

Karl Josef FUCHS, Universität Salzburg

Die Funktion – Basiselement der Informatik

0. Prolog

Dem aufmerksamen Leser wird nicht entgehen, dass die Überschrift weder mit einem Rufzeichen noch mit einem Fragezeichen abgeschlossen ist. Im Falle eines Rufzeichens am Ende des Beitragstitels wären die folgenden Darstellungen als Rechtfertigung für diese Behauptung zu sehen, im Falle eines Fragezeichens würde man eine Antwort auf die Darlegungen spätestens im Epilog zu diesem Beitrag erwarten.

1. Eine aktuelle Bestandsaufnahme

Richten wir also zunächst unseren Blick auf den Stellenwert des Funktionsbegriffs in wesentlichen Beiträgen zur Didaktik der Informatik. In der aktuellen Ausgabe *Didaktik der Informatik* von Peter Hubwieser [2] finden wird das Thema **prominent** in zwei Kapiteln (Funktionale Modellierung - Modellierung mit Hilfe von Funktionen) vertreten. Im Lehrbuch *Didaktik der Informatik* von Sigrid Schubert und Andreas Schwill [8] finden wir unter der Bewertung der verschiedenen programmiersprachlichen Zugänge das Prädikat **funktional**. Auch im *Modulkonzept – einem zeitgemäßen Ansatz zur informatischen Bildung für alle Schülerinnen* von Ludger Humbert [3] ist die Funktionale Modellierung als fächerkoordinierende Idee zwischen Mathematik und Informatik namhaft vertreten. Wir könnten also meinen, eine gewisse **Quantität** spricht für die Funktion als Basiselement der Informatik.

Die Publikationen von Hans – Stefan Siller von der Universität Salzburg zeigen schließlich, dass es sich auch um ein **aktuelles** Thema der Mathematik- bzw. Informatikdidaktik handelt [6; 7].

Nun aber zu zwei zentralen Aspekten der Funktionsbetrachtung.

2. Die historische Kontinuität

Für diese Annahme ist es zunächst unbedingt notwendig auf die junge Entwicklungsgeschichte der Didaktik der Informatik zu blicken:

Was wollen wir als den Beginn dieser wissenschaftlichen Disziplin ansetzen?

Ist es der Zeitpunkt des Erscheinens spezifischer fachdidaktischer Zeitschriften, so wäre dies bei der *Informatica didactica* das ausgehende vorige

Jahrhundert, bei der Zeitschrift LOGIN der Anfang der 90er Jahre. Grundlegende Arbeiten von Petra Knöb [4] über die *Fundamentalen Idee der Informatik im Mathematikunterricht* oder die *Fundamentalen Ideen der Informatik* von Andreas Schwill [5] aus Didaktik der Informatik sind zum einen Ende der 80er Jahre, zum anderen anfangs der 90er Jahre, als Bücher zur Didaktik der Mathematik erschienen bzw. wurden in Zeitschriften zur Didaktik der Mathematik publiziert, wobei Schwill den Funktionsbegriff als Basiselement der Informatik anspricht [5, Funktionale Programmiersprachen, S. 21]. Es wird daher sinnvoll sein, unseren historischen Blick über die ‚reine‘ Didaktik der Informatik als selbstständige Wissenschaft hinaus in die Didaktik der Mathematik weiterzuführen, wollen wir zu tatsächlicher **Kontinuität** des Begriffs gelangen.

Erweitert man den Blick, so wird man an die Diskussion der LOGO – Philosophie am Ende der 80er Jahre erinnert [1; 9]. Zieht man nun außerdem in Betracht, dass LOGO ein ‚Abkömmling‘ von LISP (= LIST Processing), einer streng funktionalen Sprache, ist, so verwundert es wenig, dass sich viele dieser Diskussionsbeiträge auch mit dem Grundelement ‚Funktion‘ kritisch auseinander setzten.

Gehen wir noch weiter in die Vergangenheit: Wir werden sehen, dass auch bei der Einführung des Taschenrechners Ende der 70er Jahre das Thema Funktion einen zentralen Stellenwert einnahm. Gemeint ist dabei die Diskussion über so genannte UPN – Rechner (UPN = Umgekehrte Polnische Notation – abgeleitet von der Polnischen Notation des Jan Łukasiewicz), die die klammerfreie funktionale PostFix – Notation verwendeten, und AOS – Rechner (= Algebraic Operation System), bei denen die Eingabe in der üblichen algebraischen InFix – Notation erfolgte.

Während die Berechnung des Ausdrucks $(3 + 5) * 7$ bei einem AOS – Rechner gemäß der Angabe als $(3 + 5) * 7 (=)$ zu erfolgen hat, muss für einen UPN – Rechner der Ausdruck funktional aufbereitet werden und lautet somit: 3 (ENTER) 5 + (ENTER) 7 *.

Damit noch nicht genug. Da der Code in der bereits genannten funktionalen Programmiersprache LISP aus vordefinierten und selbst definierten Funktionen bestand, war die Implementierung von Funktionen bereits Ende der 50er Jahre ein Thema der Informatik.

Die mittlerweile weitgehend unbekannt Programmiersprache FORTH (= FO(U)RTH Generation Languages, kurz: 4GL), welche die UPN – Notation bei der Kodierung verwendet, wurde in den 60er Jahren entwickelt.

3. Die Funktion als Brückenelement zwischen Mathematik und Informatik

Um nachzuweisen, dass es sich bei der Funktion um einen Brückenbegriff zwischen Mathematik und Informatik handelt bzw. seit geraumer Zeit handelte, muss es uns gelingen, nachzuweisen, dass beide Sichtweisen, also von der Mathematik zur Informatik, aber auch jene von der Informatik zur Mathematik wesentlich zu einem umfassenden Bild von Funktionen beitragen.

Fakt ist, dass die Mathematik als Ausgangspunkt die Eigenschaften einer Funktion definiert, wobei jedem Element einer Ausgangsmenge **GENAU EIN** Element der Zielmenge zugeordnet wird. Diese ‚strenge‘ Festsetzung wurde von der Informatik in so genannten Funktionalen Programmiersprachen bzw. Anwendersystemen (Tabellenkalkulationsprogramme) implementiert und realisiert.

Was steuert aber nun umgekehrt die Informatik zum mathematischen Verständnis von bzw. Umgang mit Funktionen bei?

Durch ihre vielseitige Nutzung von funktionalen Darstellungen in Programmiersprachen sowie Anwendersystemen gewinnt die Mathematik in zweifacher Hinsicht.

Zum einen durch die Verwendung **sprechender Namen** wie

= 1 + GANZZAHL(ZUFALLSZAHL()*6)

(für die Simulation eines Würfelwurfs mit EXCEL) oder

(last(quote (a b c)))

(zur Extrahierung von c aus der Liste (a b c) in LISP).

Denken wir im Gegensatz dazu an unsere Schüler/Innen, für die Funktionen ausschließlich nur den Namen *f* tragen dürfen.

Zum anderen gewinnt die Mathematik durch die in der Informatik bei der Implementierung nahezu selbstverständliche Verwendung

- einer **beliebigen Anzahl von Funktionsargumenten** (bis hin zu nil – adischen Funktionen ohne Argument wie die zuvor verwendete ZUFALLSZAHL()),
- von **Verkettungen von Funktionen** (wofür ebenfalls die beiden zuvor genannten Kodierungsbeispiele Belege sind).

Denken wir an unsere Schüler/Innen, die im Mathematikunterricht nur Funktionen mit einem Argument kennen lernen, wobei dieses nur x lauten soll. Jeder Lehrende / Jede Lehrende kann, bezogen auf diese Überlegungen, sicherlich über Probleme im Zusammenhang mit der Kettenregel der Differentialrechnung berichten.

4. Epilog

Kehren wir zum Anfang zurück:

Würde ein Rufzeichen an das Ende des Beitragstitels stehen, hoffe ich, überzeugende Gründe zur Rechtfertigung der Funktion als Basiselement informatischer (Aus)bildung geliefert zu haben.

Im Falle des Fragezeichens würde ich die Fragestellung mit **JA** beantworten. Die Darstellungen in den Abschnitten 1 bis 3 wären dann als Argumentationen für meine Zustimmung anzusehen.

Literatur

- [1] Bender, Peter (1987): *Kritik der LOGO – Philosophie*. In: JMD – Zeitschrift der Gesellschaft für Didaktik der Mathematik, Heft 1./2., S. 3 - 103.
- [2] Hubwieser, Peter (2007): *Didaktik der Informatik*. Berlin, Heidelberg: Springer Verlag.
- [3] Humbert, Ludger (2002): *Das Modulkonzept – ein zeitgemäßer Ansatz zur informatischen Bildung für alle Schülerinnen*. In: Informatica didactica 5; <http://www.informatica-didactica.de>
- [4] Knöß, Petra (1989): *Fundamentale Ideen der Informatik im Mathematikunterricht*. Deutscher Universitätsverlag.
- [5] Schwill, Andreas (1994): *Fundamentale Ideen der Informatik*. Zentralblatt für Didaktik der Mathematik, S. 20 – 31.
- [6] Siller, Hans - Stefan (2008): *PROGRAPH – Diagrams – A new old system for teaching functional modeling*. Angenommen zur Präsentation auf der ICME 11, TSG 22, Monterrey.
- [7] Siller, Hans - Stefan (2008): *Two Subjects Sharing One Concept – Functions as Structuring Principles in Mathematics and Informatics Education*. Angenommen zur Präsentation auf der bzw. zur Publikation in den Proceedings zur ISSEP, Torun.
- [8] Schubert, Sigrid; Schwill, Andreas (2004): *Didaktik der Informatik*. Berlin: Spektrum Akademischer Verlag.
- [9] Ziegenbalg, Jochen (1987): *Anmerkungen zur „Kritik der Logo-Philosophie“*. In: JMD, Heft 4, S. 305 – 313.