Vom Nutzen und vom Nachteil der Informatik für den Mathematikunterricht

Reinhard Oldenburg, Heidelberg

Wenn ein Titel auf Friedrich Nietzsche anspielt, kann man zu Recht erwarten, dass der Autor provozieren will — dies war im vorliegenden Fall ursprünglich auch beabsichtigt. Es zeigte sich aber im Verlauf der Arbeitskreis-Tagung, dass die Positionen zu Fragen der Einbeziehung informatorischer Inhalte in den Mathematikunterricht extrem weit auseinander liegen, so dass auch die hier aufgestellten Thesen recht unterschiedlich bewertet wurden und werden. Die Überlegungen folgenden dicht dem Tagungsthema. Es soll diskutiert werden, welche Inhalte und Methoden der Informatik den Mathematikunterricht bereichern können.

1 Ausgangspunkte

Computer sind in unserer Gesellschaft ein extrem leistungsfähiges Hilfsmittel zur Problemlösung, Problemanalyse und manchmal auch Problemgenerierung geworden. Eine kurze Recherche im Internet und die Beachtung der enormen Dynamik dieser Entwicklung sollte Teil a) der folgenden These bestätigen:

> These: a) Der wechselseitige Einfluss von Computern und Mathematik wird — gerade in der Schule — immer noch unterschätzt. b) Deshalb werden die Möglichkeiten des PCs zur Problemlösung im Schulunterricht nicht adäquat abgebildet.

Teil b) der These ist problematischer. Natürlich müssen Schüler Einblicke erhalten in die technologische Basis unserer Gesellschaft, und im Mathematikunterricht ist es Aufgabe, die mathematischen Anteile daran transparent zu machen. Wie viel es aber dazu bedarf, wie konkret Schüler die Lösung mathematischer Probleme mit dem Computer und die Nutzung in Anwendungen kennen lernen müssen, bedarf der didaktischen Diskussion.

Während die Fachwissenschaft Mathematik mitsamt ihren Vernetzungen ein dynamischer Prozess ist, ist die Schulmathematik eher starr und immer noch stark im 19. Jh. verwurzelt. Dies soll an zwei Fragestellungen der analytischen Geometrie exemplarisch kontrastiert werden:

Traditionelle Fragestellung Welcher Anteil des Volumens eines Tetraeders nimmt die einbeschriebene Kugel ein? Steigerung: Wie hoch steigt der Anteil, wenn auch die vier freien Ecken mit je einem Kügelchen maximaler Größe ausgefüllt werden?

Aktuelle Fragestellung Eine Videokamera nimmt Bilder vorbeifahrender Autos und ihrer Fahrer auf. Daraus soll ein 3D-Modell der Gesichtsoberflächen berechnet werden, aus dem sich biometrische Daten extrahieren lassen. Das grundlegende Verfahren zur Lösung dieser Fragestellung kann im Unterricht vollständig behandelt werden. Die Schüler lernen dabei zweierlei, zum einen, wie man an Fördergelder des Bundesinnenministeriums herankommt, und zum zweiten, wie man gesellschaftlich-kritisch denkt.

2 Mathematik und Informatik — ein schwieriges Verhältnis

Die Mathematik stellt eine der wichtigsten Wurzeln der Informatik dar, für die nicht Hardwaregebundenen Teile sogar die wichtigste. In der rasanten Entwicklung dieses Fachs hat die Informatik aber einen erheblichen Bestand an eigenen Methoden und Inhalten hervorgebracht, der die Eigenständigkeit als universitäres Fach wie als Schulfach eindrucksvoll unterstreicht. Für die Didaktik der Informatik gilt ähnliches. Insbesondere im Zuge der didaktischen Aufarbeitung der Methoden des Designs von Informatiksystemen, wie sie besonders am Vordringen der Objektorientierten Modellierung (OOM) in den Informatikunterricht sichtbar wird, hat sich eine Mathematikferne Grundhaltung etabliert. Dies spiegelt sich wieder in den deutlichen Worten der Abgrenzung zur Mathematik, wie man sie in der Literatur der Informatikdidaktik findet. So argumentieren beispielsweise (Schubert & Schwill, 2004): „..Mathematik [arbeitet] meist mit kontinuierlichen und elementaren Größen... Informatik ... mit strukturierten Objekten...?. Eine solche Unterscheidung ist für die Fachwissenschaft falsch, man denke etwa an reich strukturierte mathematische Begriffe wie Vektorraumbündel oder Tensorkategorien. Sie ist tendenziell richtiger im Bereich der Schulmathematik, aber auch dort muss widersprochen werden, denn beispielsweise sind Funktionsgraphen keineswegs elementar, und eine Reklamation des ganzen Gebietes der diskreten Mathematik für die Informatik kann auch nicht im Sinne des Mathematikunterrichts sein, zumal in einer Zeit, in der dieses Gebiet im Mathematikunterricht zunehmend Unterstützung erfährt. These: Eine deontologische Abgrenzung kann wegen zu enger Verwandtschaft der Fächer nicht gelingen.

Damit soll nicht gesagt werden, dass es nicht Inhalte gäbe, die man eindeutig in einem der Fächer verorten kann, aber die Fachcharakteristika ermöglichen keine klare Trennlinie, und es gibt daher einen riesigen Überschneidungsbereich. Es ist daher sinnvoll, wenn es beide Fächer gibt, und wenn die Aufgabenverteilung nach pragmatischen Kriterien, wie der Verfügbarkeit von Lehrkräften, den organisatorischen Rahmenbedingungen (z.B. Informatik als Wahl(pflicht)fach oder als Pflichtfach) und natürlich nicht zuletzt nach inhaltlichen Bildungszielen erfolgt.

Das angesprochene Überschneidungsgebiet soll nun durch eine unvollständige Aufzählung etwas belichtet werden: Logik, Berechenbarkeit, Komplexität, Korrektheitsbeweise, Zahlentheorie, Kryptographie, Informationstheorie, Datenkompression, diskrete Mathematik, Graphentheorie, Kombinatorik, Optimierung, Numerik, Computeralgebra, Computergrafik, Virtual Reality, Künstliche Intelligenz, Bildverarbeitung, künstliches Sehen, Robotik, Statistik, Monte-Carlo-Simulationen.

Neben dieser großen Grauzone zwischen den Fächern gibt es genuin mathematische Gebiete, beispielsweise die Topologie, und genuin informatorische Gebiete, wie Softwaredesign, OOM, Echtzeitsysteme, Theorie der Betriebssysteme u.s.w.. Die Eigenständigkeit dieser Gebiete untermauert die Stellung der Informatik als eigenes Fach.

Neben inhaltlichen Anregungen kann der Mathematikunterricht auch methodischen Input von der Informatik nutzen. Im Informatikunterricht ist der Projektunterricht eine tragende Form. Das ist möglich, weil der Computer den Schülerinnen weitreichende Handlungsmöglichkeiten eröffnet. Für den Mathematikunterricht gilt analog das gleiche. Es ist bedenkenswert, ob man unter diesem Gesichtspunkt die Inhalte des gegenwärtigen Mathematikunterrichtes überdenken muss, denn Unterricht wird nicht nur durch Inhalte legitimiert, sondern auch durch die Methoden und Prozesse, die in ihm möglich sind.

3 Algorithmen sind gut!

Algorithmisierung ist allgemein als fundamentale Idee der Mathematik anerkannt (siehe z.B. Tietze et al., 1997, 38ff) Bedauerlicherweise verzichten allerdings die KMK-Bildungsstandards für den mittleren Schulabschluss (KMK, 2003) auf eine entsprechende Würdigung des Algorithmenbegriffs. Möglicherweise liegt dies begründet in der Beobachtung, dass im Mathematikunterricht das kalkülhafte Arbeiten einen zu hohen Stellenwert einnimmt. Man beachte aber, dass Algorithmisierung als fundamentale Idee im Mathematikunterricht nicht dazu führen sollte, dass die Schüler wiederholt Algorithmen ausführen. Satt dessen sollen sie Algorithmen entwerfen, bewerten und über ihre Korrektheit und angemessene Nutzung kritisch reflektieren. Für eine sinnvolle Nutzung sollte die Algorithmisierung im Mathematikunterricht wirklich als Prozess erlebbar werden. Algorithmen kodieren und kombinieren Erkenntnisse. Dazu müssen Erfahrungen gesammelt, zu mathematischen Aussagen verdichtet und in ein Verfahren übersetzt werden, das wieder Gegenstand mathematischer Reflexion sein wird. Exemplarisch soll dies am Beispiel des Euklidischen Algorithmus zur Bestimmung des größten gemeinsamen Teilers zweier Zahlen erläutert werden. Die Phasen des Algorithmisierungsprozesses können sein:

1. Erfahrung:
   Zahlenbeispiele werden gesammelt
2. Beobachtungen werden zu Regeln verdichtet: $ggT(n,n) = n$, $ggT(n,1) = 1$, $ggT(n,m) = ggT(m,n)$, $ggT(n,m) = ggT(n,m-n)$
3. Umsetzung in einen Algorithmus:

```
ggT(n,m):=
   if n>m       : ggT(m,n)
   else if n=1  : 1
   else if n=m  : n
   else         : ggT(n,m-n)
```

4. Korrektheitsbeweis

Wenn Schüler einen solchen Prozess durchlaufen können, ist das Zitat „Algorithmen sind gut!? von (Freudenthal, 1972) bestätigt.

4 Mathematik als Zuliefer-Wissenschaft

Schüler kennen in der Regel viele Anwendungen, in denen mathematische Verfahren angewendet werden, beispielsweise Bildverarbeitungsprogramme und Soundeditoren. Eine Strategie des Mathematikunterrichts kann darin bestehen, solche Programme punktuell zu öffnen und die mathematischen Grundlagen den Schülern direkt zugänglich zu machen.

Bildbearbeitungsprogramme bieten viele Möglichkeiten, Helligkeit und Kontrast eines Bildes zu ändern. Dabei wird eine Funktion bestimmt, die die Menge der Helligkeitswerte (also die ganzen Zahlen von 0 bis 255) auf sich selbst abbildet. Im kostenlosen Bildbearbeitungsprogramm Gimp[1] kann diese Funktion (genauer gesagt, ihr Funktionsgraph) explizit angezeigt und sogar mit der Maus verändert werden (siehe Abb. 16.1). Viele Computeralgebraprogramme (MuPAD, Maple, Mathematica) oder ein kleines Programm (siehe Abb. 16.2) des Autors ermöglichen auch, die Transformationsfunktion algebraisch zu

[1] http://www.gimp.org

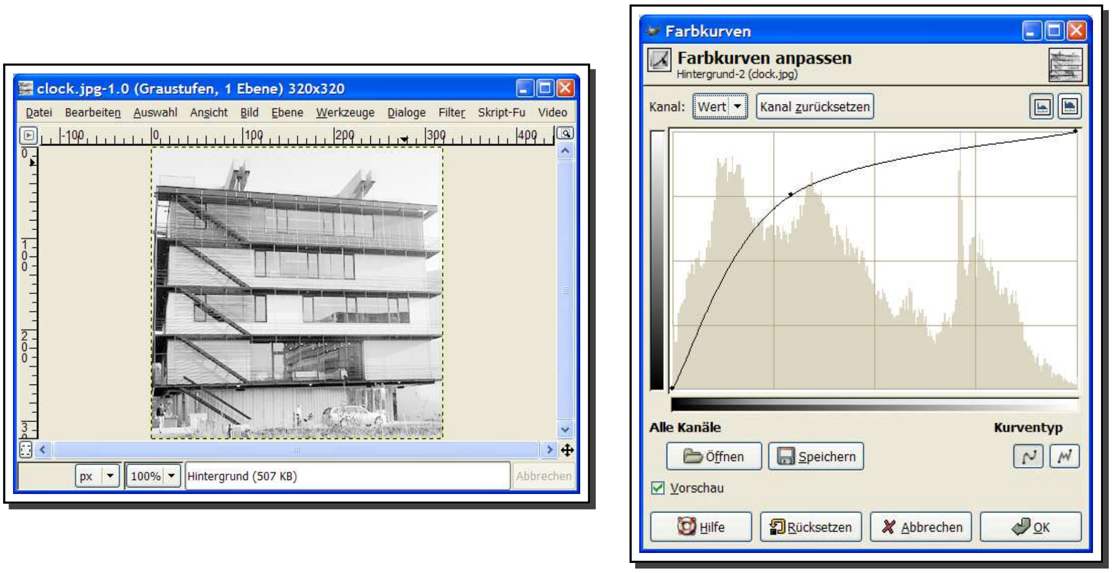

Abbildung 16.1: Einstellung der Helligkeitswerte in GIMP

Abbildung 16.2: Anwenden einer durch einen Term gegebenen Funktion auf die Helligkeitswerte eines Bildes

spezifizieren. Dies ermöglicht eine schöne Darstellung des Simultanaspektes von Funktionen.

In eine ähnliche Richtung geht die Verwendung von Funktionstermen zur Modellierung von Klängen. Mit einem vom Autor erhältlichen Zusatzmodul kann das CAS MuPAD Funktionen als Schalldruckpegel eines zu erzeugenden Tonsignals verstehen. Beispielsweise erzeugt

```
PlayFunction(
   0.5*(sin(2*PI*441*t)
       +sin(2*PI*440*t)),
        t=0..2.0)
```

eine zwei Sekunden lange Schwebung. Schülern machen solche und auch einfachere Beispiele (etwa ein Ton, dessen Frequenz langsam ansteigt) von Schallmodellierung viel Spaß. Algebra wird zum Mittel, es lustig piepsen zu lassen.

Ein weiteres Beispiel bilden die in Mailprogrammen integrierten Spamfilter. Welche Mail als Spam gilt, hängt vom Nutzer ab, es ist eine individuelle Entscheidung. Deswegen sollten Spamfilter lernfähig sein. Zum Lernen aus Erfahrung bietet sich die Bayesstatistik an, die ermöglicht, aus der beobachteten Häufigkeit eines bestimmten Wortes in der bisher als Spam klassifizierten Mail zu errechnen, mit welcher Wahrscheinlichkeit eine einkommende Mail, in der dieses Wort gefunden wurde, Spam ist:

$$P(\text{Spam}\,|\,\text{„Viagra“}) = \frac{P(\text{„Viagra“}\,|\,\text{Spam})}{P(\text{„Viagra“}\,|\,\text{Spam}) + P(\text{„Viagra“}\,|\,\text{Ham})}$$

Natürlich stützt man sich in der Praxis nicht nur auf ein Wort, sondern auf viele, und auch über deren geschickte Auswahl kann man statistisch argumentieren. Eine große Klasse von realen Anwendungen erschließt sich der Mathematikunterricht, wenn er Optimierungsalgorithmen behandelt, da dies in Oldenburg (2006) ausführlich dargestellt werden soll, bleibt es hier bei diesem Hinweis.

4.1 Simulationen

Simulationen können den MU an vielen Stellen bereichern, vor allem bei Versuchen zur Wahrscheinlichkeitstheorie. Dabei können fertige Programme eine wichtige Rolle spielen. Besser ist aber, wenn die Schüler lernen, mit einem Werkzeug beliebige Fragestellungen simulieren zu können. Dies kann z.B. mit einer Tabellenkalkulation erfolgen. Dabei treten allerdings technische Schwierigkeiten (z.B. das Auszählen von Erfolgen in einer Spalte) auf, die den Problemen beim Programmieren u.U. ebenbürtig sein können. Gänzlich an seine Grenzen stößt ein solches Werkzeug, wenn sehr große Versuchszahlen durchzuführen sind. Beispielsweise zeigt Meyer (2006) wie der für Anwendungen extrem wichtige t-Test auf Basis von mit Simulationen gewonnen Erkenntnissen angewendet werden kann, ohne wesentliche Teile der traditionellen Theorie zu benötigen. Dazu sind allerdings große Anzahlen ($> 10^6$) von Versuchsdurchführungen nötig, und das ist mit einer Tabellenkalkulation unmöglich. Natürlich ist es nicht prinzipiell nötig, dass jeder Schüler ein solches Simulationsprogramm selbst schreibt, man sollte aber bedenken, dass eine solche Simulation nur dann Überzeugungskraft hat, wenn die Simulationsprinzipien vollkommen klar sind und im Vorfeld ausreichend vertrauensbildende Maßnahmen durchgeführt wurden. Ein Weg dazu ist das eigenständige Programmieren. Eine Bewertung dieses Weges im Vergleich zu Alternativen steht aus.

Hier soll ein weiteres Themengebiet vorgestellt werden, das durch Simulation behandelt werden kann, und das sich durch den Fächerübergriff von Mathematik, Informatik und Physik auszeichnet. Unter dem Begriff Perkolation fasst man eine Reihe statistischer Modelle zusammen. Das einfachste Perkolations-Modell besteht aus einem zweidimensionalen Quadratgitter, dessen Knoten mit einer bestimmten vorgegebenen Besetzungswahrscheinlichkeit als besetzt (sonst als frei) markiert werden. In einem Realmodell kann man eine Mischung von Glas- und Stahlkugeln herstellen und in eine zweidimensionale Schicht bringen. Eine sinnvolle Frage ist dann, ob eine elektrische Verbindung über die Stahlkugeln vom linken zum rechten Rand der Schicht hergestellt wird. Natürlich ist sofort klar, dass die Wahrscheinlichkeit P einer leitenden Verbindung mit der Besetzungswahrscheinlichkeit p (Stahlkugelanteil) steigt. Das — für viele überraschende — Simulationsergebnis ist, dass P unstetig von p abhängt, für kleine Werte von p ist $P = 0$, bei einem Schwellenwert aber springt P auf 1, d.h. das Modell zeigt einen Phasenübergang. Eine Erkenntnis, die enorme praktische Rückwirkungen hat bei Fragen der Ausbreitung von Waldbränden oder beim Einschluss von Erdgas in porösen Gesteinsschichten. Das Simulationsprogramm (siehe Abb. 16.3) zeigt auch, dass an der Sprungstelle fraktale leitende Gebilde entstehen.

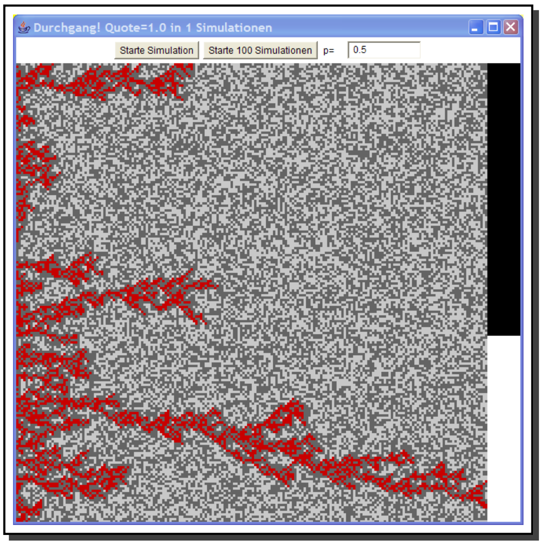

Abbildung 16.3: Fraktalbildung bei Perkolation

4.2 Working Models — das Prinzip verstehen

Für viele gängige und bekannte Programmarten, z.B. Compiler, Interpreter, Browser, Dynamische Geometrie, Computeralgebra, Klassensystem, Chatserver, können im Informatikunterricht „Working models“ also Funktionsmodelle erstellt werden. Diese Programme sollen nicht trivial sein, sondern wirklich funktionieren, dabei aber soweit reduziert sein, dass ihre Arbeitsweise im Detail verstanden werden kann. Ziel ist Transparenz herzustellen und eine Grundlage für Aktivitäten der Schüler zu geben, die dabei Einsicht in reale Probleme der Softwareentwicklung gewinnen können.

Aus dem Sicht des Mathematikunterrichts ist es beispielsweise interessant, wenn im Informatikunterricht ein Dynamisches Geometrieprogramm (DGS) erstellt wird. In diesem Fall bringt die Mathematik die Beschreibungsmethoden für Graden, Strecken und Kreise und die Berechnungsmethoden für Lote, Schnittpunkte und ggf. auch Tangenten u.ä. in den Informatikunterricht ein. Der Informatikunterricht dagegen wird an diesem Beispiel vor allem schätzen, dass die OOM daran überzeugend dargestellt werden kann. Dies zeigt einmal mehr, dass MU und IU unterschiedliche Ziele verfolgen. Im MU kann das Thema auch ohne eigenes Programmieren genutzt werden. Die Arbeitsweise eines DGS kann dekonstruiert werden, d.h. die Schüler beschreiben umgangssprachlich, nach welchem Algorithmus und auf Basis welcher Informationen die neue Konfiguration im Zugmodus aus der alten berechnet werden kann.

Eine ähnliche Quelle von Anregungen bietet die 3D-Computergrafik, deren Nützlichkeit für den Mathematikunterricht von Andreas Filler gegenwärtig gründlich untersucht wird (siehe Filler, 2008, in diesem Band, S. 63).

5 Prozess-Objekt-Dualität

Auf den ersten Blick ist klar, was Daten (Objekte) und was Prozeduren sind. Interessanterweise hält dies einem zweiten Blick nicht stand: Datenstrukturen und Prozeduren können identifiziert werden. In der Informatik kennt man schon seit fast 30 Jahren die Entstehung von Objekten aus Prozeduren (Operationen) und nutzt sie z.B. bei der Implementation von Klassensystemen in Lisp. In der Mathematikdidaktik hat diese Erkenntnis verspätet und unabhängig Einzug gehalten. Die Lehre aus dieser Beobachtung ist, dass die Informatik in der Lage ist, sinnvolle metaphorische Beschreibungen von Lernprozessen im Mathematikunterricht zu liefern. Ein konkretes Beispiel ist die Repräsentation von nichtabbrechenden Dezimalbrüchen. Schüler haben regelmäßig ontologische Probleme mit Dezimalzahlen mit unendlich vielen, nichtperiodischen Stellen, die ja nie vollständig angegeben werden können. Eine informatorische Modellierung einer solchen Dezimalzahl ist gegeben durch eine realisierbare Funktion (also durch einen Algorithmus), die als Eingabe die Anzahl der zu berechnenden Stellen nimmt und einen entsprechenden Näherungswert produziert. Ähnlich wie auch bei der Beherrschung des Grenzwertbegriffs werden die „unendlich vielen“ Stellen durch „beliebig viele“ Stellen ersetzt.

Eine weitere metaphorische Leistung der Informatik liegt im Bereich der Begriffsbildung. Angenommen, es soll der folgende größte gemeinsame Teiler berechnet werden: $ggT(1, 1111^{4}444)$. In den meisten Programmiersprachen würden zunächst die Argumente ausgewertet, das heißt die Exponentiation wird ausgerechnet — eine erhebliche Arbeit mit riesigem Ergebnis. Dieser Wert wird allerdings gar nicht benötigt, denn die erste Zahl im Aufruf von ggT ist 1 und also muss auch der ggT selbst 1 sein. Aus der Einsicht, dass solche Situationen auch in weniger künstlichen Situationen vorkommen können, wurden funktionale Programmiersprachen mit sogenannter *lazy evaluation* entwickelt. Diese schieben die Auswertung von Ausdrücken solange wie möglich auf. Genau das passiert auch bei der Begriffsbildung: $5/2$ und $\sqrt{3}$ sind zunächst Handlungsanweisungen. Die löbliche Faulheit besteht darin, diese Operationen zunächst nicht auszuführen, die Operationen also „einzufrieren“, so dass sie zu einem handhabbaren Eisblock (Objekt) werden. Bei Bedarf kann dieser aufgetaut werden — und dann zeigt sich seine Bedeutung. Dies liefert ein durch die Informatik gestütztes Modell der Begriffsbildung, das wichtige Erkenntnisse stützt, die bereits auf anderem Wege gewonnen wurden, etwa das Begriffsbildung kein schneller Prozess ähnlich einer mathematischen Definition sondern ein aktiver Konstruktionsschritt ist. Nicht ganz so offensichtlich ist die Folgerung, zum Zwecke der Begriffsbildung Aktivität zu Gunsten von Reflexion gezielt zurück zu stellen.

6 Methoden-Werkzeugkasten

Methoden der Informatik können nicht nur als Metapher in didaktischen Überlegungen nützlich sein, sondern u.U. lassen sie sich unmittelbar im Unterricht einsetzen. Hier sollen einige Beispiele zur Algebra zusammen gestellt werden.

Kortenkamps Klammergebirge (siehe Kortenkamp, 2008, in diesem Band, S. 77) nutzen Methoden wie sie auch bei der Programmierung von Parsern verwendet werden, um Schülern beim Umgang mit geklammerten Termen zu helfen. Graphische Beschreibungsmethoden erleben in der Informatik eine Blüte, im Mathematikunterricht sind dagegen die Termbäume so selten geworden, dass sich viele Lehramtsstudenten nicht mehr daran erinnern.

Objekte für den schnellen Zugriff zu sortieren ist eine bekannte Technik. Interessante Diskussionen mit Schülern habe ich angesichts der Ordnung in einer Formelsammlung erlebt zur Frage, wie man Terme (alphabetisch?) sortieren kann. Dies ist eine wunderbare offene Aufgabe, bei der viel sonst nur implizites Wissen über den Aufbau von Termen explizit gemacht werden muss.

Eine weit verbreitete Strategie der Softwareentwicklung ist die Model-View-Abstraktion, d.h. die Daten und Methoden, die die Modellbeschreibung betreffen, sollen getrennt werden von denen, die die Darstellung betreffen. Der gleiche Term, verschiedene Schreibweisen, das kennt die Mathematik schon lange, hier aber noch mal eine neue Anregung über Termschreibweisen und die Übersetzung zwischen ihnen nachzudenken.

Ähnliches gilt für Normalformen und kanonische Formen von Termen. Wenn die Schüler im Sinne der Metakognition über ihre Möglichkeiten zur Termumformung reflektieren, helfen diese Begriffe aus dem Bereich der Computeralgebra bei der Systematisierung!

Ebenfalls aus der Computeralgebra bekannt ist das Pattern-Matching, also die Problemstellung, einen Term an ein bestimmtes Muster anzupassen. Das liefert viele Aufgabenstellungen unterschiedlichen Schwierigkeitsgrades. Eine eher einfache Aufgabe ist: Passt der Term $5-x$ auf das Muster $Ax+B$?

7 Problematische Aspekte

In einer unvoreingenommen Diskussion dürfen Punkte nicht ausgespart bleiben, bei denen die Informatik einen problematischen Einfluss ausübt. In der mathematischen Praxis geht man an vielen Stellen flexibel mit Begriffen um. Kanonische Isomorphien werden oft vergessen, etwa wenn die konstanten Funktionen mit den reellen Zahlen identifiziert werden, oder, ähnlich gelagert, wenn zwischen Funktion und Funktionsterm nicht konsequent unterschieden wird. In der Informatik muss man solche Fragen etwas strenger sehen, um die passenden Datentypen auswählen zu können. Andererseits werden auch mathematisch traditionell unterschiedene Objekte in Informatiksystemen oft gleich repräsentiert(z.B Listendarstellung sowohl von Vektoren als auch von Teilern einer Zahl). Auch bei der Anwendung von Informatiksystemen ist zu beachten, dass in der Mathematik häufig vorgenommene Identifizierungen nicht gelten, so gilt etwa in vielen Geometriesystemen, dass eine Parabel als Funktionsgraph etwas andere ist als eine Parabel als Kegelschnitt.

Die Informatik erfordert präzise Beschreibungen und sorgfältige Planungen, und der Informatikunterricht fordert diese ein. Dies geht teilweise parallel zu Zielen des Mathematikunterrichts, aber es gibt auch Differenzen. Der Mathematikunterricht versucht u.a. auch zu einem experimentellen, interaktiven Arbeiten anzuregen. Dies sind Arbeitsweisen, die in der informatorischen Praxis zwar auch vorkommen, die aber von der stark durch Theorie geleiteten Informatikdidaktik als defizitär angesehen werden. Zwar gibt es auf diesem Feld eine leichte Entspannung, aber das Ziel der konsequenten Objektorientierten Modellierung vor der (und oft auch ohne) Realisierung ist doch bestimmend. Experimentelle Aktivitäten mit dem Rechner müssen im Mathematikunterricht deshalb bewusst als Gegenmodell zur Praxis des Informatikunterrichts angeboten werden.

Im Zusammenhang damit steht die seit 100 Jahren im Mathematikunterricht propagierte fundamentale Idee des funktionalen Zusammenhangs. In der Informatik hat funktionales Denken zwar auch wichtige Spuren hinterlassen, der Informatikunterricht setzt gegenwärtig aber stärker auf objektorientierte denn auf funktionale Modellierung. Wenn sich die Lehrer beider Fächer über diese unterschiedliche Schwerpunktsetzung im klaren sind, muss daraus aber kein Nachteil erwachsen, im Gegenteil, Methodenpluralismus kann fruchtbar wirken! Allerdings zeigt gerade die Objektorientierte Modellierung die Ambivalenz der Beziehung der beiden Schulfächer. Aus dem Mathematikunterricht ist die Mengenlehre wieder weitgehend verschwunden, ihre informatorische Übersetzung (Klasse=Menge, Objekt=Element, Unterklasse=Teilmenge, abstrakte Klasse=Kategorie) aber schickt sich gerade an, in die (bayerische) Sekundarstufe I einzudringen. Die Freude, hier mathematische Konzepte im Informatikunterricht zu sehen, weicht der Ernüchterung, dass dieser Ansatz Gefahr laufen könnte, ähnlich zu scheitern wie die mathematische Mengenlehre. Einige der Ansätze in diese Richtung laufen darauf hinaus, dass die Schüler eine Sprache lernen (es gibt konsequenter Weise auch Abschnitte zum Lernen der „Vokabeln?), für die es keinen wirklichen Bedarf gibt. Aber etwas Neues zu lernen, nur um bekannte Dinge anders beschreiben zu können, motiviert Schüler nicht dauerhaft, es muss mit den neuen Werkzeugen auch wirklich etwas getan werden.

8 Abschlussthesen

Dieser Aufsatz hat versucht zu belegen, dass der Mathematikunterricht von vielen Ideen der Informatik profitieren kann, sowohl auf inhaltlicher wie auf methodischer Ebene. Vieles davon ist ohne Programmierkenntnisse der Schüler und Schülerinnen möglich. Für einige Anwendungen sind diese aber nützlich, um im Sinne eines handlungsorientierten, auf Konstruktion gerichteten Unterrichts nennenswerte Eigentätigkeit zu ermöglichen. Das ist vor allem auch deshalb sinnvoll, weil der Informatikunterricht sich weit von seinen mathematischen Wurzeln entfernt hat. Dabei haben neue, eigenständig informatorische Inhalte viele traditionelle mathematische Programme verdrängt. Wenn Schüler mathematische Algorithmen, wie das Sieb des Erathostenes, das Heronverfahren, den Euklidischen Algorithmus oder die Bisektion zur Lösung von Gleichungen kennen lernen sollen, ist das exklusiv eine Aufgabe des Mathematikunterrichts.

Literatur

Filler, Andreas (2008): Dynamische Aspekte von Parameterdarstellungen: Generieren von Bewegungsbahnen sowie von Geraden und Kurven als Punktmengen. In: Kortenkamp et al. (2008), 63–72

Freudenthal, Hans (1972): Mathematik als pädagogische Aufgabe. Klett

KMK (2003): Bildungsstandards im Fach Mathematik für den Mittleren Schulabschluss. Bonn: Sekretariat der Ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland Ref. IV A

Kortenkamp, Ulrich (2008): Algorithmen im Mathematikunterricht. In: Kortenkamp et al. (2008), 77–86

Kortenkamp, Ulrich, Hans-Georg Weigand & Thomas Weth (Hg.) (2008): Informatische Ideen im Mathematikunterricht. Bericht über die 23. Arbeitstagung des Arbeitskreises „Mathematikunterricht und Informatik“, Hildesheim: Franzbecker

Meyer, Jörg (2006): Ein einfacher Zugang zu t-Tests. In: ISTRON-Band, Franzbecker

Oldenburg, Reinhard (2006): Numerische Optimierung — ein schneller Weg zur Modellbildung. In: ISTRON-Band, Franzbecker

Schubert, Sigrid & Andreas Schwill (2004): Didaktik der Informatik. Heidelberg, Berlin: Spektrum Akademischer Verlag

Tietze, Uwe, Manfred Klika & H. Wolpers (1997): Mathematikunterricht in der Sekundarstufe II