Schöpferisches Denken in der Mathematik

Vortrag anlässlich der Hessischen Landestagung des deutschen Vereins zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts (MNU) am 15. September 2005 in Fulda

Timm Grams, Fachhochschule Fulda, www.fh-fulda.de/~grams

An den Anfang stelle ich eine Erkenntnis der Verhaltensforschung, die der Biologe und Erziehungswissenschaftler Felix von Cube immer wieder betont: Lust ohne Anstrengung ist ein Langweilefaktor. Die verdiente Belohnung von Anstrengung empfinden wir dauerhafter und intensiver als jede geschenkte. Das liegt daran, dass der Mensch von der Evolution für eine harte Wirklichkeit – sozusagen für den Ernstfall – programmiert ist und nicht für das Schlaraffenland.

Viele unserer Studienanfänger kennen diese Lust an Leistung leider nicht[1]. Sie haben nicht erfahren, dass das durch eigene Anstrengungen bewältigte Problem tiefe Befriedigung geben kann.

Mir geht es in diesem Beitrag um Freude an der Mathematik durch kreatives Denken. Die Vermittlung von Freude am Selbstvollbrachten muss in der Schule beginnen. Zwei Grundgedanken stehen im Zentrum meiner Überlegungen:

1. Ohne Basisfertigkeiten und etwas Routine im Umgang mit Zahlen und Formeln geht es nicht.
2. Schöpferisches Arbeiten mit Mathematik lässt sich lernen.

In einer Schlussbemerkung will ich auf die Praxisrelevanz des hier verfolgten Lehrkonzepts eingehen.

Lassen Sie mich zunächst aus dem vergangenen Semester berichten. Zwei Erlebnisse haben mir erneut Defizite in der Mathematikausbildung unserer Studienanfänger vor Augen geführt. Das erste Erlebnis betrifft die Basisfertigkeiten und das zweite die damit zusammenhängenden Schwierigkeiten, Kreativitätstechniken überhaupt zu verstehen.

Zwei ernüchternde Erlebnisse

Wir sind in der Klausur zum Fach „Einführung in die Informatik“. Die Teilnehmer haben regulär ihr zweites Semester hinter sich. In das Aufgabenblatt habe ich eine einfache Aufgabe aus dem Bereich der Computerarithmetik aufgenommen: „Dividieren Sie die Zahl $(1111.011)_2$ durch $(1.1)_2$. Bleiben Sie im Stellenwertsystem zur Basis zwei. Machen Sie die Probe – ebenfalls im Zweiersystem.“

Nachdem die Aufgabenblätter ausgeteilt sind, fragt ein Teilnehmer: „Herr Grams, sagen sie mal. Was ist das eigentlich – eine Probe?“ Von den 90 Teilnehmern liefern 30 die korrekte Lösung; 19 liefern nur Teilergebnisse; und 41 der Teilnehmer können mit der Aufgabe gar nichts anfangen.

Nun zum zweiten Erlebnis. Wir sind in der Lehrveranstaltung „Programmieren“ des vierten Semesters. Es geht um eine Kreativitätstechnik. Unter anderem kann sie bei der Optimierung zeitkritischer Algorithmen helfen.

Die Grundidee: Tritt in einer Schleife ein komplizierter Ausdruck auf, dann lassen sich oftmals Berechnungen einsparen, indem man für den Ausdruck eine Variable einführt und von Schleifendurchlauf zu Schleifendurchlauf nur deren Änderung berechnet. Die Definition der

---

[1] „Lust an Leistung“ heißt ein Buch von Felix von Cube (Piper, München 1998). Ich beziehe mich gern auf sein Buch „Gefährliche Sicherheit. Die Verhaltensbiologie des Risikos“, Hirzel, Leipzig 1995 (Kernsätze: S. 81).

Variablen ist eine *Invariante*, ein Ausdruck also, der bei jedem Schleifendurchlauf an den relevanten Stellen gelten muss.

Zwei Studenten legen mir gegen Ende des Semesters ihre Version eines C-Programms vor, das Linien in das Raster des Bildschirms zeichnet:

```
x=y=0;
while (x<=a) {
  draw(x, y);
  x++;
  if (2*y*a-2*x*b+a<0) y++;
}
```

Ich mache die Bemerkung, dass man für den länglichen Ausdruck in der If-Bedingung (hier fett hervorgehoben) doch eine Variable einführen könne und erinnere an den Trick mit der Invarianten. Die beiden schauen mich verständnislos an. Ich setze mich zu ihnen und wir arbeiten uns am Programmtext Schritt für Schritt zu einem wesentlich zeiteffizienteren Algorithmus vor. Wen es interessiert – hier ist er:

```
x=y=0;
c=2*b;  d=2*a-c;  e=a-c;
while (x<=a) {
  draw(x, y);
  x++;
  if (e<0) {y++; e+=d;} else e-=c;
}
```

Nun fallen sämtliche Multiplikationen der If-Bedingung weg und eine der beiden Additionen. Da die Berechnungen bei jedem Punkt der Geraden anfallen, ist die Effizienzsteigerung enorm: Beim Bildschirm-Scrollen müssen alle elementaren geometrischen Figuren wieder und wieder gezeichnet werden. Die Qualität der graphischen Algorithmen ist augenfällig: entweder es ruckelt oder es ruckelt nicht.

Als das Ziel erreicht ist, sind die beiden begeistert. Sie bringen zum Ausdruck, dass ich solche Dinge künftig nicht mathematisch „umschreiben“ möge. Besser hätte ihnen gefallen, „wenn alles in C erklärt werden würde“.

Offenbar haben die beiden den grundlegenden Trick mit der Invarianten seinerzeit nicht verstanden. Das Aha-Erlebnis kommt erst, als ich die Sache am Programm erkläre. Das finde ich bedauerlich, denn in der Sprache der Mathematik wird eine Einfachheit, Schönheit und Allgemeingültigkeit der Darstellung erreicht, die der Programmtext nicht leisten kann[2].

Von der Freude des Findens

Die üblichen Aufgaben lassen sich mit naheliegenden Methoden und nach bekannten Regeln lösen. Das macht vielleicht ein wenig Spaß; aber echte Freude kommt dabei nicht auf. Anders ist das bei Problemen: Es gibt Widerstände auf dem Weg zur Lösung. Feste Regeln fehlen und man muss auf Entdeckungsreise gehen. Entsprechend hoch ist der Lustgewinn im Erfolgsfalle.

Auch das Explorieren lässt sich trainieren. Es gibt Vorgehensweisen, die dem Denken auf die Sprünge helfen, die *Heuristiken.* Schon das Wort drückt die Freude am Problemlösen aus: Heureka! – ich hab’s gefunden – soll Archimedes bei der Entdeckung der Gesetze des Auftriebs freudig ausgerufen haben. Eine Heuristik ist also ein *Lösungsfindeverfahren*.

---

[2] Der Kern der Verbesserung steckt in der Invarianten „e=2*y*a-2*x*b+a“. Sie muss bei der If-Abfrage gültig sein. Alles Wesentliche über Invarianten beim Algorithmenentwurf bietet das Buch „The Science of Programming“ von David Gries (Springer, New York 1981). Die Anwendung auf Grafik hat Niklaus Wirth im Beitrag „Drawing Lines, Circles, and Ellipses in a Raster“ zum Buch „Beauty is Our Business“ dargestellt (Hrsg.: Feijen, van Gasteren, Gries, Misra; Springer, New York 1990).

Die Freude am Problemlösen sollten schon die Schulkinder erfahren, ebenso die Freude an der Schönheit der Mathematik. Im nächsten Abschnitt stelle ich die wichtigsten Heuristiken für den Schulgebrauch zusammen.

Was ich hier erzähle, ist im Grunde die Methode von Georg Pólya. Er hat sie in seinem herrlichen Bändchen „Schule des Denkens" beschrieben und vorgeführt[3]. Pólya unterweist Lehrer in einer Fragetechnik, mit der sie Schülern zu neuen Erkenntnissen verhelfen können. Pólyas Methode wurde von den Informatikern aufgegriffen und sie genießt besonders auf dem Feld des Algorithmenentwurfs große Wertschätzung[4]. Wir sollten sie an die Schule zurückbringen.

Grundlegende Heuristiken

Neues finden heißt, die richtigen Fragen stellen. Für die grundlegenden Heuristiken führe ich eine Kurzbezeichnung ein und gebe die Fragen an, die man sich beim Problemlösen vorlegen soll.

Analogie: Habe ich etwas Ähnliches schon einmal gesehen? Kenne ich ein verwandtes Problem?

Generalisierung: Bringt mich der Übergang von einem Objekt zu einer ganzen Klasse von Objekten weiter?

Spezialisierung: Komme ich weiter, wenn ich zunächst einmal einen leicht zugänglichen Spezialfall löse?

Variation: Kann ich durch die Veränderung der Problemstellung der Lösung näher kommen? Kann man das Problem anders ausdrücken?

Rückwärtssuche: Hilft es, wenn ich beim gewünschten Resultat anfange? Welche Operationen können mich zu diesem Ergebnis führen?

Teile und herrsche: Lässt sich das Problem in leichter lösbare Teilprobleme zerlegen?

Vollständige Enumeration: Kann ich mir Lösungen verschaffen, die wenigstens einen Teil der Zielbedingungen erfüllen? Kann ich mir sämtliche Lösungen verschaffen, die diese Bedingungen erfüllen?

Diese Heuristiken bewegen sich im Rahmen der „Schule des Denkens". Auf der Web-Seite www.fh-fulda.de/~grams/Heuristik/Heuristik.htm führe ich das Arbeiten mit diesen Heuristiken an ausgewählten schulgemäßen Problemen vor.

Für diesen Beitrag will ich noch eine „höhere" Heuristik hinzunehmen.

Invariante: Lässt sich das Problem in mehreren Stufen oder durch Verzweigungen lösen? Lassen sich Aussagen oder Gleichungen finden, die auf jeder Stufe (in jedem Zweig) der Problemstruktur gültig sind und die zur Problemlösung beitragen? Wie lässt sich auf einfache Weise die Gültigkeit der Aussagen herstellen und aufrechterhalten? Lässt sich die Berechnungseffizienz steigern?

An einem Preisrätsel wollen wir die Anwendung der Heuristiken einmal ausprobieren.

Binäres Spielchen

Das Preisrätsel aus dem Oktoberheft 2004 von Spektrum der Wissenschaft lautet so: Zwei Informatiker, Marc und Ben, spielen. Vorab haben sie eine Liste von Primzahlen vereinbart, beispielsweise (5, 7).

---

[3] Original: How to Solve It. Princeton, N. J. 1945

[4] Michalewicz, Z.; Fogel, D. B.: How to Solve It: Modern Heuristics. Springer, Berlin, Heidelberg 2000

Marc schreibt nun eine Folge von binären Ziffern auf, indem er mit einer 1 beginnt und ständig weitere Ziffern 0 oder 1 anfügt. Dabei entstehen der Reihe nach Binärzahlen, zum Beispiel 1 (1), 10 (2), 100 (4), 1001 (9), 10010 (18), 100100 (36), ... In Klammern stehen die Dezimaldarstellungen der Zahlen.

Ben darf Marc jederzeit unterbrechen und selbst eine der Ziffern 0 oder 1 anhängen. Ist die so entstandene Zahl durch eine in der Liste vorkommende Primzahl teilbar, so hat Ben gewonnen. Wenn Ben dies nie schafft, dann hat Marc gewonnen. Enthält die Liste der Primzahlen die Zahl 73, dann gewinnt im obigen Beispiel Ben. Er braucht an die 100100 nur eine 1 anzuhängen und es ergibt sich die Zahl 1001001 (73).

Enthält die Primzahlliste eine 2 oder 3, dann gewinnt immer Ben: Nach der ersten 1 hängt er sofort eine weitere 1 an und erreicht 11 (3); durch Anhängen einer 0 an beliebiger Stelle erreicht er eine durch 2 teilbare Zahl – wenn sie nicht sowieso schon durch zwei teilbar war.

Besteht die Liste der Primzahlen ausschließlich aus der 7, so gewinnt Marc, wenn er nach der ersten 1 nur noch Nullen anfügt. Ähnliches gilt für die Primzahl 5. Hier kann Marc nach der ersten 1 immer abwechselnd 1 und 0 anhängen. Gesucht ist das in der Summe kleinste Pärchen Primzahlen, das nicht die Zahlen 2 und 3 enthält, bei dem a) Ben gewinnt, b) Marc gewinnt.

Ein Weg zur Lösung

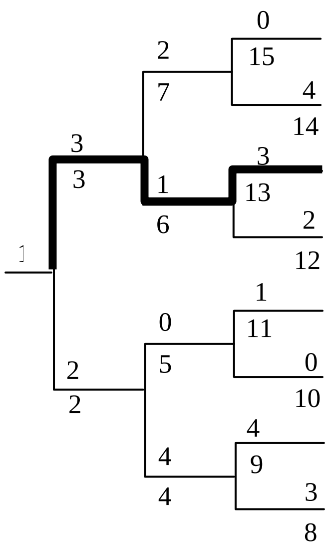
Die Aufgabe sieht schwierig aus. Wie anfangen?

Wir machen uns zunächst die Struktur des Problems klar: Offenbar muss man sich von Stufe zu Stufe hangeln. Mit jedem angehängten Zeichen wird eine neue Stufe erreicht. Auf jeder Stufe gibt es Alternativen für das weitere Voranschreiten. Wir suchen Zuflucht zur ersten Heuristik, *Analogie:* Habe ich etwas Ähnliches schon einmal gesehen? Kenne ich ein verwandtes Problem?

Na klar: In der Wahrscheinlichkeitsrechnung gibt es mehrstufige Zufallsversuche. Für die Darstellung dieser Versuche werden Baumdiagramme genutzt. Damit ist bereits ein wesentlicher Schritt getan: Wir haben eine Darstellungstechnik gefunden, das *Baumdiagramm.*

Wir stellen die Binärzahl als binären Baum dar. Die Baumwurzel (Startzweig) bekommt wie jeder Zweig, der nach oben abgeht, eine 1 zugeordnet. Abzweigungen nach unten entsprechen einer 0.

Jetzt ist Zeit für *Spezialisierung*. Wir nehmen das Beispiel mit der einelementigen Primzahlliste (5) und zeichnen den Baum für die ersten vier möglichen Spielzüge von Marc. Unter jedem Zweig steht die zur Binärfolge gehörende Dezimalzahl und darüber steht der Divisionsrest bei ganzzahliger Division durch 5. (Im allgemeinen Fall ist es ein Folge von Divisionsresten.)



Primzahlliste (5)

Wir sehen, an welchen Stellen Ben eingreifen und gewinnen könnte. Aber Marc kann durch geschickte Spielzüge diese Teile des Baumes vermeiden, beispielsweise indem er die Folge 1101010101... erzeugt. Dem entspricht die Folge der Divisionsreste 1, 3, 1, 3, 1, 3, ...

Nun etwas Theorie, also *Verallgemeinerung:* Eigentlich sind wir nur an den Divisionsresten interessiert. Wir suchen nach einer einfachen Regel oder Formel für den Zusammenhang zwischen den Divisionsresten.

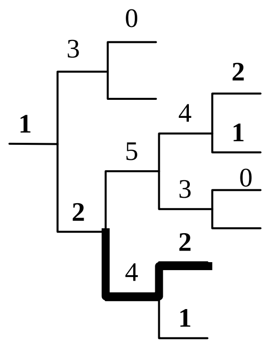
Nach der problemlösenden *Invarianten* muss man hier nicht lange suchen: Die Zahl r oberhalb eines Zweiges ist der Divisionsrest, der sich ergibt, wenn man die unterhalb des Zweiges stehende Zahl z durch p (hier: 5) teilt. Folglich gilt $z = ap + r$ mit $r<p$.

Eine Möglichkeit zur Aufrechterhaltung der Invarianten haben wir schon. Wir brauchen auf jeder Stufe nur die Zahl z durch p zu teilen und erhalten so den gesuchten Rest r. Die Heuristik legt nahe, nach *effizienteren* Verfahren zu suchen.

Die Zahl des nach oben abgehenden Zweiges (Kindes) ergibt sich zu $2z+1$, für das untere Kind erhalten wir $2z$. Die zugehörigen Divisionsreste sind r_1 und r_0. Die Berechnung wollen wir möglichst vereinfachen. Wir können versuchen, die Gültigkeit der Invarianten für den Vaterzweig, nämlich $z = ap + r$ mit $r<p$, auszunutzen.

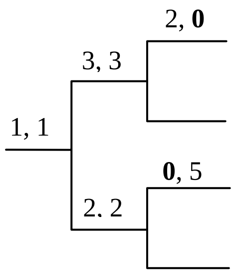
In den Zahlendarstellungen für die Kindzweige ersetzen wir z durch $ap + r$ und erhalten $2ap + 2r+1$ bzw. $2ap + 2r$. Daraus folgt ein einfacher Zusammenhang zwischen den Divisionsresten von Vater- und Kindzweigen: Ist r der Divisionsrest des Vaterzweigs, dann ist der Divisionsrest des oberen Kindes gleich dem Divisionsrest von $2r+1$ geteilt durch p. Für das untere Kind ist $2r$ durch p zu teilen.

Daraus folgt, dass die Zahlen z für die Rechnung nicht mehr benötigt werden und dass nur noch relativ kleine Zahlen in die Berechnungen eingehen. Das vereinfacht die Lösung mit Papier und Bleistift und ist Basis eines leistungsfähigen Computerprogramms ohne Overflow-Risiko. Ab jetzt schreiben wir bloß noch die Divisionsreste an die Zweige des Baums.



Primzahlliste (7)

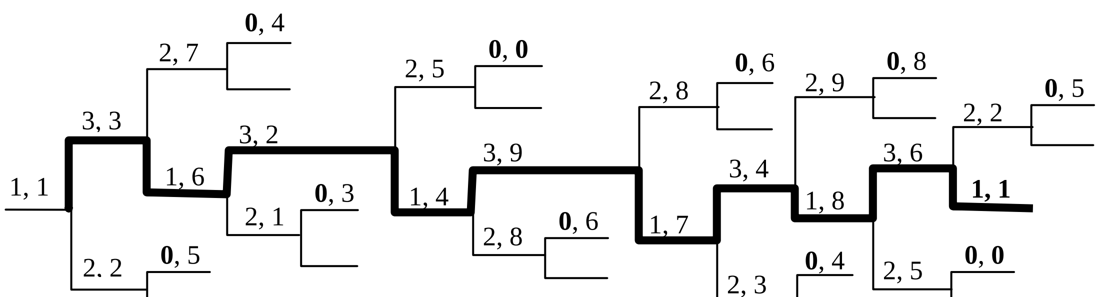
Die Beschränkung der Berechnungen auf die Divisionsreste führt auch auf eine einfache Bedingung dafür, dass Marc gewinnen kann: Wenn Ben bisher keine Möglichkeit des Eingreifens hatte und wenn die momentan erreichte Restefolge auf dem bisher zurückgelegten Pfad bereits einmal vorgekommen ist, sollte Marc die Binärzeichenfolge ab dieser Stelle fortlaufend wiederholen. Dann erhält Ben auch zukünftig keine Chance für erfolgreiches Eingreifen.



Primzahlliste (5, 7)

Im Falle der Primzahlliste (7) gibt es beliebig viele Gewinnmöglichkeiten für Marc, beispielsweise die Folgen 10000... und 1001010101... Nur für den letzen Fall ist im Baumdiagramm die Wiederholungsfolge fett markiert. Außerdem sind Kombinationen möglich, beispielsweise diese: 10010000101011111...

Mit der eben skizzierten Methode lässt sich prüfen, dass beim Primzahlpaar (5, 7) Ben gewinnt. Marc gewinnt beim Primzahlpaar (5, 11). Seine Gewinnfolge ist 11010101010... In Dezimalschreibweise entspricht das der Folge der Zahlen 1, 3, 6, 13, 26, 53, 106, 213, 426, 853, 1706, ... Die beiden Primzahlpaare sind die jeweils kleinstmöglichen bei Ausschluss von 2 und 3. Damit ist das ursprüngliche Problem gelöst.



Primzahlliste (5, 11)

Einige unbotmäßige Bemerkungen zum Schluss

Immer wieder wird von manchen Kollegen und Pädagogen angemahnt, von Anfang an praxisnah zu lehren. Für alles, was man lehrt, soll nach deren Auffassung die Anwendung in Sichtweite sein. Sie meinen, dass nur auf diese Weise praxisrelevantes Wissen vermittelt werden könne. Außerdem könne man so die Lernenden besser bei Laune halten. Sie könnten ja stets unmittelbar sehen, wofür die ganze Plackerei gut ist.

Das halte ich für falsch. Eine strikte Praxisbindung führt zu einer lückenhaften und trivialisierten Lehre. Ich halte dem entgegen, dass man praxisrelevante Techniken und Methoden schon sehr früh einüben kann und muss, selbst wenn die Anwendungen noch in der Ferne liegen.

Praxiserprobung findet normalerweise in einem hoch komplexen Weltausschnitt statt. Ein Nachbau solcher Weltausschnitte für jeden Lehrzweck ist ineffizient.

Ein entsprechender Einwand gilt auch für zu früh angesetzte Projektarbeit. Ich halte sehr viel von Projektarbeit. Meines Erachtens sollte sie gegen Ende des Studiums in exemplarischer Form stattfinden und nicht bereits zu Beginn. Erst dann haben die Studierenden genügend Techniken und Fertigkeiten erworben, die sie befähigen, die Komplexität in vollem Umfang zu bewältigen.

Das spricht nicht gegen das spielerische Erkunden von Sachverhalten mit Miniprojekten. Diese haben typischerweise das Ziel, Interesse für ein Studium oder einen Lehrgegenstand zu wecken. Die Erarbeitung von Lehrstoff ist demgegenüber nachrangig.

Projekte, die ich meine, zeichnen sich durch konstruktive und kreative Elemente aus, sie sind in Teamarbeit zu bewältigen, sie ziehen sich über ein ganzes Semester hin und nehmen einen beträchtlichen Teil der Wochenarbeitszeit der Studierenden in Anspruch. Und das Wichtigste ist, dass sie durch den Lehrenden konsequent kontrolliert werden. Solche Projekte stellen an alle Beteiligten die allerhöchsten Anforderungen und liefern schließlich – im Vergleich zu anderen Lehrformen – die größte Befriedigung. Früchte müssen reifen, bevor sie schmecken.

Die *Lehre-Praxis-Verzögerung* stellt den Lehrer in unserer heutigen Gesellschaft vor eine schwere Aufgabe: Er muss die Praxisrelevanz seiner Lehre glaubwürdig darstellen, ohne sich in Praxisdetails zu verlieren. Er muss Techniken und Methoden lehren, wohl wissend, dass deren Nutzen erst später richtig spürbar wird. Dabei steht er vor einer Jugend, zu deren Erfahrung gehört, dass man Belohnung fast ohne Anstrengung und stets sofort bekommen kann[5].

Dieser Beitrag soll auch deutlich machen, wie der Lehrende die Lücke zwischen Lehre und Praxis überbrücken kann: An einem Beispiel aus dem Gebiet des Algorithmenentwurfs zeigte ich, dass sich das Denken in *Invarianten* in praktischen Problemlagen bezahlt macht. Diese Anwendung liegt noch außerhalb des Erfahrungshorizonts der Schüler. Aber sie können an dem Beispiel erahnen, welchen Nutzen das Erlernte „im Ernstfall“ hat.

Eingeübt werden die praxisrelevanten Techniken aber an ziemlich künstlichen Problemlagen. Hier habe ich einmal etwas aus dem Bereich der Denksportaufgaben und Rätsel genommen. Solchermaßen reduzierte Problemstellungen sind auf dem Niveau des Schülers versteh- und

---

[5] „Doch auch in bürgerlichen Familien verkommen Kinder – vor Fernsehern und Computern in stundenlang ungebremstem Medienkonsum. ‚Wohlstands- oder Verwöhnungsverwahrlosung’ nennen Psychologen den Zustand, in dem Kinder alles haben bis auf die Zeit und Zuwendung der Eltern“ (DER SPIEGEL, Nr. 29/18.7.05, S. 127).

Neil Postman beschreibt den Trend hin zur oberflächlichen Unterhaltung 1985 in seinem Buch „Amusing Ourselves to Death“ so: „Watchers of Sesame Street ... are learning that it must always be entertaining, that learning is largely a matter of images, and that learning has to involve immediate gratification“.

lösbar. An derartigen Übungen ist nichts Verwerfliches, sie sind nicht L’art pour l’art, und sie folgen dem Gebot der *Weltorientierung der Mathematik.*

Kurz gesagt: Gute Lehre ist praxisorientiert, nicht praxisverhaftet.