WEBER, Christof
Luzern

Zur Bedeutung von Algorithmen und algorithmischem Denken im Mathematikunterricht: ein Versuch

Algorithmen und algorithmisches Denken sind zentrale Elemente der mathematischen Kultur. Sie durchziehen daher die Schulmathematik auf allen Stufen. Ihr Bildungswert wird jedoch gelegentlich in Frage gestellt. Ist dies der Grund für das Fehlen einer breit abgestützten Begriffsfassung algorithmischen Denkens? Und was könnte algorithmisches Denken im Sinne eines „deep knowledge of algorithms“ (Fan & Bokhove, 2014, S. 484) bedeuten? In diesem theoretischen Beitrag werden einige zentrale Eigenschaften von Algorithmen sowie einige mögliche Befassungen mit Algorithmen vorgestellt, die für den Mathematikunterricht der Zukunft bedeutsam sein könnten.

Eigenschaften von Algorithmen

Algorithmen sind eine spezifische Art, Probleme zu lösen (Maurer 1998). Beispiele aus dem Unterricht sind die schriftliche Multiplikation, die Probedivision zur Überprüfung einer natürlichen Zahl auf Primzahleigenschaft oder die „Tachometerstrategie“ zur Erzeugung aller möglichen Kombinationen mehrerer Eigenschaften. Jeder dieser Algorithmen …

(1) … besteht aus einer *vordefinierten* und *festen Folge* von endlich vielen Anweisungen;

(2) … löst eine *Klasse* von (un)endlich vielen Einzelproblemen;

(3) … löst jedes Einzelproblem in *endlich vielen* Schritten; dazu geht er *systematisch* vor.

Die Eigenschaft (1) stellt sicher, dass die Reihenfolge der Anweisungen bereits vor ihrer Ausführung feststeht und nicht je nach Einzelproblem vor oder während des Prozesses geändert wird. Die Eigenschaft (2) stellt sicher, dass der Algorithmus nicht nur ein einziges Einzelproblem (wie „12 · 3= ?“), sondern das verallgemeinerte Problem löst („a · b = ?“). Die Eigenschaft (3) besagt, dass der Algorithmus nach endlich vielen Schritten (und damit in endlicher Zeit) terminiert und die korrekte Lösung angibt. Nur selten wird explizit gemacht, dass er dazu systematisch vorgehen muss (Maurer 1998). Dies könnte daran liegen, dass jeder Algorithmus zwangsläufig systematisch vorgehen muss, um die Lösung in endlich vielen Schritten zu finden. Will man etwa alle möglichen Kombinationen zweier Eigenschaften generieren und geht unsystematisch vor, weiß man zu keinem Zeitpunkt, ob man bereits alle möglichen Kombinationen gefunden hat. Folglich würde die Suche unbegrenzt viele Schritte erfordern und der Algorithmus nicht terminieren.

In: P. Ebers, F. Rösken, B. Barzel, A. Büchter, F. Schacht & P. Scherer (Hrsg.),
Beiträge zum Mathematikunterricht 2024.
57. Jahrestagung der Gesellschaft für Didaktik der Mathematik. WTM.
https://doi.org/10.37626/GA9783959872782.0

In diesem Zusammenhang weisen Autoren wie Modeste (2012) auf die Variabilität der Anzahl ausgeführter Schritte hin. Jeder Algorithmus …

(4) … löst jedes Einzelproblem in einer Anzahl von Schritten, die von der „Größe“ der Elemente des Einzelproblems *abhängt*.

Die Eigenschaft (4) drückt aus, dass ein Algorithmus zur Lösung eines Einzelproblems in der Regel mehr bzw. weniger Schritte ausführt, als er Anweisungen enthält. Dies wird durch Strukturen wie Iterationen und Verzweigungen realisiert. Sie besagt aber auch, dass eine Formel wie die Lösungsformel für quadratische Gleichungen noch kein Algorithmus im eigentlichen Sinne ist (Modeste, 2012): Sie erfüllt die Eigenschaft (4) nicht, da die Anzahl der Schritte zur Berechnung der Lösung nicht davon abhängt, ob die Koeffizienten der zu lösenden quadratischen Gleichung „groß“ oder „klein“ sind. Darüber hinaus macht die Eigenschaft (4) deutlich, dass Vergleiche wie „Algorithmen sind wie Rezepte“ hinken. Rezepte bestehen zwar aus einer festen Abfolge endlichen vieler Anweisungen (die, wenn sie korrekt ausgeführt werden, zu einer „Lösung“, dem Menü, führen). Da sie jedoch keine Iterationen oder Verzweigungen enthalten, sollten sie eher als Verfahren oder Strategien und nicht als Algorithmen im engeren Sinne betrachtet werden.

Ein weiteres Missverständnis betrifft die Gleichsetzung von Algorithmen mit Programmen, das „amalgame entre algorithme et programme“ (Modeste, 2012, S. 127). So können Algorithmen zwar in eine Programmiersprache übertragen werden, um als Programm auf einem Computer ausgeführt zu werden. Aber: Die Mathematik befasste sich schon mit Algorithmen, lange bevor es Programmiersprachen und Computer gab. Außerdem basiert nicht jedes Programm auf einem Algorithmus im engeren Sinne; für manche Anwendungen ist es geradezu unerlässlich, dass ein Programm nicht abbricht (z.B. die Steuerung von Ampelanlagen). Für diese „Vermischung“ von Algorithmen und Programmen dürfte mitverantwortlich sein, dass wir über keine standardisierte Notation zur Darstellung von Algorithmen verfügen. Es ist daher naheliegend, auf eine Schreibweise aus der Informatik (Programmiersprache, Pseudocode, Flussdiagramm usw.) zurückzugreifen.

Algorithmen in der Schule und im Lehrplan

Algorithmen sind aus mathematischer Sicht eine feine Sache: Mit ihrer Hilfe werden Aufgaben, die zunächst als anspruchsvolle Probleme erscheinen, zu reinen Routineaufgaben. Im schulischen Kontext wird diese Stärke jedoch gerne zu einem zweischneidigen Schwert: Einerseits erleichtern Algorithmen den Lösungsprozess, andererseits muss man nicht verstehen, warum ein Algorithmus ein Problem löst, wie er funktioniert; es reicht völlig aus, seinen Anweisungen zu folgen, um zur Lösung zu gelangen.

Dementsprechend begannen in den 1980er Jahren empirische Untersuchungen, das Lernen und Lehren von Algorithmen (v.a. in der Grundschule) zu problematisieren. „Algorithmisch“ wurde zunehmend mit „Auswendiglernen“ assoziiert und tendenziell abwertend verwendet, etwa als Gegensatz zu „verständnisorientiert“ oder „kreativ“. Dies mag ein Grund dafür sein, dass einige Länder die schriftliche Division und andere Standardalgorithmen aus ihren Lehrplänen gestrichen haben (z.B. Kanada, Fan & Bokhove, 2014).

Aber hat man damit nicht das Kind mit dem Bade ausgeschüttet? Liegt das Problem nicht weniger in den Algorithmen selbst als in der Art, wie wir sie im Unterricht behandeln (ebd., S. 490)? Was könnte algorithmisches Denken im Sinne eines „deep knowledge of algorithms“ (ebd., S. 484) bedeuten?

Befassungen mit Algorithmen: Zwischen Nachmachen und Entwickeln

Im Kontext des Lehrens und Lernens von Mathematik spricht man etwa von „funktionalem Denken“ oder „kombinatorischem Denken“. Auch das Adjektiv „algorithmisch“ erhält durch den Zusatz „Denken“ derzeit neuen Glanz. Nicht zuletzt dürfte auch die bildungspolitische Forderung nach „computational thinking“ in der (Informatik-)Bildung dafür verantwortlich sein, dass „algorithmisches Denken“ gerne in Weißbüchern, die die Bildung der Zukunft skizzieren, als Bildungsziel gefordert wird (z.B. OECD, 2023).

Dennoch scheint es wenig Konsens darüber zu geben, was algorithmisches Denken im Mathematikunterricht bedeuten kann und soll. In Anlehnung an die Auffassung sprachlich verwandte Konstrukte (funktionales Denken etc.) könnte algorithmisches Denken als eine menschliche Aktivität verstanden werden, die „typisch“ für den Umgang mit Algorithmen ist (Kortenkamp & Lambert, 2015; Maurer, 1998). Diese Umschreibung verschiebt die Frage nach den konstitutiven Elementen des algorithmischen Denkens auf die Frage nach möglichen unterrichtlichen Befassungen mit Algorithmen.

Die traditionelle Behandlung von Algorithmen im Unterricht besteht darin, dass ein fertiger Algorithmus von einer Lehrperson *vorgeführt* und von der Klasse *nachgemacht* wird. In Absetzung davon werden die Lernenden deshalb manchmal ermutigt, eigene, idiosynkratische Lösungsstrategien und Algorithmen zu *entwickeln*, entweder „unplugged“ oder mit Hilfe programmierbarer Taschenrechner. Fasst man diese beiden Befassungen als Pole eines Spektrums auf, so sind mindestens zwei weitere dazwischen denkbar:

Eine dritte Art, sich im Unterricht mit Algorithmen zu befassen, ist die *Analyse* ihrer Effektivität, ihrer Effizienz oder ihrer Grenzen: *Aus welchen Gründen löst ein Algorithmus tatsächlich das Problem, das er zu lösen verspricht? Löst der Algorithmus das Problem mit der minimalen Anzahl von Rechenschritten? Welche Einzelprobleme gehören nicht mehr zu der Klasse von*

Problemen, die er löst? Während die Mathematik die Effektivität von Algorithmen beweist, stellt die Informatik deren absolute Effizienz sicher (Komplexitätstheorie). Im Mathematikunterricht kann zumindest die relative Effizienz thematisiert und analysiert werden (Modeste, 2012).

Eine weitere, vierte Form ist das *Vergleichen* von Algorithmen. So können verschiedene Algorithmen, die dasselbe Problem lösen (z.B. die schriftliche, ziffernweise Multiplikation von rechts nach links mit der ziffernweisen Multiplikation von links nach rechts, die Polynomdivision mit dem Horner-Schema, eine idiosynkratische Strategie mit dem Standardalgorithmus) miteinander verglichen werden: *Welcher Algorithmus ist kürzer? Welcher ist schneller? Welcher Algorithmus ist (für uns) übersichtlicher?* Darüber hinaus können in Fällen, in denen zwei Probleme mathematisch verwandt sind (z.B. die schriftliche Division und das Logarithmieren (Weber, 2019)), die zugehörigen Algorithmen miteinander verglichen und reflektiert werden.

Diese beiden Befassungen zielen auf ein „thinking *about* algorithms" (Maurer, 1998, S. 24): Algorithmen werden von einem Werkzeug zur Lösung zu einem *Objekt*, das selbst untersucht wird. Der damit einhergehende Perspektivenwechsel ist charakteristisch für das Verstehen mathematischer Begriffe (Sfard 1991): So wie Schülerinnen und Schüler lernen, über Zahlen oder Funktionen zu sprechen und nachzudenken, könnte ein Bildungsziel der Zukunft sein, Algorithmen nicht nur fehlerfrei ausführen zu können, sondern auch *über* Algorithmen sprechen und nachdenken zu können.

Literatur

Fan, L., & Bokhove, C. (2014). Rethinking the role of algorithms in school mathematics: A conceptual model with focus on cognitive development. *ZDM–The International Journal on Mathematics Education*, *46*, 481–492.

Kortenkamp, U., Lambert, A. (2015). Wenn …, dann … bis …: Algorithmisches Denken (nicht nur) im Mathematikunterricht. *mathematik lehren*, *188*, 2–9.

Maurer, S. B. (1998). What is an algorithm? What is an answer? In L. J. Morrow & M. J. Kenney (Hrsg.), *The teaching and learning of algorithms in school mathematics* (S. 21–31). National Council of Teachers of Mathematics.

Modeste, S. (2012). Enseigner l'algorithme pour quoi? Quelles nouvelles questions pour les mathématiques? Quels apports pour l'apprentissage de la preuve? [Doktorarbeit, Universität Grenoble]. https://theses.hal.science/tel-00783294.

OECD (2023). *The future of education and skills: OECD learning compass for mathematics*. OECD. https://www.oecd.org/education/2030/OECD-Learning-Compass-for-Mathematics-2023-13-Oct.pdf.

Weber, C. (2019). Comparing the structure of algorithms: the case of long division and log division. In U. T. Jankvist, M. van den Heuvel-Panhuizen, & M. Veldhuis (Hrsg.), *Proceedings of the Eleventh Congress of the European Society for Research in Mathematics Education* (S. 698–705). Utrecht University.