

Michael KALLWEIT, Bochum

Halboffene Eingabeformate für digitale Mathematikaufgaben

Durch den Einsatz von Computeralgebrasystemen in der automatischen Auswertung von digitalen Aufgaben sind eine Vielzahl neuer Möglichkeiten für die Mathematiklehre entstanden. Wir stellen in dem Vortrag ein neues Eingabeformat vor und vergleichen diesen halboffenen Typ mit herkömmlichen geschlossenen und offenen Formaten hinsichtlich Akzeptanz und didaktischem Potenzial.

Motivation

Für den Einsatz digitaler Medien und Werkzeuge in der Hochschullehre gibt es viele Motivationen. Neben Zeit- und Ortsunabhängigkeit, die eine Flexibilität und Individualisierung ermöglichen, ist durch die COVID-19-Pandemie eine neue Notwendigkeit entstanden Digitalisierung noch stärker in viele Bereiche des Lebens einzubringen. Darüber hinaus gibt es jedoch auch weitere Vorteile, die es auszuloten gilt und in der Technologie selbst stecken. Ganz im Sinne der Transformation-Stufen des SAMR-Modells (Puentedura 2006) sind gerade die Einsatzszenarien die interessantesten, die ohne digitale Tools nur schwer oder gar nicht möglich wären und Neues eröffnen.

Aufgaben nehmen in der Beschäftigung mit Mathematik eine zentrale Rolle ein. In Schule und Hochschule sind sie ein wichtiges Mittel der Lehre und des Lernens.

Digitale Aufgabenformate für die Mathematik

Mit einer technischen Perspektive lassen sich digitale Aufgabenformate grob in zwei Klassen einteilen. Dabei ist die Freiheit bei der Eingabe der ausschlaggebende Faktor; also wie und in welcher Form die Antwort des Nutzers an das System übergeben wird. Bei geschlossenen Formaten gibt es klar definierte und begrenzte mögliche Antworten, die dem Teilnehmer zur Auswahl angezeigt werden. Bei offenen Formaten werden keine Antworten vorgegeben, sondern die Eingabe geschieht in freier Form (z.B. als Eingabe in ein oder mehrere Textfelder).

In den gängigen Learning-Management-Plattformen finden sich eine Vielzahl von Aufgabenformaten. Das offizielle Plugin-Verzeichnis von Moodle zählt allein 75 sogenannte Question Types. Darunter finden sich viele Variationen des Multiple-Choice-Prinzips (wie Zuordnung, Sortierung, Drag'n'Drop) aber auch Formate mit freien Eingabemöglichkeiten (wie Kurzantwort, Reguläre Ausdrücke, Lückentext, Freitext).

Speziell für die Mathematik stehen neben sehr elementaren Aufgabentypen (wie numerische Eingabe oder Berechnete Formel) auch das Plugin STACK zur

Verfügung. Mit diesem sind automatisch auswertbare Aufgaben mit individuellem Feedback möglich (Kallweit 2016). Hierbei wird im Hintergrund auf das Computeralgebrasystem Maxima zurückgegriffen. Damit sind randomisierte Aufgabenstellungen und verschiedene Eingabeformate möglich, eine Übersicht steht in Tabelle 1.

Ein Argument, dass oftmals gegen freie Eingaben durch Nutzer gegeben wird, ist die Problematik der Eingabe der Antworten. Mathematische Ausdrücke müssen in technischen Systemen durch eine festgelegte Syntax eingegeben werden. STACK nutzt zur Kompensation dieser Schwierigkeiten den Ansatz der Validierung vor der Abgabe: Die Interpretation der Eingabe durch das System (speziell das Computeralgebrasystem) wird direkt während des Eintippens angezeigt und muss gegebenenfalls durch einen Klick auf „Prüfen“ bestätigt werden. Erst dann wird das Eingegebene ausgewertet. Diese Auswertung geschieht durch die Ausführung eines sogenannten Rückmeldebaumes, der Bewertung und Feedbackgenerierung vereint und vom Aufgabenersteller modelliert werden muss.

Problematik von geschlossenen Formaten

Aufgrund der festen Anzahl von Distraktoren können geschlossene Formate nicht auf die Spezifität der Lösungsideen eines Teilnehmers reagieren: Wenn dieser eine sehr konkrete Antwort im Kopf hat, gibt es zu viele Möglichkeiten (die nicht benötigt werden und vielleicht verwirren). Wenn er nur eine ungefähre Vorstellung von der Antwort hat, kann die vorgelegte Liste zu viele Informationen über die richtige Lösung verraten oder zumindest einige deutliche Hinweise geben.

Im folgenden stellen wir genau ein solches neues Aufgabenformat vor, welches auf die Konkretheit der Lösungsideen eines Teilnehmers reagieren kann und sich daher zwischen offenen und geschlossenen Formaten positioniert und somit halboffen genannt werden kann.

Autocomplete-Choice

Das Konzept der automatischen Vervollständigung von Computer-Eingaben existiert schon seit den 1980ern. Ziel dabei ist es, die Benutzereingaben sinnvoll zu ergänzen und Tipparbeit zu ersparen.

Bei der Autovervollständigung von Eingaben muss der Benutzer zunächst einen Startpunkt finden und diesen eingeben. Danach präsentiert das System mögliche passende Antworten, aus denen der Benutzer schließlich auswählen kann. Wir haben diese Idee adaptiert und für digitale Aufgaben nutzbar gemacht.

Unser Eingabetyp Autocomplete-Choice liefert auswählbare Vorschläge passend zur gemachten offenen Eingabe. Je weniger Zeichen getippt werden, desto größer ist die Auswahlliste, d.h. es werden weniger Informationen über die Lösung(en) preisgegeben. Mehr Eingabezeichen des Benutzers führen zu einer kleineren Auswahlliste, dementsprechend werden weniger irrelevante

Antwortmöglichkeiten gezeigt. Es ist dabei aber weiterhin möglich eine freie Eingabe zu tätigen ohne etwas aus der vorgeschlagenen Liste auszuwählen.

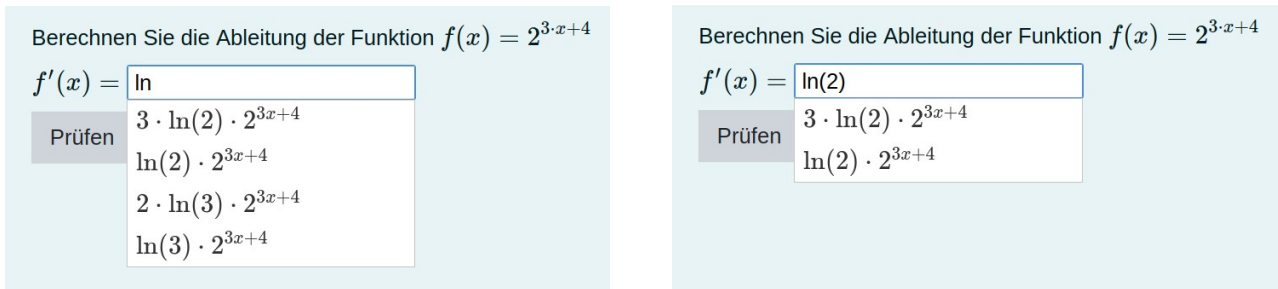


Abbildung 1: Verschiedene Vorschläge, basierend auf der getätigten Eingabe

Zusätzlich zu Textfragmenten ist es auch möglich Operatoren in der erforderlichen Syntax einzugeben um entsprechende Vorschläge zu erhalten. So liefert z.B. die Eingabe von „^“ Vorschläge, die Potenzen enthalten. Wenn man dann einen Vorschlag auswählt, wird der Ausdruck in der erforderlichen Schreibweise in das Eingabefeld eingetragen, siehe Abbildung 2.

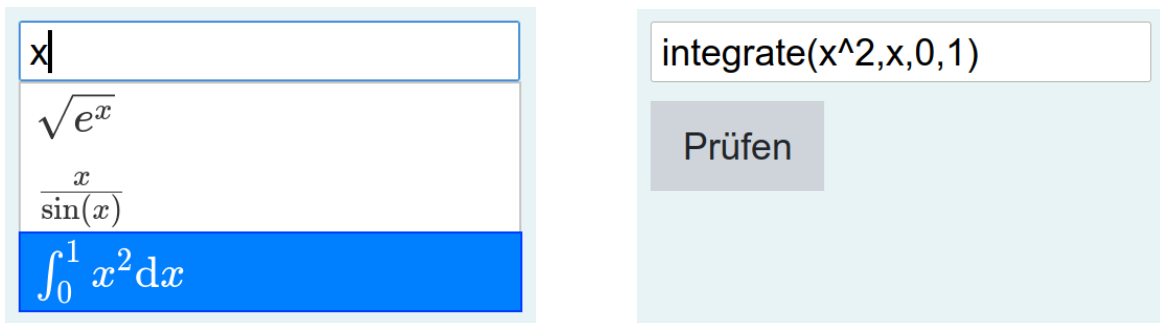


Abbildung 2: Vervollständigung mit korrekter Syntax

Umsetzung

Das Eingabeformat Autocomplete-Choice ist mit Hilfe von Javascript realisiert und verwendet die Bibliotheken JQuery, JQuery-UI und KaTeX. Es kann auf jedem Assessmentssystem genutzt werden, welches offene Eingabeformate bietet und zusätzliches HTML und Javascript im Aufgabentext erlaubt. Bei der Entwicklung wurde Wert auf eine einfache Verwendbarkeit gelegt. Wir zeigen die Nutzung beispielhaft in Moodle mit STACK, siehe auch Listing 1.

```
<div class="autocompleteme" data-tags="['(a+b)^2','a^2+b^2']" >
    [[input:ans1]] [[validation:ans1]]
</div>
<script src="autocomplete.js"></script>
```

Listing 1: Beispiel für Einbettung des Autocomplete-Features

Das vom System bereitgestellte Eingabefeld wird in eine DIV-Umgebung gesetzt, welches das Klassenattribut „autocomplete“ zugewiesen bekommt. Die in der Vervollständigung vorgeschlagenen Zeichenketten können in einem separaten data-Tag angegeben werden. Zuletzt muss noch das externe Script, welches die Autocomplete-Funktionalität bereitstellt, eingebunden werden.

Die Kombination mit STACK bietet einen weiteren Vorteil: So lassen sich die Vorschläge mit dem Computeralgebrasystem systematisch und bei Bedarf randomisiert erzeugen.

Akzeptanz von Autocomplete-Choice

Im Rahmen einer Masterarbeit wurde der Einsatz des Autocomplete-Choice Aufgabentyps pilotiert. Es nahmen 32 Schülerinnen und Schüler aus drei Oberstufenkursen an Moodle-Tests teil, die aus geschlossenen, offenen und den halboffenen Autocomplete-Choice Aufgaben bestanden. Im Anschluss wurde die Akzeptanz des Formats mit Hilfe eines vom UTAUT-Modell (Venkatesh et al. 2003) adaptierten Fragebogens erfragt. Ebenso wurde die Einschätzung der zugehörigen Lehrkräfte eingeholt.

Die Ergebnisse der Erprobung zeigen, dass halboffene Aufgaben ein Komplexitätsniveau haben, welches zwischen den altbekannten Aufgabenformaten offen und geschlossen liegt. Schülerinnen und Schüler mit einer mittleren fachlichen Selbsteinschätzung gaben dem halboffenen Format die höchsten Akzeptanzwerte. Die weitere Befragung zeigt, dass sowohl Schülerinnen und Schüler als auch Lehrkräfte den halboffenen Fragetyp als sinnvolle Ergänzung empfinden und solche Aufgaben in Zukunft häufiger nutzen würden.

Literatur

- Glasmachers, E., Kallweit, M. (2018). Adaptive Selbstlernaufgaben mit STACK. Proceedings of the first international STACK conference. <http://doi.org/10.5281/zenodo.2590188>
- Kallweit, M. (2016). CAS-unterstütztes Assessment von Mathematik. Computeralgebra-Rundbrief, Heft 59, 22-24
- Puentedura, R. (2006). Transformation, Technology, and Education. <http://www.hippasus.com/resources/tte/>
- Venkatesh, V.; Morris, M.; Davis, G.; Davis, F. (2003). "User Acceptance of Information Technology: Toward a Unified View". MIS Quarterly. 27 (3): 425–478. doi:10.2307/30036540. JSTOR 30036540.