

Leitfaden zum Fragetyp STACK

Aufgabenerstellen in optes (ILIAS)

Miriam Weigel

Akademische Mitarbeiterin optes

DHBW Mannheim

Mai 2017



Dieses Werk wurde im Rahmen des Projekts optes erstellt und steht unter der Lizenz
Creative Commons 'Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Unported'.

Das Projekt optes+ wird im Rahmen des Qualitätspakts Lehre aus Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen O1PL17012A gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Inhalt

1 Einleitung.....	3
1.1 Über optes.....	3
1.2 Über STACK.....	3
2 Erstellen einer STACK-Frage	4
2.1 Beispielaufgabe	4
2.1.1 Erste Schritte	4
2.1.2 Fragen-Variablen und Aufgabenhinweis	5
2.1.3 Spezifisches Feedback und Test-Eingaben	6
2.1.4 Rückmeldebäume.....	7
2.1.5 Testphase	10
2.2 Überarbeitung der Aufgabe	10
3 Anhang.....	14
3.1 Antwortüberprüfung.....	14
3.2 Maxima-Befehle	14
4 Quellen	15

1 Einleitung

1.1 Über optes

Das Hochschulverbundprojekt „optes - Optimierung der Selbststudiumsphase“ dient der Unterstützung des Selbstlernprozesses in der Studienvorbereitung und der Studieneingangsphase. Ein wichtiges Element sind die Lernzielorientierten Kurse mit Inhalten aus der Schul- und Hochschulmathematik. Diese Kurse umfassen theoretische Erklärungen, Trainingseinheiten zum Üben des erworbenen Wissens und Tests zur Kontrolle des Gelernten. Eine wesentliche Komponente des erfolgreichen Selbststudiums ist ein aufschlussreiches Feedback, welches möglichst automatisch erfolgen sollte. In diesem Sinne setzt das optes-Teilprojekt „Formatives eAssessment und Propädeutik“, das an der Dualen Hochschule Baden-Württemberg Mannheim angesiedelt ist, den Schwerpunkt für die Aufgabenentwicklung im Selbststudium auf die automatisierte Rückmeldung. Die technische Unterstützung durch die Lernplattform ILIAS ermöglicht unterschiedliche Fragetypen für die Aufgabenstellung in den optes-Kursen.

1.2 Über STACK

Für das Training und den Selbsttest kommt oft der Fragetyp STACK zum Einsatz, der mittels des Computer-Algebrasystems Maxima an das Test- und Aufgabentool angebunden ist und von Chris Sangwin im Jahr 2005 entwickelt wurde. 2014 implementierten Fred Neumann und Jesus Copado STACK als Plugin in ILIAS.

Im Vergleich zu üblichen Fragetypen bietet der STACK-Fragetyp durch Zufallsvariablen in der Aufgabenstellung eine große Vielfalt an Aufgabenvarianten. Durch die kontrollierbare Randomisierung können mehrere ähnliche Aufgaben erzeugt werden, die bezüglich des Schwierigkeitsgrads vergleichbar sind. Der Fragetyp STACK ist für die Übungsaufgaben (Training) und den Selbsttest besonders geeignet, da bei jedem Aufruf die Aufgabe mit veränderten Parametern erscheinen kann.

Ein weiterer Vorteil des STACK-Fragetyps für selbständiges Lernen liegt im qualifizierenden Feedback, das speziell an die Eingabe des Studierenden angepasst werden kann. STACK-gebundenes Feedback kann auf die Eingaben eingehen, die dann durch einen Rückmeldebaum abgebildet werden. Daher müssen im Vorfeld die Ziele der Aufgabe festgelegt und typische Fehler der Studierenden einbezogen werden. Die Schwierigkeit für den Entwickler besteht darin, den Rückmeldebaum übersichtlich zu halten und gleichzeitig die relevanten Eingabemöglichkeiten abzudecken. Die von STACK unterstützte Vielfalt der Fehleranalyse ermöglicht außerdem eine „faire“ Punktevergabe bei teilweise richtigen Aufgabenlösungen und Folgefehlern.

Zusammenfassend lässt sich sagen, dass der Fragetyp STACK neue Möglichkeiten bei der Entwicklung von Übungsaufgaben eröffnet, insbesondere in Bezug auf individuelles Feedback. Allerdings ist das Erstellen und Testen von STACK-Aufgaben mit hohem Zeitaufwand verbunden, der sich nur dann lohnt, wenn die Aufgaben wiederholt zum Einsatz kommen.

Alle Lernmaterialien, die im Rahmen des optes-Projekts entwickelt werden, sind Open Source und können von interessierten Hochschulen genutzt werden. Dies gilt auch für die Fragen vom Typ STACK: Nach der internen Qualitätskontrolle und dem Einsatz im optes-Vorkurs werden alle Übungsaufgaben auf der optes-Plattform zur Verfügung gestellt.

2 Erstellen einer STACK-Frage

2.1 Beispielaufgabe

Im folgenden Abschnitt wird die Erstellung einer STACK-Frage anhand einer Beispielaufgabe erklärt. Hierbei wird verdeutlicht, wie der STACK-Fragetyp mathematische Eigenschaften der studentischen Antwort prüfen kann. Ziel ist es, die folgende Aufgabe mit unendlich vielen Lösungsmöglichkeiten zu erstellen und die studentischen Eingaben auf ihre Richtigkeit zu prüfen:

Funktion mit drei vorgegebenen Nullstellen

2.1.1 Erste Schritte

Als erstes wird der „Frage erstellen“-Button oberhalb des Fragenpools angeklickt.

Daraufhin wird in der Auswahlliste „STACK Frage“ ausgewählt und „Erstellen“ geklickt.

Jetzt kann die Eingabe des Titels, des Autors und ggf. der Beschreibung erfolgen. Schließlich wird in das Feld „Frage“ die Aufgabenstellung geschrieben. Hierbei sind das Eingabefeld für die studentische Antwort „`[[input:ans1]]`“ und der Validieren-Button „`[[validation:ans1]]`“ bereits vorgegeben. Im Fragefeld sind Texteingaben, HTML-Code (via HTML-Button) und LaTeX-Code (durch „`$LaTeX-Code$`“ oder „`\(LaTeX-Code\)`“ gekennzeichnet) möglich. Auch die später definierten Maxima-Ausdrücke können in das Fragefeld geschrieben werden. Die Zufallsvariablen werden dabei in der Form „`@Zufallsvariable@`“ geschrieben.

Tip: Mit dem „TEX“-Button kann markierter LaTeX-Code in einer Vorschau gezeigt und verändert werden. Nach der Übernahme der Änderungen erscheint der Code in grün. Anschließend muss mit dem „Radiergummi“-Button die Formatierung zurückgesetzt werden.

Wichtig: Damit Änderungen nicht verloren gehen, muss regelmäßig „Speichern“ geklickt werden. Außerdem werden neue Eingabefelder, Knoten und Rückmeldebäume erst nach dem Speichern angezeigt.

Solange keine Musterlösung unter „ans1“ eingegeben wurde, erscheint eine Fehlermeldung, sobald die Aufgabe gespeichert wird. Außerdem müssen die im Fragetext verwendeten Zufallsvariablen („nst1“, „nst2“, „nst3“) unter „Fragen-Variablen“ definiert werden.

2.1.2 Fragen-Variablen und Aufgabenhinweis

Im Feld „Fragen-Variablen“ können CAS Variablen definiert und verändert werden. Diese Variablen sind in allen anderen Teilen der Aufgabe verfügbar. Eine besondere Rolle spielen die Zufallsvariablen. Es gibt verschiedene Möglichkeiten Zufallsvariablen zu erzeugen (siehe [1]):

- „rand(n)“ erzeugt eine Zufallszahl aus dem Intervall [0, n-1]
- „rand([a,b,...,z])“ wählt eine Zahl aus der Liste [a,b,...,z]
- „rand_with_step(a,b,n)“ gibt eine Zahl aus der Liste [a,a+n,a+2n,...,b-n,b] wieder
- „rand_with_prohib(a,b,[x])“ wählt eine Zahl z aus dem Intervall [a,b] mit $z \neq x$
- ...

Variablenzuweisungen werden in Maxima mit „:=“ vorgenommen. Funktionen werden durch „:=“ definiert und das Ende eines Kommandos mit „;“ gekennzeichnet. Kommentare können durch „/*Kommentar*/“ eingegeben werden.

Mit Hilfe des Aufgabenhinweises kann zwischen verschiedenen zufälligen Versionen einer Frage unterschieden werden. Daher sollten die aufgabenspezifischen Zufallsvariablen in den „Aufgabenhinweis“ geschrieben werden. Ist der Aufgabenhinweis leer und es kommen Zufallsvariablen in der Frage vor, so erscheint eine Fehlermeldung.

Unter Fragen-Variablen wird der folgende Code geschrieben:

```
nst1: rand_with_step(-7,7,1);
nst2: rand_with_prohib(-7,7,[nst1]);
nst3: rand_with_prohib(-7,7,[nst1,nst2]);
tans1: (x-nst1)*(x-nst2)*(x-nst3);
```

Der Aufgabenhinweis lautet:

$\backslash(x_1=\backslash)@nst1@, \backslash(x_2=\backslash)@nst2@, \backslash(x_3=\backslash)@nst3@$

Fragen-Variablen	<pre>nst1: rand_with_step(-7,7,1); nst2:rand_with_prohib(-7,7,[nst1]); nst3:rand_with_prohib(-7,7,[nst1, nst2]); tans1:(x-nst1)*(x-nst2)*(x-nst3);</pre>
Aufgabenhinweis	<pre>\(x_1=\backslash)@nst1@, \backslash(x_2=\backslash)@nst2@, \backslash(x_3=\backslash)@nst3@</pre>

Tip: Am Anfang der Seite befindet sich der „Info-Meldungen aktivieren/deaktivieren“-Button. Sind Info-Meldungen aktiviert, so werden Hilfestellungen zum Ausfüllen der Felder angezeigt.

2.1.3 Spezifisches Feedback und Test-Eingaben

Unter „Spezifisches Feedback“ wird standardmäßig das Feedback für jeden Rückmeldebäum angezeigt. Dieses Feedback kann in den Aufgabentext verschoben werden. Aufgaben mit mehreren Eingabefeldern, die unabhängig voneinander sind und verschiedene Rückmeldebäume haben, können so eine sinnvoll gegliederte Rückmeldung erhalten.

Wird das Feld „Zeige Fragenoptionen“ geöffnet, so können die Standardrückmeldungen für korrekte, teilweise korrekte und falsche Antworten geändert werden. Zusätzlich kann unter „Allgemeines Feedback“ eine Musterlösung angegeben werden, die unabhängig von der erreichten Punktzahl nach Bearbeitung einer Aufgabe erscheint.

Unter „Test-Eingaben“ befinden sich die Einstellungen zu den Eingabefeldern. Weitere Eingabefelder können hinzugefügt werden, indem „[[input:ans2]]“, „[[input:ans3]]“, etc. in den Fragetext geschrieben und anschließend „Speichern“ geklickt wird. Sind die „Validieren“-Buttons gewünscht, so müssen zusätzlich „[[validation:ans2]]“, „[[validation:ans3]]“, etc. hinter die Eingabefelder in der Fragestellung geschrieben werden. Wurde das Feld „ans1“ geöffnet, so kann unter „Musterlösung“ die gewünschte Antwort eingegeben werden. Nach dem Speichern erscheint jetzt keine Fehlermeldung mehr.

Tip: Statt „ans1“, „ans2“, etc. können beliebige Bezeichnungen gewählt werden. Dabei gilt es zu beachten, dass in allen Teilen der Aufgabe die Ausdrücke entsprechend ersetzt werden müssen (insbesondere bei der Validierung und den Rückmeldebäumen).

Tip: Unter „Verbotene Worte“ können Zeichenketten (mehrere Eingaben werden durch Kommata getrennt) eingegeben werden, die als Teilnehmer-Antwort nicht erlaubt sind.

TEST-EINGABEN

▼ Einstellungen der Eingabe ans1 *

Eingabetyp *	Algebraische Eingabe
Musterlösung *	ans1
Eingabebreite	15
Strikte Syntax	<input checked="" type="checkbox"/>
Sterne einfügen	<input type="checkbox"/>
Syntax-Hinweis	
Verbotene Worte	
Gleitkomma verbieten	<input checked="" type="checkbox"/>
Wörter zulassen	
Vollständige Kürzung verlangen	<input type="checkbox"/>
Antwort-Typ überprüfen	<input type="checkbox"/>
Verifizierung erforderlich	<input checked="" type="checkbox"/>
Validierung anzeigen	<input checked="" type="checkbox"/>
Zusätzliche Optionen	

2.1.4 Rückmeldebäume

Der wesentliche Teil einer STACK-Aufgabe besteht aus dem Rückmeldebaum. Hier wird die studentische Antwort auf ihre mathematischen Eigenschaften untersucht und die Richtigkeit der Eingabe überprüft. Die Antwortüberprüfung kann auf verschiedene Arten erfolgen: **AlgEquiv** überprüft beispielsweise, ob die Differenz von „SAns“ und „TAns“ die Zahl 0 ergibt. Eine Liste der verschiedenen Antwortüberprüfungen befindet sich im Anhang.

Da die Beispielaufgabe unendlich viele Lösungen besitzt, kann durch einen Vergleich mit der Musterlösung nicht jede richtige Antwortvariante erfasst werden. Stattdessen wird die studentische Antwort auf ihre mathematischen Eigenschaften geprüft.

Im ersten Knoten wertet Maxima die studentische Funktion an der ersten Nullstelle aus und prüft, ob das Ergebnis 0 ergibt. Das geschieht, indem unter „SAns“ der Ausdruck „ $ev(ans1, x=nst1)$ “ („ans1“ ist die studentische Funktion, „nst1“ bezeichnet die erste Nullstelle, die zuvor unter Fragen-Variablen definiert wurde) und unter „TAns“ die Zahl „0“ geschrieben wird und mittels der Antwortüberprüfung „**AlgEquiv**“ die algebraische Äquivalenz überprüft wird. In das Feld „Punkte wenn wahr“ wird „0.33“ eingegeben. In das Feld „Punkte wenn falsch“ wird die Zahl „0“ eingetragen.

Tip: Die Punktzahlen können bei mehreren Knoten addiert werden. Dabei muss beachtet werden, dass die richtige Lösung insgesamt einen Punkt erhält. Bei einer Gesamtpunktzahl zwischen 0 und 1 erscheint das Standard-Feedback „Ihre Antwort ist teilweise korrekt“ und bei 0 Punkten „Ihre Antwort ist nicht korrekt“. Wurde die maximale Punktzahl erreicht, so erscheint die Rückmeldung „Ihre Antwort ist korrekt“.

Mit Hilfe der Felder „Feedback wenn wahr“ (Antworthinweis „prt-0-T“) und „Feedback wenn falsch“ (Antworthinweis „prt-0-F“) kann knotenspezifisches Feedback gegeben werden. Unter „Feedback wenn falsch“ wird der Satz „Ihre Funktion nimmt für $x=@nst1@$ nicht den Wert 0 an.“ geschrieben. Nach Speichern der Aufgabe erscheinen die Antworthinweise „prt-0-T“ und „prt-0-F“ automatisch.

Unter dem Feld „Neuer Knoten“ können weitere Knoten erstellt werden. Nach Speichern des neuen Knotens kann der erste Knoten durch „Knoten wenn wahr/falsch“ auf den Zweiten verweisen.

Tip: Unter den Einstellungen des Rückmeldebaums befindet sich das Feld „Zeigen“. Wird dieses ausgeklappt, so kann mit Hilfe der „Feedback-Variablen“, die Studentische Eingabe zusammen mit den

Aufgabenvariablen manipuliert werden, bevor der Rückmeldebaum durchlaufen wird. Variablen, die hier definiert werden, können überall im Rückmeldebaum benutzt werden.

Antwortüberprüfung
 SAns *
 TAns *
 Test-Optionen
 Feedback unterdrücken
 Lösche Knoten

AlgEquiv
 ev(ans1,x=nst1)
 0
 Ja
 Lösche Knoten

Mod
 Punkte wenn wahr: + 0.33
 Abzüge wenn wahr: 0
 Knoten wenn wahr: 1
 Antworthinweis wenn wahr: prt-0-T
 Feedback wenn wahr: [Rich Text Editor]

Mod
 Punkte wenn falsch: - 0
 Abzüge wenn falsch: 0
 Knoten wenn falsch: 1
 Antworthinweis wenn falsch: prt-0-F
 Feedback wenn falsch: [Rich Text Editor]

Ihre Funktion nimmt für $\backslash(x=@nst1@)\backslash$ nicht den Wert $\backslash(0)\backslash$ an.

Tipp: Eine Übersicht der Maxima-Befehle ist unter [1] und [2] zu finden. Um die Funktionsweise der Befehle besser zu verstehen, bietet es sich an, diese in das Feedback des Rückmeldebaums zu schreiben und die Ausgabe beim Testen der Aufgabe zu überprüfen. Anschließend können die Befehle gelöscht und das gewünschte Feedback verfasst werden.

Als nächstes wird das Feld „neuer Knoten“ ausgewählt und die Felder analog zu „Knoten 0“ ausgefüllt (hierbei muss überall „nst1“ durch „nst2“ ersetzt werden). Nach Speichern kann unter „Knoten 0“ im Feld „Knoten wenn wahr/falsch“ jeweils „1“ ausgewählt und beide Knoten miteinander verknüpft werden.

Antwortüberprüfung
 SAns *
 TAns *
 Test-Optionen
 Feedback unterdrücken
 Lösche Knoten

AlgEquiv
 ev(ans1,x=nst2)
 0
 Ja
 Lösche Knoten

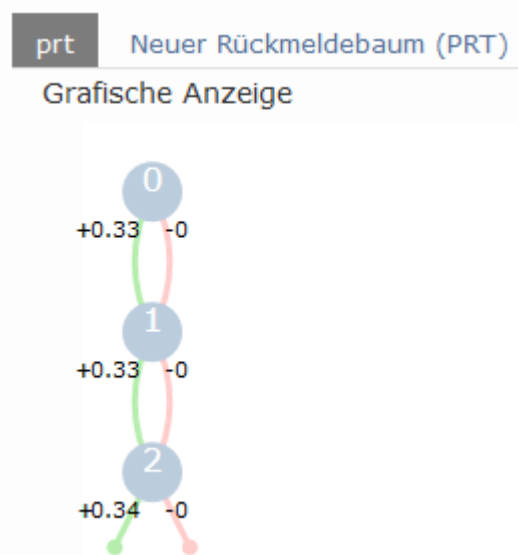
Mod
 Punkte wenn wahr: + 0.33
 Abzüge wenn wahr: 0
 Knoten wenn wahr: 2
 Antworthinweis wenn wahr: prt-1-T
 Feedback wenn wahr: [Rich Text Editor]

Mod
 Punkte wenn falsch: - 0
 Abzüge wenn falsch: 0
 Knoten wenn falsch: 2
 Antworthinweis wenn falsch: prt-1-F
 Feedback wenn falsch: [Rich Text Editor]

Ihre Funktion nimmt für $\backslash(x=@nst2@)\backslash$ nicht den Wert $\backslash(0)\backslash$ an.

Nach erneutem Speichern wird ein weiterer Knoten hinzugefügt und die Felder analog zu „Knoten 1“ ausgefüllt (hierbei muss überall „nst2“ durch „nst3“ ersetzt werden). Außerdem wird unter „Punkte wenn wahr“ die Zahl „0,34“ geschrieben, damit die volle Punktzahl (1 Punkt) erreicht werden kann. Nach dem Speichern wird in „Knoten 1“ unter „Knoten wenn wahr/falsch“ jeweils die Zahl „2“ ausgewählt und wieder gespeichert.

Der fertige Rückmeldebaum besteht aus drei Knoten mit unterschiedlicher Punktevergabe:



Tipp: Unter „Neuer Rückmeldebaum“ können weitere Rückmeldebäume erstellt werden. Nach Speichern erscheint „[[feedback:prt1]]“ in dem Feld „Spezifisches Feedback“ für den neuen Rückmeldebaum „prt1“.

2.1.5 Testphase

Besonders wichtig beim Erstellen einer STACK-Frage ist die Testphase. Hier wird untersucht, ob verschiedene Eingaben das gewünschte Feedback erhalten.


Tip: Es sollten verschiedene Darstellungsformen der Musterlösung geprüft werden. Es kann passieren, dass durch die Antwortüberprüfung äquivalente Ausdrücke in unterschiedlichen Zweigen enden.


Nach Speichern der letzten Änderungen wird das Feld „Vorschau“, das sich in der Zeile „Fragenpool für Tests“, „Seite bearbeiten“, „Eigenschaften“, etc. befindet, angeklickt. Anschließend wird die Frage mit Zufallsvariablen erzeugt. Die Eingabe der studentischen Antwort in die vorgegebene Lücke erfolgt analog zu den üblichen Eingaben in einen Taschenrechner über AsciiMath Code. Durch den optionalen „Validieren“-Button wird die Syntax der studentischen Antwort überprüft und es wird eine Vorschau gezeigt.

Funktion mit drei vorgegebenen Nullstellen

Geben Sie eine Funktion f mit den Nullstellen $x_1 = 7$, $x_2 = 4$ und $x_3 = -5$ an!

$f(x) =$





Mit Hilfe des „Validieren“-Buttons kann der Student seine Eingaben beliebig oft überprüfen, ohne dass seine Antwort ausgewertet wird. Die Validierungsoption prüft die Syntax der Eingabe. Erst wenn „Rückmeldung anfordern“ geklickt wurde, erfolgt die Überprüfung der Antwort mit Hilfe des Rückmeldebaums.

Geben Sie eine Funktion f mit den Nullstellen $x_1 = 7$, $x_2 = 4$ und $x_3 = -5$ an!

$f(x) =$

Ihre Antwort wurde wie folgt interpretiert:
 $17 \cdot (x - 7)^3 \cdot (x - 4) \cdot (x + 5)^6$

Ihre Antwort ist korrekt.

Geben Sie eine Funktion f mit den Nullstellen $x_1 = 7$, $x_2 = 4$ und $x_3 = -5$ an!

$$f(x) = (x-7) \cdot (|x+\frac{1}{2}| - \frac{9}{2})$$

Ihre Antwort wurde wie folgt interpretiert:

$$(x - 7) \cdot (|x + \frac{1}{2}| - \frac{9}{2})$$

Rückmeldung anfordern

Ihre Antwort ist korrekt.

Geben Sie eine Funktion f mit den Nullstellen $x_1 = 7$, $x_2 = 4$ und $x_3 = -5$ an!

$$f(x) = (x-7) \cdot (x-4)^2$$

Ihre Antwort wurde wie folgt interpretiert:

$$(x - 7) \cdot (x - 4)^2$$

Rückmeldung anfordern

Ihre Antwort ist teilweise korrekt.

Ihre Funktion nimmt für $x = -5$ nicht den Wert 0 an.

Geben Sie eine Funktion f mit den Nullstellen $x_1 = 7$, $x_2 = 4$ und $x_3 = -5$ an!

$$f(x) = (x-1) \cdot (x-2) \cdot (x-3)$$

Ihre Antwort wurde wie folgt interpretiert:

$$(x - 1) \cdot (x - 2) \cdot (x - 3)$$

Rückmeldung anfordern

Ihre Antwort ist nicht korrekt.

Ihre Funktion nimmt für $x = 7$ nicht den Wert 0 an.

Ihre Funktion nimmt für $x = 4$ nicht den Wert 0 an.

Ihre Funktion nimmt für $x = -5$ nicht den Wert 0 an.

2.2 Überarbeitung der Aufgabe

Als nächstes wird das Feedback der Beispielaufgabe überarbeitet. Hierzu wird die Aufgabe „Funktion mit drei vorgegebenen Nullstellen“ ausgewählt und kopiert.

Nach Klick auf den „Einfügen“-Button erscheint eine Kopie der Frage „Funktion mit drei vorgegebenen Nullstellen (2)“, die anschließend überarbeitet werden soll.

Mit dem Befehl „plot()“ können Funktionsgraphen erstellt werden. In die Felder „Feedback wenn wahr/falsch“ von „Knoten 2“ wird jeweils der „HTML“-Button angeklickt und folgender Text in das neue Fenster geschrieben:

`<p>Das Schaubild zeigt die Musterlösung und Ihre Antwort:</p>`

`<p>@plot([tans1, ans1], [x,-10,10],[y,-50,50])@</p>`

Nachdem „Update“ geklickt wurde erscheint der Text „Musterlösung“ in blauer und „Ihre Antwort“ in roter Schrift.

In der Vorschau können nun verschiedene Eingaben und ihr Feedback getestet werden.

Funktion mit drei vorgegebenen Nullstellen (2)

Geben Sie eine Funktion f mit den Nullstellen $x_1 = -1$, $x_2 = 1$ und $x_3 = -5$ an!

$f(x) =$

Ihre Antwort wurde wie folgt interpretiert:

$(x - 5) \cdot (x + 1) \cdot (x - 7)$

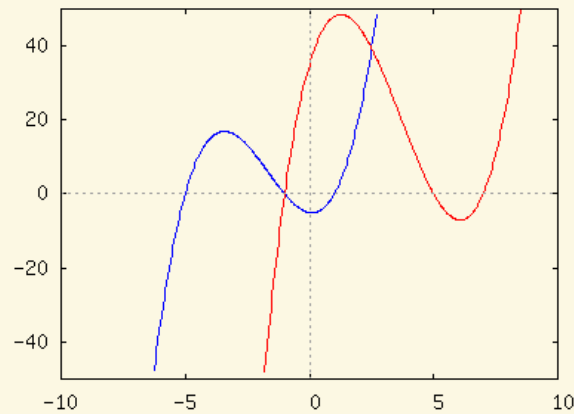
[Rückmeldung anfordern](#)

Ihre Antwort ist teilweise korrekt.

Ihre Funktion nimmt für $x = 1$ nicht den Wert 0 an.

Ihre Funktion nimmt für $x = -5$ nicht den Wert 0 an.

Das Schaubild zeigt die **Musterlösung** und **Ihre Antwort**:



Viel Spaß beim Ausprobieren!

3 Anhang

3.1 Antwortüberprüfung

Für die Antwortüberprüfung eines Knotens gibt es verschiedene Möglichkeiten. Unter [1] befindet sich eine Liste der Antwortüberprüfungen mit Erläuterungen. Hier wird ein kleiner Überblick über die geläufigsten Überprüfungsverfahren gegeben:

AlgEquiv	Prüft, ob die studentische Antwort mit der geforderten Antwort algebraisch übereinstimmt, d.h. ob „ $\text{simplify}(\text{SAns}-\text{TAns})=0$ “ gilt.
EqualComAss	Prüft, ob „SAns“ und „TAns“ übereinstimmen, dabei werden Kommutativ- und Assoziativgesetz berücksichtigt nicht aber das Distributivgesetz, d.h. „ $a+b=b+a$ “ aber „ $x+x \neq 2x$ “.
CasEqual	Prüft, ob „SAns“ und „TAns“ äquivalent sind.
SingleFrac	Prüft, ob „SAns=TAns“ gilt und ob „SAns“ als ein Bruch geschrieben wurde.

3.2 Maxima-Befehle

Unter [1] und [2] befindet sich eine Übersicht der häufig genutzten Maxima-Befehlen. Ein paar wichtige Befehle werden im Folgenden erklärt:

cardinality(a)	Gibt die Mächtigkeit (Kardinalität) einer Menge „a“ zurück.
denom(a)	Gibt den Nenner des Ausdrucks „a“ zurück, wenn „a“ ein Quotient ist. Ist „a“ kein Quotient, so wird „a“ zurückgegeben.
expand(a)	Produkte und Potenzen von Summen werden ausmultipliziert. Die Nenner von rationalen Ausdrücken, die Summen sind, werden in ihre Terme aufgespalten. Produkte werden in Summen herein multipliziert.
ev(a,x1,...,xn)	Wertet den Ausdruck „a“ in einer Umgebung aus, die durch die Argumente „x1,...,xn“ spezifiziert wird.
factor(a)	Faktoriert den Ausdruck „a“, der eine beliebige Zahl an Variablen oder Funktionen enthalten kann, in irreduzible Faktoren über die ganzen Zahlen.
integrate(a,x)	Sucht die symbolische Lösung des Integrals für den Ausdruck „a“ und der Integrationsvariablen „x“; der Befehl löst das unbestimmte Integral.
integrate(a,x,m,n)	Sucht die Lösung des bestimmten Integrals von „a“ nach „x“ mit den Integrationsgrenzen „m“ und „n“.
intersection(a,b)	Gibt die Schnittmenge der Mengen „a“ und „b“ zurück.
num(a)	Gibt den Zähler des Ausdrucks „a“ zurück, wenn „a“ ein Quotient ist. Ist „a“ kein Quotient, wird „a“ zurückgegeben.
setdifference(a,b)	Entfernt die Elemente der Menge „b“ aus der Menge „a“ und gibt das Ergebnis zurück.
setify(a)	Konstruiert eine Menge aus den Elementen der Liste „a“. Doppelte Elemente der Liste „a“ werden entfernt.
solve(a,b)	löst eine algebraische Gleichung „a“ nach der Variablen „x“ auf. Ist „a“ keine Gleichung, so wird „a=0“ angenommen.

4 Quellen

- [1] <http://stack.bham.ac.uk/moodle/question/type/stack/doc/doc.php/>
- [2] <http://maxima.sourceforge.net>
- [3] Derr, K., Hübl, R., Mechelke-Schwede, E., Podgayetskaya, T., Weigel, M. (2017). Inhaltliche und technische Aspekte des automatisierten Feedbacks. Einsatz des Fragetyps STACK im formativen eAssessment. Institut für Mathematik der Universität Potsdam (Hrsg.), Beiträge zum Mathematikunterricht 2017. Münster: WTM-Verlag