Zeitschrift: Elemente der Mathematik
Herausgeber: Schweizerische Mathematische Gesellschaft
Band: 36 (1981)
Heft: 5

Artikel: Berlekamp-Algorithmus und programmierbarer Taschenrechner
Autor: Baptist, P.
DOI: https://doi.org/10.5169/seals-35552

$\tan\alpha = \tan\beta \tan\gamma$, in accordance with his result. The second inequality of (5.2) could be new, together with the fact that the path of M is a straight line.

O. Bottema and J.T. Groenman, Delft

REFERENCES

1 R.M. Sutton: Demonstration experiments in physics, p.26 (1938). "A double cone will appear to roll up an incline made of two gradually separating rails if the slope is such that the center of gravity of the body is slowly lowered as it rolls."
2 J. Violle: Cours de physique, tome I, p.164 (1883). «... le double cône homogène que l'on place sur deux règles inclinées réunies à leur sommet par une charnière permettant de les écarter plus ou moins; si l'angle de ces deux lames a une valeur convenable, le double cône roule vers la partie la plus haute des lames inclinées et paraît remonter, tandis qu'en réalité son centre de gravité ne fait que descendre.»
3 F. Auerbach and W. Hort: Handbuch der Physikalischen und Technischen Mechanik, Band II. Technische und Physikalische Mechanik starrer Systeme, p.527-528 (1927).
4 Müller-Pouillet: Lehrbuch der Physik, 1.Band. 1.Teil: Mechanik punktförmiger Massen und starrer Körper, p.632 (1929). «Wir erwähnen alle diese Apparätchen, weil sie nun einmal zum Inventar der physikalischen Kabinette gehören und deshalb dem Physiker bekannt sein müssen.»
5 H. Fleury: Condition d'équilibre du double cône sur deux droites concourantes et également inclinées sur le plan horizontal. Nouvelles Annales de Mathématiques, tome 13, p.211-219 (1854).

Elementarmathematik und Didaktik

Berlekamp-Algorithmus und programmierbarer Taschenrechner

Zahlentheoretische Probleme werden am Gymnasium fast ausschliesslich in der Orientierungsstufe behandelt. Zu diesem Themenbereich gehört auch die Bestimmung des grössten gemeinsamen Teilers (ggT) zweier natürlicher Zahlen, die im Schulunterricht meist mit Hilfe der Primzahlzerlegung erfolgt. Die Einbeziehung von programmierbaren Taschenrechnern in den Unterricht ermöglicht es, derartige Problemstellungen in höheren Klassenstufen nochmals anzusprechen. Hierbei kommt der wohlbekannte euklidische Algorithmus, der in der Orientierungsstufe m.E. zu wenig beachtet wird, wieder voll zur Geltung. Da sich zudem der ggT zweier natürlicher Zahlen a und b als Linearkombination von a und b mit ganzzahligen Koeffizienten darstellen lässt, möchte man diese Koeffizienten ebenfalls mit dem Rechner ermitteln.

Eine Möglichkeit besteht darin, die Zwischenergebnisse, die man im Verlauf der Rechnung erhält, in die letzte Gleichung des euklidischen Algorithmus sukzessive einzusetzen und auf diese Art rückwärts die gewünschte Linearkombination zu berechnen. Dazu müssen aber vorher alle Zwischenergebnisse gespeichert werden, und dies kann, insbesondere wenn a und b ziemlich gross sind, erheblichen Speicherbedarf erfordern.

Man ist folglich an einem Algorithmus interessiert, der mit weniger Speicherplatz auskommt. Um dies zu erreichen, muss die Speicherung der Zwischenergebnisse

vermieden werden. Der in dieser Note betrachtete Algorithmus nach Berlekamp besitzt diese Eigenschaft.

Der Berlekamp-Algorithmus

Der ggT wird analog zum euklidischen Algorithmus berechnet. Zusätzlich werden zwei Folgen erzeugt, die die gesuchten ganzzahligen Koeffizienten der Linearkombination liefern.

Algorithmus. *Seien* $a,b\in\mathbf{N}$. *Setze* $r_0:=a$, $r_1:=b$ *sowie* $p_0:=0$, $p_1:=1$, $q_0:=1$, $q_1:=0$. *Berechne:*

$$r_k=a_{k+1}r_{k+1}+r_{k+2},\qquad 0\leqq r_{k+2}<r_{k+1},\qquad \textit{wobei}\quad a_{k+1}:=\left[\frac{r_k}{r_{k+1}}\right],$$

$$p_{k+2}=a_{k+1}p_{k+1}+p_k,$$
$$q_{k+2}=a_{k+1}q_{k+1}+q_k.$$

Das Verfahren wird abgebrochen, sobald $r_k=0$.

Die Eigenschaften der so erzeugten Folgen $\{r_k\}$, $\{p_k\}$ und $\{q_k\}$ sind im folgenden Satz zusammengefasst.

Satz. *Es gibt ein* $n\geqq 0$ *mit* $r_n\neq 0$ *und* $r_{n+1}=0$. *Für dieses n gilt:*
(*i*) $r_n=ggT(a,b)$,
(*ii*) $bp_n-aq_n=(-1)^{n+1}r_n$.

Bevor dieser Satz bewiesen wird, soll der Ablauf des Algorithmus an einem Beispiel demonstriert werden. Der Übersichtlichkeit halber legt man eine Tabelle an. Es sei $a=232$ und $b=184$. Die ersten beiden Zeilen der Tabelle sind bis auf $a_1=[r_0/r_1]$ vorgegeben.

k	r_k	a_k	p_k	q_k
0	232	-	0	1
1	184	1	1	0
2	48	3	1	1
3	40	1	4	3
4	8	5	5	4
5	0			

Die Aussagen des Satzes sind erfüllt:
$n=4$
$r_n=r_4=ggT(232,184)=8$
$r_{n+1}=r_5=0$
$p_nb-q_na=-8=(-1)^{n+1}r_n$

Die Zeile $k+1$ der Tabelle erhält man aus den Zeilen $k-1$ und k folgendermassen (vgl. Algorithmus):
r_{k+1}: Subtraktion des a_k-fachen der k-ten Zeile von der $(k-1)$-ten Zeile,
p_{k+1},q_{k+1}: Addition des a_k-fachen der k-ten Zeile zur $(k-1)$-ten Zeile,
$a_{k+1}=\left[\frac{r_k}{r_{k+1}}\right]$.

Man benötigt also nur die Daten der beiden Vorgängerzeilen, um die Daten einer neuen Zeile zu berechnen. Das heisst für jeden Schritt braucht man nur sieben vorher berechnete Werte: ein a und je zwei r,p,q. Alle anderen Zwischenergebnisse können vergessen werden. Bei diesem Verfahren werden daher nur wenige Speicherplätze belegt.

Der Beweis des obigen Satzes wird nun nachgeholt: Die Existenz eines n ist gesichert, da die Folge $\{r_k\}$ monoton abnimmt und nach unten beschränkt ist. Die Eigenschaft von r_n, ggT zu sein, ist vom euklidischen Algorithmus her bekannt. Es bleibt noch die Aussage (ii) nachzuweisen.

Zu zeigen: Für $n \geqq 0$ gilt: $bp_n - aq_n = (-1)^{n+1} r_n$ bzw. da $r_0 = a$ und $r_1 = b$:

$$r_1 p_n - r_0 q_n = (-1)^{n+1} r_n . \tag{1}$$

Der Nachweis von (1) erfolgt induktiv. Die Gleichung (1) ist richtig für $n=0$ und $n=1$:

$$r_1 \cdot 0 - r_0 \cdot 1 = (-1)^1 r_0 ,$$
$$r_1 \cdot 1 - r_0 \cdot 0 = (-1)^2 r_1 .$$

Wenn (1) für $n-1$ und n richtig ist, so folgt:

$$\begin{aligned} r_1 p_{n+1} - r_0 q_{n+1} &= r_1 (a_n p_n + p_{n-1}) - r_0 (a_n q_n + q_{n-1}) \\ &= a_n (r_1 p_n - r_0 q_n) + r_1 p_{n-1} - r_0 q_{n-1} \\ &= a_n (-1)^{n+1} r_n + (-1)^n r_{n-1} = (-1)^{n+1} (a_n r_n - r_{n-1}) \\ &= (-1)^{n+1} (-r_{n+1}) = (-1)^{n+2} r_{n+1} . \end{aligned}$$

Der Berlekamp-Algorithmus im Ring der ganzen Zahlen

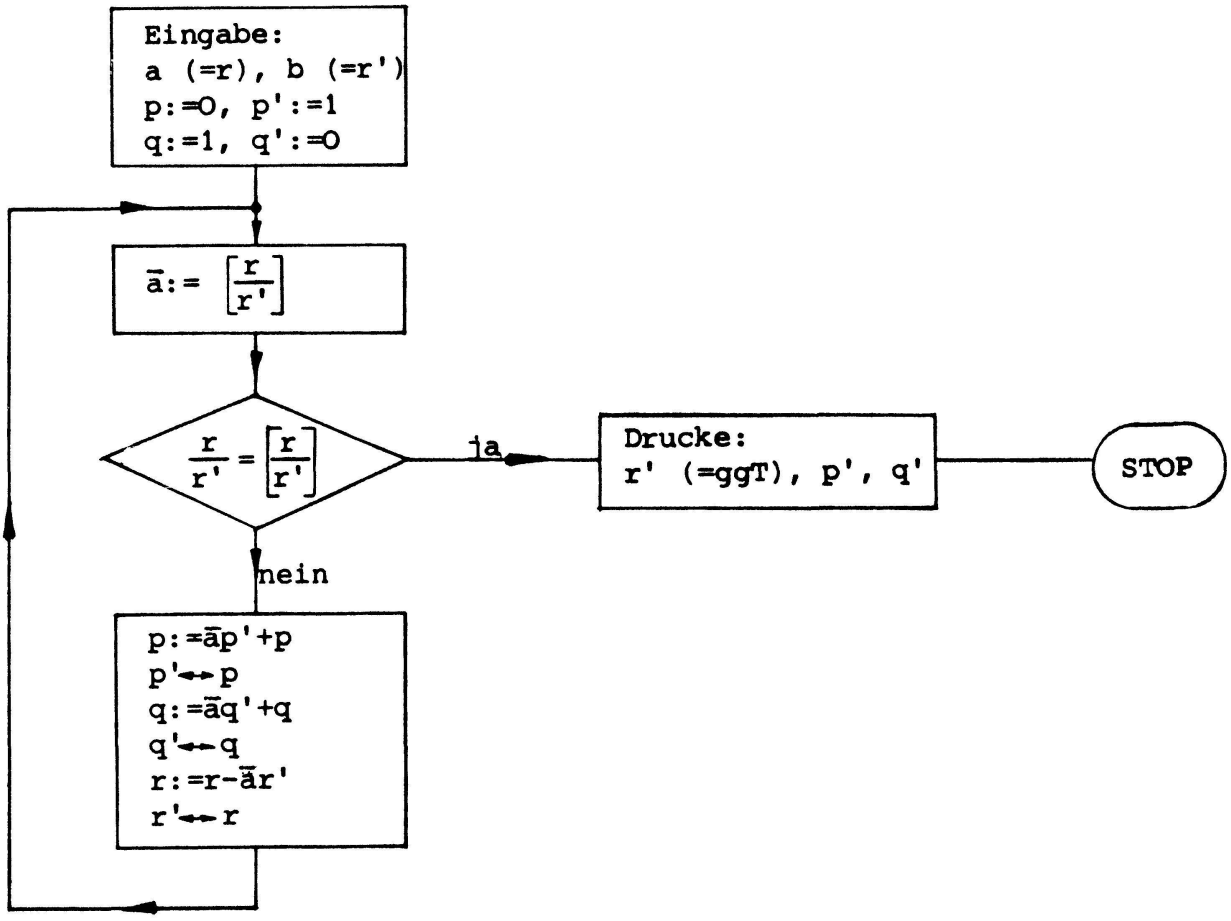
Ein Nachteil der in der Schule üblichen Definition des ggT ist es, dass sie nicht auf allgemeine Ringe erweitert werden kann, da die meisten Ringe nicht die Anordnungseigenschaft besitzen. Daher definiert man (siehe [4]):

Definition. *R sei ein Ring und $a,b,c \in R$. c heisst ein ggT von a und b, falls:*
(i) $c \mid a$ und $c \mid b$,
(ii) ist d gemeinsamer Teiler von a und b, so teilt d auch c.

Die Menge der grössten gemeinsamen Teiler von a und b wird mit $\mathrm{ggT}(a,b)$ bezeichnet.
Im Ring der ganzen Zahlen $(\mathbf{Z}, +, \cdot)$ gilt dann, falls c die obige Definition erfüllt: $\mathrm{ggT}(a,b) = \{-c, c\}$.
Der Berlekamp-Algorithmus behält auch für $a,b \in \mathbf{Z}$ seine Gültigkeit, falls $b \neq 0$ ist. Lediglich die Aussage (i) des Satzes muss abgeändert werden. Sie heisst jetzt: $r_n \in \mathrm{ggT}(a,b)$. Das zu r_n assoziierte Element erhält man durch Multiplikation von r_n mit (-1).

Die Programmierung des Berlekamp-Algorithmus

Zuerst soll der Programmablaufplan angegeben werden, anhand dessen die Programmierung erfolgen kann. Dadurch ist das nachfolgende Programm für den Taschenrechner TI 58 bzw. 59 sofort verständlich[1]). Der Ablaufplan kann ebenso als Orientierung dienen, wenn das Programm für einen anderen Rechnertyp abgeändert bzw. für einen Tischrechner in *Basic* geschrieben werden soll.
Da das Verfahren nur die beiden Vorgänger zur Berechnung der jeweiligen neuen Grösse benötigt, wird nach Abschluss jeder Rechnung der Vorvorgänger überschrieben. Gestrichene Grössen bezeichnen den Nachfolger der entsprechenden ungestrichenen Grössen.



Speicherplätze werden für folgende Parameter reserviert:

Register	R_{00}	R_{01}	R_{02}	R_{03}	R_{04}	R_{05}	R_{06}	R_{07}
Registerinhalt	r	r'	$\bar{a}$	p	p'	q	q'	Zwischenspeicher

1) Steht kein Drucker zur Verfügung, so sind die *Print*-Anweisungen wegzulassen bzw. durch *Stop*-Anweisungen zu ersetzen.

```
000  76 LBL
001  11  A
002  42 STO
003  00  00
004  99 PRT
005  91 R/S
006  76 LBL
007  12  B
008  42 STO
009  01  01
010  99 PRT
011  98 ADV
012  00  0
013  42 STO
014  03  03
015  42 STO
016  06  06
017  01  1
018  42 STO
019  04  04
020  42 STO
021  05  05
022  76 LBL
023  13  C
024  43 RCL
025  00  00
026  55  ÷
027  43 RCL
028  01  .01
029  95  =
030  42 STO
031  07  07
```

```
032  59 INT
033  42 STO
034  02  02
035  32 X:T
036  43 RCL
037  07  07
038  67  EQ
039  14  D
040  43 RCL
041  02  02
042  65  ×
043  43 RCL
044  04  04
045  85  +
046  43 RCL
047  03  03
048  95  =
049  48 EXC
050  04  04
051  42 STO
052  03  03
053  43 RCL
054  02  02
055  65  ×
056  43 RCL
057  06  06
058  85  +
059  43 RCL
060  05  05
061  95  =
062  48 EXC
063  06  06
064  42 STO
065  05  05
```

```
066  43 RCL
067  00  00
068  75  -
069  43 RCL
070  01  01
071  65  ×
072  43 RCL
073  02  02
074  95  =
075  48 EXC
076  01  01
077  42 STO
078  00  00
079  13  C
080  76 LBL
081  14  D
082  43 RCL
083  01  01
084  99 PRT
085  98 ADV
086  43 RCL
087  04  04
088  99 PRT
089  43 RCL
090  06  06
091  99 PRT
092  91 R/S
```

Benutzeranleitung:

Eingabe	Taste	Bemerkung
a	A	
b	B	Ablauf des Programms. Ausgabe von: ggT, p', q'

Beispiele:

```
210.          A
 78.          B

  6.        GGT

  8.         P'
  3.         Q'
```

```
873.          A
449.          B

  1.        GGT

 35.         P'
 18.         Q'
```

```
4964.         A
3796.         B

 292.       GGT

   4.        P'
   3.        Q'
```

Bei diesen Beispielen wurde der Ausdruck des Rechners mit der Druckanweisung *Op* 06 noch zusätzlich markiert.

VERDANKUNG

Für wertvolle Hinweise möchte ich Herrn Prof. Dr. M. Jeger herzlich danken.

P. Baptist, Bayreuth

LITERATURVERZEICHNIS

1 E.L. Berlekamp: Algebraic Coding Theory. McGraw-Hill, New York 1968.
2 A. Engel: Elementarmathematik vom algorithmischen Standpunkt. Klett, Stuttgart 1977.
3 M. Jeger: Zur Behandlung des euklidischen Algorithmus bei Polynomen mit einem programmierbaren Taschenrechner. El. Math. *35*, 25-42 (1980).
4 H. Lüneburg: Vorlesungen über Zahlentheorie. Birkhäuser, Basel 1979.

Kann man ohne Rechner entscheiden, ob e^π oder π^e grösser ist?

In [1] findet sich dazu die folgende elegante Lösung: In der Ungleichung $e^q > q+1$ für $q \neq 0$ setzt man $q=(\pi/e)-1$ und erhält

$$e^{\frac{\pi}{e}-1} > \frac{\pi}{e}.$$

Daraus folgt $e^{\pi/e} > \pi$ und daraus

$$e^\pi > \pi^e. \tag{1}$$

Geht man diese Herleitung nochmals durch, so erkennt man, dass dabei die Zahl π insofern keine wesentliche Rolle spielt, als alle Ungleichungen gültig bleiben, wenn man an Stelle von π irgendeine positive Zahl $\neq e$ einsetzt.

Man schliesst daraus:
Genau die Zahl $a=e$ hat die Eigenschaft

$$\bigwedge_{x>0} a^x \geq x^a. \tag{2}$$

Dass *nur* die Zahl e diese Eigenschaft hat, folgt aus der Tatsache, dass die Gerade $y=x+1$ nur für $a=e$ Tangente an die Kurve $y=a^x$ ist; für jede Basis $a \neq e$ gibt es demnach ein q mit $|q|<1$, so dass $a^q < q+1$. Für $x=a(q+1)$ gilt dann

$$a^{\frac{x}{a}-1} < \left(\frac{x}{a}-1\right)+1 = \frac{x}{a}$$

und daraus wie in (1): $a^x < x^a$.
Damit ist (2) bewiesen.
Einen anderen Zugang zum Satz (2) erhält man, wenn man die Ungleichung $a^x \geq x^a$ logarithmiert. Man erhält dann die äquivalente Ungleichung $x \cdot \ln a \geq a \cdot \ln x$ und daraus