Teiler einer gegebenen Zahl zurück.

```
> with(numtheory);
> ?divisors
```

ÜBUNG [04]:

1.) Verwenden Sie den Hauptsatz über endlich erzeugte abelsche Gruppen, um bis auf Isomorphie alle abelschen Gruppen mit 5, 6, 8, 12 oder 15 Elementen zu bestimmen.
2.) Schreiben Sie ein Programm, das zu gegebenem $n \in \mathbb{N}$ eine Liste aller möglichen Isomorphieklassen abelscher Gruppen mit n Elementen ausgibt. Sie dürfen hierbei eine Gruppe wie im Hauptsatz als Liste $[\alpha_1, \ldots, \alpha_k]$ ausgeben.
3.) Bestimmen Sie alle endlich erzeugten abelschen Gruppen von Ordnung 900.