HRSG.: ST. SCHUY/M. MUHR/ G. GETZINGER

BUCHBERGER

LEHR- UND LERNPROZESSE IN DER INGENIEURAUSBILDUNG

ZEITSCHRIFT FÜR HOCHSCHULDIDAKTIK

Beiträge zu Studium, Wissenschaft und Beruf

Jahrgang 9/1985 Sonderheft 10

Franz Lichtenberger/Bruno Buchberger

MATHEMATIK FÜR INFORMATIKER: EIN ALGORITHMENORIENTIERTER ANSATZ AN DER UNIVERSITÄT LINZ

Gesamtkonzept

An der Universität Linz wird seit dem Wintersemester 79/80 die viersemestrige Vorlesung "Mathematik für Informatiker" neu aufgebaut. Die Autoren wurden bei der Konzeption und Strukturierung des Gesamtzyklus von folgenden Grundgedanken geleitet:

1. Mathematik ist die Technik des rationalen Problemlösens. Der Vorgang des Problemlösens in seiner Ganzheit, beginnend bei der Analyse des meist nur sehr diffus gestellten Problems bis zur übersichtlichen Präsentation des fertigen Lösungsverfahrens und der Ergebnisse sollte deshalb im Mittelpunkt der Mathematikausbildung stehen.

2. Die Schulung der vielen sehr verschiedenen intellektuellen und psychischen Fähigkeiten, die das Lösen eines Problems vom Problemlöser erfordert (Geduld zum Zuhören; Fähigkeit, gezielte Fragen zu stellen; Sehen von Strukturen in unstrukturierten Realitäten; Präzision im Ausdruck; Verstehen und Formulieren von Sachverhalten in beliebigen Notationen; Kreativität und Flexibilität; Fähigkeit zur Nutzbarmachung vorhandener Informationen; Abstraktionsvermögen und Fähigkeit zur Anschaulichkeit etc.) fällt bei einer Ausbildung in Mathematik nicht selbstverständlich als Nebenprodukt ab. Vielmehr muß der Aspekt, daß es in der Mathematikausbildung um die Schulung aller zum Vorgang des Problemlösens notwendigen Fähigkeiten geht, sowohl vom Lehrer als auch vom Studierenden von Anfang an in bewußter

Weise verfolgt werden.

3. Die mathematischen Inhalte, die in höheren Schulen bis zur Matura angeboten werden, reichen bei weitem aus, um die Technik des Problemlösens mit der Methode der Mathematik in bewußter Weise demonstrieren und schulen zu können. Eine bewußte methodische Schulung kann deshalb an den Anfang einer Universitätsausbildung gestellt werden. Eine intensive methodische Schulung am Anfang sollte nach Meinung der Autoren die Fähigkeit zur Aneignung beliebiger mathematischer und fachwissenschaftlicher Inhalte in den folgenden Semestern entscheidend verbessern und - bei Lehrern und Studierenden - auch eine gewisse Abgeklartheit gegenüber der über alle hereinbrechenden Stoffülle ermöglichen. Auch sollten die Mathematikkenntnisse aus der Schule bei Beginn des Universitätsstudiums nicht "zum Vergessen" verurteilt, sondern als Material für weitergehende Beschäftigung willkommen geheißen werden.

4. Schulung in der Mathematischen Methode des Problemlösens wird als zwei wesentliche Pfeiler "Schulung im präzisen Sprechen" und "Schulung im korrekten Denken" haben müssen. Die Sprache der Mathematik und das Beweisen sollten deshalb nicht nur an Beispielen unbewußt "mitgelernt", sondern als Werkzeug explizit analysiert, verfeinert und trainiert werden. Dieser Aspekt spielt insbesondere für den Informatiker eine wesentliche Rolle, wo oft sehr feine algorithmische Sprachmittel mit sehr groben und wenig sorgfältig behandelten deskriptiven Sprach- und Denkwerkzeugen zusammentreffen. Aus unserem Eintreten für eine explizite Schulung in der Sprache der Mathematik und im korrekten Argumentieren sollte nicht der Gegensatz "formal-anschaulich" konstruiert werden. Wir sind nur der Meinung, daß zu einer methodischen Schulung sehr wesentlich auch eine Schulung in formalen Belangen gehört, so daß an beliebiger Stelle im Problemlösungsvorgang willentlich diejenige sprachliche Präzisionsstufe gewählt werden kann, die der jeweiligen Situation optimal entspricht.

5. Der Gedanke des Strukturierens wird oft nur im Bereich des Software-Entwurfes betont. Er durchzieht als eine Grundtechnik jedoch den gesamten Problemlösungsprozeß und kommt insbesondere bei der Problemanalyse und Problembeschreibung, bei der Entwicklung von Lösungsverfahren, bei Beweisen und schließlich bei der Präsentation und Dokumentation zum Tragen.
   Dementsprechend sollte dieser Gesichtspunkt von Anfang an in der Mathematikausbildung trainiert werden, wozu wieder eine bewußte Auseinandersetzung mit Sprachmitteln, die die Strukturierung unterstützen, notwendig ist.

6. Die Inhalte einer "Mathematik für Informatiker" sollten aus Problemen der Informatik motiviert sein.

Erstes Semester

Den hier angegebenen Grundgedanken folgend, haben wir den Vorlesungszyklus so strukturiert, daß im ersten Semester eine explizite Schulung in der "Methode der Mathematik" erfolgt. Das Skriptum zum ersten Semester ist in Buchform erschienen (/1/). Die Grobstruktur der Vorlesung "Die Methode der Mathematik" ist dabei wie folgt:

Nach einer einführenden Zusammenschau über die Methode der Mathematik und den Problemlösungsprozeß wird anhand von vier Fallstudien der Gesamtprozeß des Problemlösens mit der Methode der Mathematik demonstriert. Die Fallstudien sind dabei aus dem unmittelbaren Anlaßbereich der Informatiker genommen:

Dynamisches Programmieren,
Sortieren,
Komplexitätsanalyse eines Sortierprogramms,
Ein Nimmspiel.

Bei jeder Fallstudie werden in einer "methodischen Analyse der Fallstudie" einige Teilschritte des Problemlöseprozesses im Detail und als allgemein anwendbares Werkzeug aufbereitet. Dabei werden folgende Blöcke behandelt:

- Problemanalyse und Problembeschreibung;
- das Arbeiten mit der Literatur;
- das Präsentieren und Dokumentieren von erarbeitetem Wissen;
- die Sprache der Mathematik (deskriptive und algorithmische Sprachmittel);
- mathematische Standardmodelle und Standardprobleme;
- Korrektheit von Verfahren;
- Komplexitätsanalysen von Verfahren;
- die Technik des Beweisens.

Diese Blöcke werden je nach Anlaß in den Analysen der Fallstudien mehrmals angesprochen..

Zweites Semester

Die Lehrveranstaltung Mathematik für Informatiker II ist nun Problemlösestrategien und Algorithmentypen gewidmet, die allgemein und/oder bei speziellen Datenbereichen (Datentypen) oder bei speziellen Problemtypen anwendbar sind. Damit soll von dieser Blickrichtung aus ein inhaltlicher Überblick über die Mathematik gegeben werden.

Normalerweise würde man von einem Überblick über die Mathematik erwarten, daß einige wesentliche "Sätze" (allgemeine Einsichten über Sachverhalte in verschiedenen Gegenstandsbereichen, Datentypen) gegeben werden. Es stellt sich aber heraus, daß nichttriviale "Sätze" immer auch einen Problemlösecharakter in dem Sinne haben, daß in ihnen die Gültigkeit von Sachverhalten bzw. das Berechnen von Lösungswerten für Probleme auf die Gültigkeit einfacherer Sachverhalte bzw. auf das Berechnen von Lösungswerten für einfachere Probleme zurückgeführt wird. Umgekehrt ist ein nichttriviale algorithmische Lösung eines Problems immer nur durch zusätzliches mathematisches Wissen ("Sätze") möglich. (Es gilt der

Merksatz:

Mehr mathematisches Wissen -----→ besserer Algorithmus.)

Beispiel: Der Hauptsatz der Integralrechnung reduziert das Problem, ein bestimmtes Integral zu berechnen (d. h. im wesentlichen ein Supremum über unendlich viele endliche Summwerte zu bilden) auf

das Problem, eine Stammfunktion zu finden (dies ist oft durch einen endlichen Symbolmanipulationsvorgang möglich) und diese Funktion an zwei Stellen auszuwerten.

Aufgrund dieses Parallelismus zwischen mathematischem Wissen und Algorithmen ist es ein natürlicher (wenn auch ungewöhnlicher) Zugang, einen Überblick über die Mathematik durch Betrachtung und Einübung von Algorithmentypen bzw. Problemlösestrategien zu geben. (Es wird sich herausstellen, daß der Parallelismus sich auch sehr konkret in einem Parallelismus zwischen Wissenstypen und Algorithmentypen niederschlägt).

Für Informatiker erscheint der Zugang zur Mathematik über Algorithmentypen besonders angebracht, da ja der Informatiker wesentlich ein Spezialist im Konstruieren von Algorithmen ist. Dies steht im Gegensatz zum Mathematiker, der sich mehr mit der systematischen Erweiterung des mathematischen Wissens befaßt. Für diesen ist eine Strukturierung und Präsentation der Mathematik nach verschiedenen (axiomatisch charakterisierten) Gegenstandsbereichen (Datentypen) natürlicher und interessanter. Dies steht ferner auch im Gegensatz zum Anwender der Mathematik, der zunächst nur sein Problem artikulieren kann und will und sich die Lösung dann von einem Mathematiker und/oder Informatiker erwartet. Dabei sind die Probleme des Anwenders in Ausdrücken seiner realen Umgebung (Realwissenschaft) formuliert. Für den Anwender wäre eine Strukturierung der Mathematik nach Problemtypen bzw. nach Anwendungsgebieten am wünschenswertesten (leider kaum durchgeführt!).

In der Tat wird eine Klassifikation der Mathematik nach dem Algorithmentyp noch sehr wenig benutzt. Erst in den letzten 5 bis 10 Jahren sieht man in einigen wenigen Büchern über den Entwurf und die Analyse von Algorithmen erste Versuche, mathematische Lösungsverfahren, Algorithmen, nach ihrer Struktur, den zugrundeliegenden Basisideen, zu klassifizieren. In expliziter Form findet man in den erwähnten Algorithmenbüchern (z. B. /2/) Kapitel der Art

dynamische Programmierung,
backtracking-Verfahren,
die greedy Methode,
branch-and-bound Methode,
divide-and-conquer Methode.

In dieser Vorlesung werden folgende wichtige Algorithmentypen bzw. Problemlösestrategien besprochen und eingeübt:

Reduktion auf einfachere Probleme
rekursive Reduktion
Umwandlungen von Rekursionen in Iterationen
Möglichkeiten Ausscheiden
Optimierende Algorithmen-Transformation
Zweiparameter-Algorithmen
Parallelisieren
gleichzeitiges Berechnen vieler Instanzen eines Problems
Generalisieren
Abstrahieren, Werkzeugkiste erweitern, Formalisieren
Balanzieren
Speichern, Einführen zusätzlicher Variabler
Übergang zu anderen Datenstrukturen
Übergang von Intension zu Extension (Tabellieren)
Übergang von Extension zu Intension (Strukturablesen)
Beobachten von Berechnungen schlechter Algorithmen, Betrachten von Beispielen
Zeichnungen, Modelle betrachten
Übergang zur Metaebene
Umwandeln von impliziten Definitionen in explizite

Transformieren (Repräsentieren)
Projizieren und Interpolieren
Dekomponieren, Parsen, Abbauen
lokales Erfüllen, Prioritäten setzen
Strukturvergleich (z. B. Koeffizientenvergleich, "Ansetzen")
Erzeugende Funktionen
Einschließen, Bisektion
Koordinatentransformation
nicht-kanonisches Simplifizieren, "Vereinfachen"
kanonische Simpflifizierer, Normalformen
kritische-Paar-Bidlung
Eigenwertbildung
Unifizieren
Vervollständigen, Sättigen, Aufspannen
Abbauen - Zusammenbauen
Systematisches Aufzählen, Probieren, Generieren, Suchen
Backtracking
Branch-and-Bound
Die "greedy" Methode
Dynamische Programmierung
Konvergentes Iterieren
Vernachlässigen, Abbrechen, Abscheiden
Glätten
Diskretisieren
Linearisieren
Fortsetzen, Extrapolieren
Approximieren

Drittes und Viertes Semester

Das 3. und 4. Semester werden dann verschiedenen Teilgebieten der Mathematik gewidmet sein, die mehr im traditionellen Stil, d. h. also von der Systematik der Datentypen bzw. der Problemtypen her, entwickelt werden. Dabei werden im dritten Semester Themen aus dem Gebiet der Analysis und in vierten Semester aus der diskreten Mathematik behandelt. Hier deckt sich der Vorschlag mit jüngsten

Ideen, die an amerikanischen Universitäten entwickelt wurden (/3/) Auch dort wird vorgeschlagen, die Analysis erst im dritten Semester zu bringen und zwar nachdem im ersten Jahr Teile der diskreten Mathematik gebracht wurden.

Referenzen:

/1/ B. Buchberger, F. Lichtenberger
Mathematik für Informatiker I (Die Methode der Mathematik),
Springer, Berlin - Heidelberg - New York, 2. Auflage, 1981.

/2/ A. V. Aho, J. E. Hopcroft, J. D. Ullman;
The Design and Analysis of Computer Algorithmus.
Addison-Wesley, Reading, 1974.

/3/ A. Ralston, G. S. Young (Hrsg.),
The Future of College Mathematics.
Springer, New York - Berlin - Heidelberg, 1982.