

The background of the slide features a large, faint, circular seal of the University of Cologne. The seal depicts a religious scene with the Virgin Mary seated on the right, holding the Christ Child. Two kings are kneeling before them, offering gifts. Two other kings stand on the left, one holding a scepter. The seal is surrounded by a Latin inscription: "SIGILLUM UNIVERSITATIS COLONIENSIS".

Für Autor*innen

Einführung STACK 3

Die Verwendung der STACK-Frage in ILIAS

STACK-Team des CCE der Universität zu Köln

26.5.2022

Dieses Dokument ist unter der Creative-Commons-Lizenz [“Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 nicht portiert”](#) lizenziert. Dieses Werk darf unter folgenden Bedingungen verbreitet und neu zusammengestellt werden: Namensnennung und Weitergabe unter gleichen Bedingungen. CC BY-SA

Empfohlene Zitierweise:

STACK-Team des CCE der Universität zu Köln: Einführung STACK 3.0. Die Verwendung der STACK-Frage in ILIAS, Köln 2022.

Autor*innen dieses Dokumentes:

Eva Mix, Mario Josupeit, Tjark Eilts, Karoline Lange, Annemarie Sich

Inhaltsverzeichnis

1	Einleitung	4
2	Beitritt zum Kurs	6
3	STACK aus Sicht von Studierenden	7
4	Bestandteile der STACK-Frage	10
4.1	Fragentext	10
4.2	Eingabefelder	10
4.3	Rückmeldebäume	11
4.4	Punktevergabe	12
5	Erstellung einer STACK-Frage	13
5.1	Flowchart	14
5.2	Kurzanleitung	15
5.3	Erweiterte Anleitung	23
5.3.1	Aufgabenweite Optionen	23
5.3.2	Einstellungen der Eingabefelder	25
5.3.3	Rückmeldebäume (PRTS)	27
5.3.4	Punktevergabe	32
5.3.5	Eingabetypen	35
5.3.6	Antwortüberprüfungen	48
5.3.7	Zufallswerte	55
5.3.8	Maxima	60
5.3.9	MathJax	65
5.3.10	HTML	67
6	Verwendung von STACK in Tests	70
7	Weiterführende Links	72
8	Übungsmaterial	73
8.1	Übungsaufgaben	74
8.2	Schritt-für-Schritt-Hilfestellungen	79
9	Abkürzungsverzeichnis	92
10	Kontakt	93

1 Einleitung

STACK steht für **S**ystem for **T**eaching and **A**ssessment using **C**omputer algebra **K**ernel und ist ein Fragentyp in ILIAS, der detaillierte Bewertungen und Rückmeldungen mit Hilfe von Rückmeldebäumen ermöglicht. Vorteile der STACK-Frage sind die hohe Anpassungsfähigkeit an die Wünsche der Nutzer*innen und die Wiederverwendung von Aufgaben, die durch Zufallsvariablen vielseitig einsetzbar sind. Zudem bieten die verschiedenen Eingabetypen und die Einbindung von HTML und LaTeX-Code ein breites Angebot an Funktionen und Gestaltungsmöglichkeiten. Die Möglichkeit, mehrere Teilaufgaben innerhalb einer STACK-Frage anzuzeigen und damit verbundene spezifische Feedbacks, machen die STACK-Frage attraktiv für Übungsaufgaben und Tests.

Dank dem angeschlossenen *Computeralgebrasystem* (CAS) Maxima ist es möglich, in STACK-Fragen auf algebraische Äquivalenz zu prüfen, wodurch sie sich insbesondere für naturwissenschaftliche Fragen sehr gut eignen. Aufgrund ihrer Flexibilität findet die STACK-Frage jedoch auch immer mehr Verwendung in nicht-mathematischen Bereichen.

Die Verwendung unterschiedlicher Eingabetypen, wie algebraische Eingaben, Multiple- und Single-Choice Aufgaben, der Eingabe von Zeichenketten, die Überprüfung physikalischer Einheiten und Freitextfelder, ermöglichen innerhalb von STACK eine individuelle und problemangepasste Aufgabenstellung für mathematische und nicht-mathematische Fragestellungen. Unterschiedliche Teilaufgaben können in der Rückmeldung und Bepunktung miteinander verknüpft werden. Daraus ergibt sich die Möglichkeit, Folgefehler und systematische Fehler zu überprüfen und zielgerichtete Hinweise zu geben. Zufallszahlen bieten für mathematische Fragestellungen viele Möglichkeiten auf einfache Weise abwechslungsreiche Aufgaben zu erstellen. Doch auch für nicht-mathematische Fragestellungen bieten die Zufallswerte viele kreative Anwendungsmöglichkeiten. So kann damit aus einer Liste von Wörtern oder Sätzen zufällig ein Eintrag ausgewählt und verwendet werden. Es können also nicht nur Zahlen, sondern auch andere Elemente zufällig ausgegeben werden.

Die STACK-Frage wurde ursprünglich für die Lernplattform moodle entwickelt, wofür ausführliche Dokumentationen über die Verwendung vorhanden sind. Dieses Dokument beschäftigt sich als erste Dokumentation ausführlich mit der Verwendung von STACK für die Lernplattform in ILIAS. Die Dokumentationen für moodle sind gelegentlich trotzdem hilfreich, da manche Features analog funktionieren.

Das Dokument startet mit einer Darstellung der Bestandteile der STACK-Frage, um sich innerhalb dieser orientieren zu können. Danach geht es zur Erstellung der STACK-Frage. Hier hilft zuerst die Flowchart durch eine visuelle Unterstützung. Anschließend folgt eine Kurzanleitung, um ohne viel Einarbeitung die Erstellung erster, einfacher STACK-Fragen zu

ermöglichen. Im Kapitel „Lange Anleitung“ sind alle wichtigen Einstellungen und Anwendungen im Detail erklärt. Dazu gehören die vielen Eingabetypen und Antwortüberprüfung (z.B. Multiple-Choice, Gleichungen, etc.), die Verwendung von Maxima, Html und Zufallswerte. Anschließend beschäftigt sich das 5. Kapitel mit der Einbindung der Fragen in ILIAS-Tests und Übungen. Weiterführende Links sind für die Personen gedacht, die sich noch mehr mit der STACK-Frage auseinandersetzen möchten und abschließend können die erworbenen Kompetenzen mithilfe von Übungsmaterial, bestehend auch Übungsaufgaben und -lösungen, angewendet werden.

Aktuell gibt es an der Universität zu Köln die STACK-Frage unter:



<https://www.selfassessment.uni-koeln.de/>

<https://www.ilias.uni-koeln.de/>

Einen Showroom mit Einsatzmöglichkeiten für die STACK-Frage finden Sie unter:



[https://www.selfassessment.uni-](https://www.selfassessment.uni-koeln.de/goto.php?target=cat_4968&client_id=uzk_selfea)

[koeln.de/goto.php?target=cat_4968&client_id=uzk_selfea](https://www.selfassessment.uni-koeln.de/goto.php?target=cat_4968&client_id=uzk_selfea)

Falls Sie Interesse an einer STACK-Schulung haben, schreiben Sie uns unter:



[**ilias-stack@uni-koeln.de**](mailto:ilias-stack@uni-koeln.de)

2 Beitritt zum Kurs

Auf ILIAS der Uni Köln finden Sie einen Kurs, der wichtige Informationen zur STACK-Frage zusammenträgt. In diesem Kurs können Sie die im Dokument beschriebenen Beispielfragen anschauen und eigene Fragen schreiben.

Link zu dem Kurs:



https://www.selfassessment.uni-koeln.de/goto.php?target=crs_1927&client_id=uzk_selfea

Klicken Sie auf den Link und melden Sie sich mit Ihrem Mitarbeiteraccount an. Dann öffnet sich der entsprechende Kurs. Sie können beitreten mit dem **Passwort** STACK.

Alternativ können Sie den Kurs über das Magazin finden:

Magazin → CompetenceCenter E-Learning CCE → Innovative Assessments mit dem “STACK-Fragetyp“ → Einstieg STACK 3

In dem Einführungskurs haben Sie die Möglichkeit, selbst Fragen zu schreiben und auszuprobieren, was mit STACK möglich ist. Im Abschnitt **Sandkasten zum Ausprobieren** finden Sie den Ordner **Fragenpools zum Ausprobieren**, in dem Sie einem eigenen Fragenpool anlegen können, um die STACK-Frage auszuprobieren. Benennen Sie diesen Fragenpool bitte mit Ihrem Namen.



https://www.selfassessment.uni-koeln.de/goto.php?target=fold6001&client_id=uzk_selfea

Sie sind kein*e Mitarbeiter*in der UzK? Erstellen Sie sich bitte ein Benutzerkonto unter



<https://www.selfassessment.uni-koeln.de>

und schreiben Sie uns eine E-Mail mit Ihrem Benutzernamen, dann werden wir Ihnen den Zugriff manuell ermöglichen.

Sie erreichen uns unter: ilias-stack@uni-koeln.de

3 STACK aus Sicht von Studierenden

STACK-Fragen werden innerhalb von Test verwendet, um sie Studierenden zugänglich zu machen. Eine STACK-Frage besteht hierbei aus Titel, Fragentext, einem oder mehreren Eingabefeldern sowie Buttons zur Überprüfung der Eingabe und zum Anfordern der Rückmeldung. Die Rückmeldung wird standardmäßig unter der Frage angezeigt, kann jedoch auch in den Fragentext verschoben werden, was bei Aufgaben mit mehreren Teilaufgaben besonders hilfreich ist. Das Anzeigen der bestmöglichen Lösung kann in den Testeinstellungen an bzw. ausgestellt werden.

The screenshot displays a STACK question titled "Erste einfache Rechnung". The question text is "Rechnen Sie $3 + 4 * 2$ aus." followed by an input field containing "11" and a checkmark button. Below the input field, a feedback message states: "11 wurde wie folgt interpretiert: 11". A button labeled "Rückmeldung anfordern" is visible. The feedback section, titled "Rückmeldung", shows a green bar with a checkmark and the text "Sie haben die Aufgabe richtig ausgerechnet." Below this, it says "Sie haben 1 von 1 möglichen Punkten erreicht." and "Die bestmögliche Lösung lautet:". A second, identical question box is shown below the feedback, also with the answer "11".

Eingabe von mathematischen Ausdrücken

Um Studierenden hilfreiche Tipps bei der Eingabe von Antworten zu geben und die eigenen Fragen testen zu können, ist es wichtig zu wissen, wie mathematische Ausdrücke in STACK eingegeben werden. Die häufigsten mathematischen Ausdrücke sind in der folgenden Tabelle aufgelistet:

	Mathematischer Ausdruck	Eingabe in STACK
Dezimalpunkt	$\frac{1}{2}$	0.5 <i>Hinweis: Nicht 0,5!</i>

Grundrechenarten	$+, -, *, /$ ax $3a$	$a*x$ $3*a$ <i>Hinweis:</i> Bei Multiplikation muss * zwischen einzelnen Faktoren stehen.
Exponenten	a^2	a^2
Klammern	$\frac{a-b}{c+b}$ $a^{(b+c)}$	$(a-b)/(c+d)$ $a^{(b+c)}$ <i>Hinweis:</i> Damit Ausdrücke richtig interpretiert werden, müssen Klammern sinnvoll eingefügt werden. Eckige Klammern [...] dürfen nicht verwendet werden.
Funktionen	Exponentialfunktion e^x Logarithmus $\ln(x)$ $\log_a(x)$ Wurzelfunktion \sqrt{x} $\sqrt[a]{x^b}$	$\exp(x)$ oder $\%e^x$ $\log(x)$ $\log(x)/\log(a)$ $x^{(1/2)}$ $x^{(b/a)}$
Matrizen	Potenz M^k Matrixprodukt $M * N$	M^k $M.N$

Nutzer*innen der STACK-Frage müssen zur Eingabe die Ausdrücke der rechten Spalte verwenden. Es kann sinnvoll sein, die Nutzer*innen z.B. durch [Syntax-Hinweise](#) entsprechend darauf aufmerksam zu machen.



Als Hilfestellung für STACK aus Nutzer*innen-Sicht empfehlen wir den Abschnitt „Einstieg für Studierende“ im Kurs [Einstieg STACK 3](#).

- Einstellungen der Eingabe ans1 *
- Einstellungen der Eingabe ans2 *
- Einstellungen der Eingabe ans3 *
- Einstellungen der Eingabe ans4 *

4.3 Rückmeldebäume

Die Rückmeldebäume werden für die Auswertung der Teilaufgaben verwendet. Sie ermöglichen das individuelle Feedback und bestimmen die Bepunktung abhängig von der Antwort der Studierenden. Rückmeldebäume können sehr einfach aufgebaut sein und lediglich abprüfen, ob die Antwort vollständig richtig oder falsch ist. Sie können jedoch auch sehr komplex werden und auf zuvor antizipierte häufige Fehler prüfen. Es ist möglich, mehrere Teilaufgaben gemeinsam in einem gemeinsamen Rückmeldebaum auszuwerten und eine gemeinsame Rückmeldung zu geben. Es ist jedoch auch möglich, für jede Teilaufgabe einen eigenen Rückmeldebaum zu erstellen und so ein separates Feedback zu geben. In der grafischen Anzeige kann betrachtet werden, wie der Rückmeldebaum aufgebaut ist und an welcher Stelle wie viele Punkte gegeben werden. Neben dem Rückmeldetext kann in dem Rückmeldebaum auch der Feedback-Style eingestellt werden. Dieser bewirkt, dass das Feedback farblich hinterlegt wird und so für die Studierenden leicht ersichtlich ist, ob sie die Frage z.B. richtig oder falsch beantwortet haben.

Beispiel für einen einfachen Rückmeldebaum:

prt1

prt2

prt3

prt4

Neuer Rückmeldebaum (PRT)

Grafische Anzeige

1

0

Rückmeldebaum Name *

prt1

Erster Knoten

1

Rückmeldebaum kopieren

Rückmeldebaum kopieren

► Einstellungen

1 Neuer Knoten

Antwortüberprüfung

AlgEquiv

Eine Antwortüberprüfung ist ein Test um zwei Ausdrücke dahingehend zu vergleichen, ob sie bestimmte mathematische Eigenschaften erfüllen.

SAns *

ans1

Dies ist das erste Argument der Antwortüberprüfungsfunktion. In asymmetrischen Tests wird dies als die Studierenden-Antwort angesehen, obwohl es jeder gültige CAS Ausdruck sein könnte. Es können darin auch Variablen aus der Aufgabe oder dem Feedback benutzt werden.

Inhalt: CAS-Ausdruck

TAns *

a2

Beispiel für einen komplexen Rückmeldebaum:

prt1 prt2 **prt3** prt4 prt5 prt6 Neuer Rückmeldebaum (PRT)

Grafische Anzeige

Rückmeldebaum Name *

prt3

Erster Knoten

0

Rückmeldebaum kopieren

Rückmeldebaum kopieren

► Einstellungen

0 1 2 3 4 5 6 7 8 Neuer Knoten

Antwortüberprüfung

AlgEquiv

Eine Antwortüberprüfung ist ein Test um zwei Ausdrücke dahingehend zu vergleichen, ob sie bestimmte mathematische Eigenschaften erfüllen.

SAns *

ans2

Dies ist das erste Argument der Antwortüberprüfungsfunktion. In asymmetrischen Tests wird dies als die Studierenden-Antwort angesehen, obwohl es jeder gültige CAS Ausdruck sein könnte. Es können darin auch Variablen aus der Aufgabe oder dem Feedback benutzt werden.

Inhalt: CAS-Ausdruck

TAns *

diff(ans1,x)

Dies ist das zweite Argument der Antwortüberprüfungsfunktion. In asymmetrischen Tests wird dies als die Dozierenden-Antwort angesehen, obwohl es jeder gültige CAS Ausdruck sein könnte. Es können darin auch Variablen aus der Aufgabe oder dem Feedback benutzt werden.

Inhalt: CAS-Ausdruck

Test-Optionen

Dieses Feld erlaubt Antwortüberprüfung eine Option zu verwenden, z.B. eine Variable oder eine bestimmte numerische Präzision.

Inhalt: CAS-Ausdruck

4.4 Punktevergabe

Neben dem Feedbacktext ist es möglich, die Bepunktung individuell anzupassen. Dies geschieht zum einen in den Rückmeldebäumen. Dort wird bestimmt, wie viel Prozent der Punkte für den Rückmeldebaum entlang eines Pfades vergeben werden. Zudem können die Rückmeldebäume untereinander gewichtet und die Gesamtpunktzahl für die gesamte STACK-Frage zentral angepasst werden. Dies ermöglicht es, die Bepunktung flexibel anzupassen. Eine Übersicht ermöglicht es, die möglichen Punkte auf einen Blick zu sehen.

◀ Fragenpool für Tests Seite bearbeiten Vorschau **Eigenschaften** Rückmeldung Statistik

Frage bearbeiten Bewertung Vorbereitete Zufallswerte Testfälle Fragen aus MoodleXML importieren Export nach MoodleXML

BEWERTUNGS-STATUS

Aktuelle Punktzahl 6

Dies sind die aktuell maximal erreichbaren Punkte für diese Frage.

Neue Punktzahl

Hier können sie einen neuen Wert eingeben, um die Berechnung aus den Teilpunkten zu vergleichen oder eine neue maximale Punktzahl festzulegen.

Diese Seite zeigt die Fragenbewertung in Punkten. Für jeden Knoten eines Ergebnisbaums werden die Punkte des positiven Antworttests in grün und die des negativen Antworttests in rot angezeigt. Für jeden Ergebnisbaum zeigen die maximal erreichbaren Punkte seine Gewichtung innerhalb der Frage

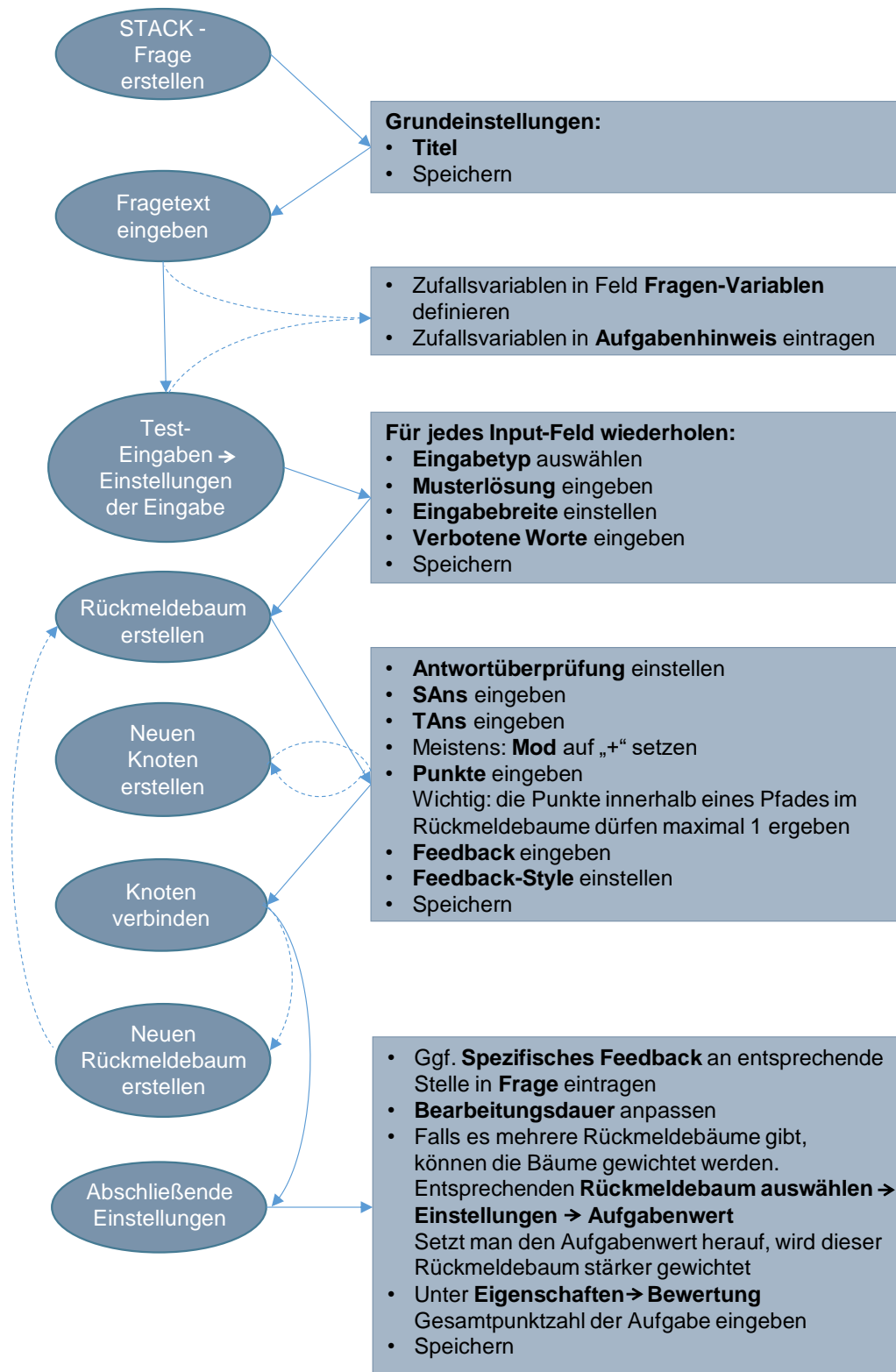
Zeige Vergleich Punktzahl speichern

Name des Auswertungsbaums: prt1	Maximale Punktzahl für diesen Auswertungsbaum: 2	
Knotenname: 1	=2	=0
Knotenname: 2	+0	+0
Name des Auswertungsbaums: prt2	Maximale Punktzahl für diesen Auswertungsbaum: 4	
Knotenname: 1	=4	=0

5 Erstellung einer STACK-Frage

Auf der folgenden Seite finden Sie ein Flowchart, auf dem die Erstellung einer STACK-Frage visualisiert ist. Es bietet sich an, dieses bei der Erstellung zur Hand zu nehmen und die einzelnen Schritte anhand des Flowchart durchzugehen. Anschließend finden Sie die Kurzanleitung für das schnelle Erstellen einfacher STACK-Fragen und anschließend eine detaillierte Erklärung zu allen wichtigen, weiterführenden Einstellungen. Wir empfehlen, das Kapitel [Erweiterte Anleitung](#) als Nachschlagewerk zu benutzen, um einzelne Einstellungen ausgehend von der Kurzanleitung zu vertiefen.

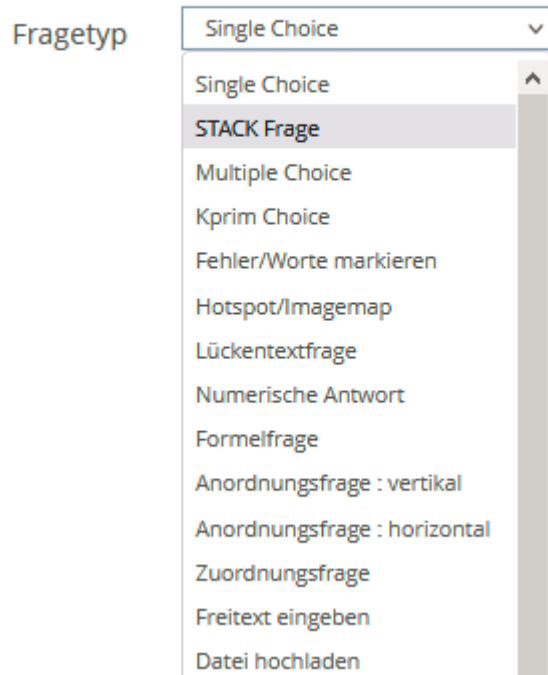
STACK-Frage erstellen



5.2 Kurzanleitung

Eine STACK-Frage kann direkt in einem Test erstellt werden oder erst in einem Fragenpool erstellt und anschließend in einen Test eingefügt werden. Wählen Sie hierfür im Drop-Down-Menü **Fragetyp** die **STACK-Frage** aus und klicken Sie anschließend auf **Erstellen**. Sie gelangen anschließend auf die Hauptansicht der Frageneinstellungen.

FRAGE ERSTELLEN



Hinweis: Eine STACK-Frage wird in mehreren Schritten erstellt. Um keine Eingaben zu verlieren, sollte sie zwischen jedem Schritt gespeichert werden.

Schritt 1: Titel und Autor*in

Zuerst wird der Titel der Frage eingetragen. Das Feld **Autor** ist bereits vorausgefüllt, kann jedoch bearbeitet werden, um beispielsweise mehrere Autor*innen einzutragen.

FRAGE BEARBEITEN


Titel *	<input type="text" value="Untitel STACK Frage"/>
Autor *	<input type="text" value="Karoline Lange"/>
Beschreibung	<input type="text"/>
Frage *	<div><div>B <i>I</i> <u>U</u> ABC Absatz </div><div>[[input:ans1]] [[validation:ans1]]</div><div>Pfad: p</div></div>

Schritt 2: Fragentext

Anschließend wird die Aufgabenstellung formuliert. Tragen Sie dazu den gewünschten Text in das Feld **Frage** ein. In diesem Feld ist es zudem möglich, mit LaTeX mathematische Ausdrücke einzutragen. Verwenden Sie hierzu den **TEX**-Button oder $\backslash()$ um den Mathmode aufzurufen. Weitere Informationen hierzu finden Sie im Abschnitt [MathJax](#). Der Text kann zudem mit Hilfe von HTML gestaltet werden. Weitere Informationen hierzu finden Sie im Abschnitt [HTML](#).

Titel *	<input type="text" value="Erste einfache Rechnung"/>
Autor *	<input type="text" value="Karoline Lange"/>
Beschreibung	<input type="text"/>
Frage *	<div><div>B <i>I</i> <u>U</u> ABC Absatz </div><div> </div><div></div><div>Rechnen Sie $\backslash(3+4*2\backslash)$ aus. [[input:ans1]] [[validation:ans1]]</div></div>

Der oben eingegebene Frage-Text sieht für Nutzer*innen so aus:



Erste einfache Rechnung

Rechnen Sie $3 + 4 * 2$ aus.

Rückmeldung anfordern

Die Platzhalter für das Eingabefeld `[[input:ans1]]` und den Button zur Überprüfung der Eingabe `[[validation:ans1]]` werden automatisch beim Erstellen der Frage eingetragen. Zusätzliche Platzhalter für weitere Eingabefelder können nach Bedarf manuell hinzugefügt werden. Kopieren Sie hierzu einen bestehenden Platzhalter und ändern Sie die Bezeichnung bspw. zu `[[input:ans2]]`. Es ist wichtig, dass eine Eingabefelder eine eigene Bezeichnung haben.

Hinweis: Die STACK-Frage muss gespeichert werden, nachdem ein neues Eingabefeld in den Fragetext hinzugefügt wurde. Erst dann können Einstellungen für die Eingabefelder vorgenommen werden.

Schritt 3: Zufallsvariablen (optional)

Optional können Zufallsvariablen in die Aufgabenstellung integriert werden, um Abschreiben zu erschweren und das mehrmalige Bearbeiten von Aufgaben zu Übungszwecken zu ermöglichen. Diese Zufallsvariablen müssen hierzu zunächst im Feld **Fragen-Variablen** definiert werden.

Variablen werden in der Form *Variablenname:Zufallsfunktion* definiert. Es stehen verschiedene Funktionen zur Definition von Zufallsvariablen zur Verfügung. Die meistverwendeten sind:

- $rand(n)$
 - n Ganzzahl: die Zufallszahl ist eine Ganzzahl zwischen 0 und $n-1$.
 - n Dezimalzahl: die Zufallszahl ist eine Dezimalzahl zwischen 0 und n .
- $rand_with_step(a,b,c)$ für eine Zufallszahl zwischen a und b mit der Schrittweite c

- `rand_with_prohib(a,b[c,d,e,...])` für eine ganzzahlige Zufallszahl zwischen a und b ohne die Zahlen aus der Liste [c,d,e,...], beispielsweise ist `rand_with_prohib(1,6[3,4,])` eine zufällige Zahl aus der Liste [1,2,5,6]

In diesem Beispiel wird der Variable `Var1` eine zufällige Ganzzahl zwischen 0 und 14 zugeordnet.

Fragen-Variablen

```
Var1:rand(15)
```

Dieses Feld erlaubt es, CAS-Variablen zu definieren und zu verändern, z.B. um Zufallsversionen zu ermöglichen. Diese Variablen sind in allen anderen Teilen der Aufgabe verfügbar.

Inhalt: CAS-Ausdruck

Im Text der Aufgabenstellung wird die Zufallsvariable mit Hilfe von Klammern eingefügt. In diesem Beispiel wird die Variable `Var1` mit `{@Var1@}` an die entsprechende Stelle eingefügt.

Wird eine Zufallsvariable definiert, muss immer auch ein **Aufgabenhinweis** angelegt werden. Dieser dokumentiert, welche Zufallswerte in einer Aufgabe angezeigt wurden. Es ist deswegen wichtig, dass der Aufgabenhinweis aussagekräftig ist. Alle verwendeten Zufallsvariablen müssen im Fragenhinweis vorkommen. Ein Beispiel für einen guten Aufgabenhinweis für diese Frage ist:

Aufgabenhinweis

```
Var1 ist {@Var1@}. Die Frage lautet somit  $3+{@Var1@} \cdot 2$ 
```

Mit dem Aufgabenhinweis soll zwischen den verschiedenen zufälligen Versionen einer Frage unterschieden werden. Zwei Fragen sind gleich, genau dann wenn die Aufgabenhinweise gleich sind. Für die spätere Analyse ist es hilfreich sinnvolle Antworthinweise zu erstellen.

Inhalt: CAS-Text

Schritt 4: Eingabetyp und Musterlösung

Nach dem Speichern des Fragentextes erscheint im Abschnitt **Test-Eingaben** → **Einstellungen der Eingabe ans1(2,3,...)** für jedes Eingabefeld die Möglichkeit, Einstellungen zu treffen. Für jedes Eingabefeld muss ein **Eingabetyp** und eine **Musterlösung** eingetragen werden.

TEST-EINGABEN

► Einstellungen der Eingabe ans1 *

Der Eingabetyp bestimmt, welche Form die Eingabe der Studierenden haben soll, damit STACK die Eingabe entsprechend auswerten kann. Der Eingabetyp ist standardmäßig die **Algebraische Eingabe**. Für viele STACK-Fragen ist dies der richtige Eingabetyp. Darüber

hinaus gibt es noch viele andere nützliche Eingabetypen, die je nach Bedarf genutzt werden können. Mehr Informationen darüber finden Sie in dem Abschnitt [Eingabetypen](#).

Die **Musterlösung** wird der*dem Nutzer*in nach Beantwortung der Frage im Abschnitt „Die bestmögliche Lösung lautet“ angezeigt. Ob dieser Abschnitt angezeigt wird oder nicht wird in den Einstellungen des Tests eingestellt.

Hinweis: Die **Musterlösung** dient lediglich der Darstellung und ist nicht für die Überprüfung der Eingabe verantwortlich. Die Überprüfung findet im Rückmeldebaum statt.

▼ Einstellungen der Eingabe ans1 *

Eingabetyp *

Algebraische Eingabe ▼

Typ des Eingabeelements, z.B. Formularfeld, Wahr/Falsch, mehrzeiliges Textfeld.

Musterlösung *

11 oder 3+Var1*2

Der Dozent muss eine Musterlösung für jedes Eingabefeld angeben. Dies muss eine gültige Maxima-Zeichenkette sein. Sie kann Variablen aus dem Aufgabentext enthalten.

Inhalt: CAS-Ausdruck

Schritt 5: Rückmeldebaum

Wenn der Fragetext erstellt und die Einstellungen für die Eingabefelder getroffen sind, muss ein Rückmeldebaum unter **Rückmeldebäume (PRTS)** angelegt werden. Dieser Rückmeldebaum entscheidet darüber, welche Antwort als richtig gewertet wird und wie viele Punkte vergeben werden. Hierzu wird im ersten Knoten unter **SAns** der Platzhalter des Eingabefeldes eingetragen, z.B. *ans1*(2,3...). Dieser Platzhalter repräsentiert die Eingabe der Nutzer*innen. Unter **TAns** wird die Lösung eingetragen, die mit der Eingabe verglichen werden soll. Dies ist im einfachsten Fall die richtige Antwort.

1 Neuer Knoten

Antwortüberprüfung

AlgEquiv ▼

Eine Antwortüberprüfung ist ein Test um zwei Ausdrücke dahingehend zu vergleichen, ob sie bestimmte mathematische Eigenschaften erfüllen.

SAns *

ans1

Dies ist das erste Argument der Antwortüberprüfungsfunktion. In asymmetrischen Tests wird dies als die Studierenden-Antwort angesehen, obwohl es jeder gültige CAS Ausdruck sein könnte. Es können darin auch Variablen aus der Aufgabe oder dem Feedback benutzt werden.

Inhalt: CAS-Ausdruck

TAns *

11 oder 3+Var1*2

Dies ist das zweite Argument der Antwortüberprüfungsfunktion. In asymmetrischen Tests wird dies als die Dozierenden-Antwort angesehen, obwohl es jeder gültige CAS Ausdruck sein könnte. Es können darin auch Variablen aus der Aufgabe oder dem Feedback benutzt werden.

Inhalt: CAS-Ausdruck

Schritt 6: Feedback festlegen

Schließlich muss im Rückmeldebaum ein **Feedback** festgelegt werden. Stimmen die **SAns** und **TAns** überein, wird der linke Weg des Baumes gegangen (grüner Pfad). Stimmen Sie nicht überein, wird der rechte Weg des Baumes gegangen (roter Pfad). Die Punkte, die dabei vergeben werden, tragen Sie in **Punkte wenn wahr** bzw. **Punkte wenn falsch** ein. Die Rückmeldung, die den Nutzer*innen angezeigt werden soll, schreiben Sie in das Feld **Feedback wenn wahr** bzw. **Feedback wenn falsch**. Darunter stellen Sie den **Feedback-Style, wenn wahr** bzw. **Feedback-Style, wenn falsch** ein. Dies entscheidet darüber, in welcher Farbe der Feedbacktext hinterlegt wird.

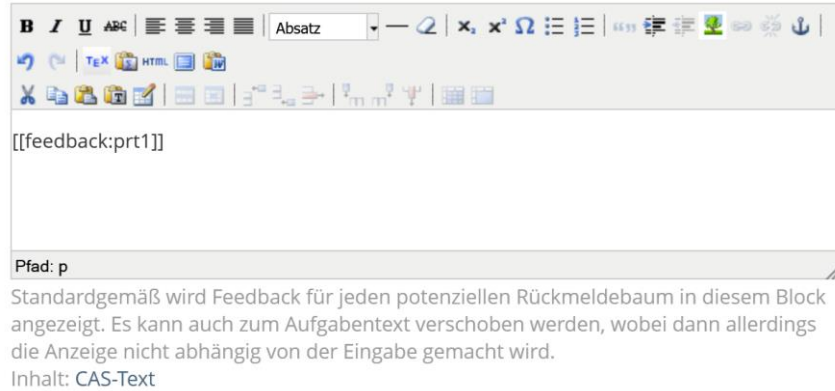
Hinweis: Die Punkte werden hierbei in Prozent angegeben, wobei 100% als 1 eingegeben werden. Den Aufbau eines Rückmeldebaumes, sowie die Punktevergabe können Sie in der graphischen Anzeige überprüfen.

In diesem Beispiel werden 100% der Punkte vergeben, wenn die eingegebene Antwort 11 lautet. Das angezeigte Feedback lautet: *Sie haben die Aufgabe richtig ausgerechnet.*

Mod wenn wahr	= <input type="button" value="v"/>	Wie die Bepunktung angepasst wird. "=" setzt die Punkte auf einen bestimmten Wert. "+/-" addieren oder subtrahieren Punkte von der aktuellen Summe.
Punkte wenn wahr	1	Bewertung wenn der Antworttest "wahr" ergibt.
Abzüge wenn wahr	0	Ziehe so viele Punkte ab.
Knoten wenn wahr	Ende <input type="button" value="v"/>	Knoten, mit dem fortgefahren wird (anhalten wenn leer).
Antworthinweis wenn wahr	prt1-1-T	Dieses Feld dient zur Berichterstattung. Es bestimmt den eindeutigen Pfad durch den Baum und das Ergebnis jeder Antwort. Es wird automatisch erzeugt, kann aber auch manuell zu etwas Sinnvollem geändert werden.
Feedback wenn wahr		
	<p>Sie haben die Aufgabe richtig ausgerechnet.</p> <p>Pfad: p</p> <p>Dieser CAS-Text kann die Aufgabenvariablen, die Eingabeelemente oder andere Feedbackvariablen verwenden. Er wird ausgewertet und angezeigt, sobald ein Studierender diesen Zweig erreicht.</p> <p>Inhalt: CAS-Text</p>	
Feedback-Style, wenn wahr	Richtig <input type="button" value="v"/>	Wählen Sie den Feedback-Style, wenn der Antworttest dieses Knotens wahr ist.

Stimmen **SAns** und **TAns** nicht überein, werden 0% der Punkt vergeben und es erscheint das Feedback: *Sie haben die Aufgabe leider nicht richtig ausgerechnet. Probieren Sie es gerne noch einmal aus!*

Spezifisches Feedback



The screenshot shows a rich text editor with a toolbar at the top containing various icons for text formatting (bold, italic, underline, font color, background color, bulleted list, numbered list, indent, outdent, link, unlink, image, video, table, etc.) and a dropdown menu set to 'Absatz'. Below the toolbar is a text area containing the placeholder code `[[feedback:prt1]]`. At the bottom of the editor, there is a small status bar showing 'Pfad: p'.

Standardgemäß wird Feedback für jeden potenziellen Rückmeldebaum in diesem Block angezeigt. Es kann auch zum Aufgabentext verschoben werden, wobei dann allerdings die Anzeige nicht abhängig von der Eingabe gemacht wird.
Inhalt: CAS-Text

Sobald ein neuer Rückmeldebaum gespeichert wird, wird der Feedback-Platzhalter für diesen Baum im Feld **Spezifisches Feedback** eingefügt. Der Platzhalter für das Feedback kann aus dem Feld **Spezifisches Feedback** per Cut-and-Paste an eine gewünschte Stelle in der Aufgabenstellung verschoben werden.

Tipp: Sinnvoll ist dies insbesondere, wenn in einer STACK-Frage mehrere Teilaufgaben mit einzelnen Feedbacks gestellt werden. Durch das Verschieben der Feedback-Platzhalter wird eindeutig, auf welchen Aufgabenteil sich das Feedback bezieht.

Schritt 7: Die Frage testen

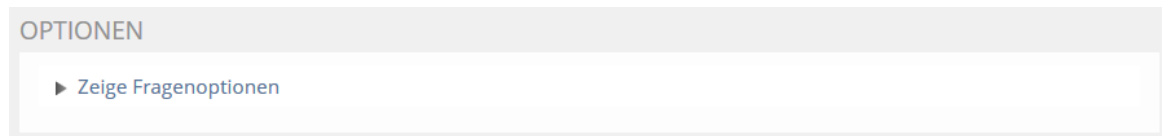
Schauen Sie sich die Frage über den Reiter **Vorschau** aus Sicht von Studierenden an. Geben Sie dort unterschiedliche Antworten ein, um alle möglichen Rückmeldungen des Rückmeldebaumes auszulösen und so zu überprüfen, ob die Frage wie gewünscht funktioniert.

5.3 Erweiterte Anleitung

Hier werden die einzelnen Einstellungen der STACK-Frage mitsamt deren Besonderheiten im Detail erklärt.

5.3.1 Aufgabenweite Optionen

Im Abschnitt **Optionen** befindet sich das Drop-Down Menü **Zeige Frageoptionen**. Hier werden Grundeinstellungen für die gesamte Aufgabe getroffen.



- Die erste Einstellung ist die **Aufgabenweite-Vereinfachung**. Aktivieren Sie diese Option, so werden Ausdrücke der gesamten STACK-Frage so weit wie möglich zusammengefasst. Dies bezieht sich insbesondere auf die Fragen-Variablen. Beispielsweise werden Brüche gekürzt und Terme zusammengefasst.

Für die Rückmeldebäume können Sie die Vereinfachung unabhängig davon ein- oder ausstellen. Um die Vereinfachung im Rückmeldebaum auszustellen, setzen Sie im entsprechenden Rückmeldebaum unter **Rückmeldebaum** → **Einstellungen** → **Vereinfachung** die Vereinfachung auf **Nein**.

Soll die Vereinfachung nur für bestimmte Fragen-Variablen ein- oder ausgestellt werden, können Sie den Befehl `simp:true` (dann ist die Vereinfachung an) oder `simp:false` (dann ist die Vereinfachung aus) in dem Feld Fragen-Variablen vor der entsprechenden Definition der Variable nutzen.

Hinweis: Ist **Aufgabenweite-Vereinfachung** aktiviert kann es sein, dass die Fragen-Variablen im Nachhinein dennoch vereinfacht werden. Deswegen kann es sinnvoll sein, die Aufgabenweite-Vereinfachung auszustellen und die Vereinfachung an den gewünschten Stellen manuell zu aktivieren.

- **Positiv annehmen** führt dazu, dass ein eingegebener Parameter, dessen Vorzeichen unbekannt ist, stets als positiv angenommen wird. (Für genauere Informationen siehe Kapitel 11.3 “Functions and Variables for Facts” (`assume_pos`) der Maxima-Dokumentation.)



http://maxima.sourceforge.net/docs/manual/maxima_59.html#Functions-and-Variables-for-Facts

- Darunter befinden sich Felder, in die Sie **Rückmeldungen** eingetragen können. Diese Rückmeldungen erscheinen abhängig davon, ob die Aufgabe richtig (100% der Punkte), teilweise richtig oder falsch (0% der Punkte) beantwortet wurde. Wir

empfehlen, diese Felder leer zu lassen und stattdessen eine umfangreiche Rückmeldung durch Rückmeldebäume zu erstellen.

Hinweis: Sie werden diese Felder in der ILIAS Versionen der UzK in der Regel leer vorfinden. In anderen STACK-Installationen kann es jedoch sein, dass hier standardmäßig ein Text eingetragen ist, der manuell entfernt werden kann.

- Anschließend folgen einige grundsätzliche Einstellungen zu der Darstellung der Frage. Hier können Sie die Anzeige der **Multiplikations-Zeichen**, der **Wurzelzeichen**, von **$\sqrt{-1}$** , der **inversen trigonometrischen Funktionen** sowie der **Matrix-Klammern** einstellen.

Wählen Sie Ihre gewünschte Darstellungsweise, indem Sie die entsprechende Option in der Drop-Down-Liste auswählen.

- In das Eingabefeld **Allgemeines Feedback** am Ende des Abschnittes **Optionen** können Sie Feedback eintragen, das immer, unabhängig von den gegebenen Antworten, unter der bestmöglichen Lösung angezeigt wird. Hier können Sie beispielsweise eine ausführliche Erklärung für die Lösung geben.

Hinweis: Das Allgemeine Feedback wird in einem Test nur angezeigt, wenn dort in den Einstellungen im Abschnitt **Direkte Rückmeldung** die Option **Bestmögliche Lösung anzeigen** aktiviert ist.

5.3.2 Einstellungen der Eingabefelder

Im Abschnitt **Test-Eingaben** haben Sie die Möglichkeit für jedes Eingabefeld im Fragentext, Einstellungen für dieses entsprechende Feld zu treffen. Klappen Sie hierzu die **Einstellungen der Eingabe** für dieses Eingabefeld auf.

TEST-EINGABEN

► Einstellungen der Eingabe ans1 *

- **Eingabetyp** gibt an, in welcher Form die Studierenden ihre Antworten eingeben sollen, damit STACK die Eingabe entsprechend auswerten kann. Die einzelnen Optionen werden im Kapitel [Eingabetypen](#) erklärt.
- Die **Musterlösung** muss für jedes Eingabefeld eingegeben werden. Sie wird den Studierenden, nachdem die Rückmeldung angefordert wurde, unter **Die bestmögliche Lösung lautet** angezeigt. Die Musterlösung wird in Maxima-Code angegeben und kann vorher definierte Fragen-Variablen enthalten.
Das Musterlösungsfeld besitzt je nach Eingabetyp auch noch andere Aufgaben. Dies ist im Kapitel [Eingabetypen](#) genauer erklärt.
Hinweis: Die Anzeige des Feldes **Die bestmögliche Lösung lautet** können Sie im Test für die Studierenden ausstellen, falls Sie dies wünschen.
Wichtig: Die Musterlösung entscheidet nicht darüber, ob eine Aufgabe als richtig oder falsch bewertet wird. Sie dient lediglich der Darstellung.
- Die **Eingabebreite** stellt ein, wie groß das entsprechende Eingabefeld angezeigt werden soll.
Hinweis: Für einige Eingabetypen wird die Eingabebreite automatisch ermittelt und angepasst.
- Ist **Strikte Syntax** aktiviert, wird strikte Maxima-Syntax bei der Eingabe erfordert, sodass Maxima das Eingegebene eindeutig interpretieren kann. Das wirkt sich beispielsweise darauf aus, ob Leerzeichen in den Fragen-Variablen erlaubt sind. Möchten Sie Leerzeichen zur Übersichtlichkeit des Codes in den Fragen-Variablen verwenden, ist es notwendig, dass Sie die Strikte Syntax ausstellen.
- Die Option **Sterne einfügen** bewirkt, dass Sterne automatisch als Multiplikationszeichen in die Eingabe eingefügt werden, sofern die Studierenden diese vergessen haben sollten. Hierbei kann es allerdings zu Fehlinterpretationen kommen.

- Unter **Syntax-Hinweis** können Sie einen Hinweis einfügen, der in dem Eingabefeld angezeigt, bevor die Studierenden etwas eintippen. Es bietet sich z.B. beim Eingabetyp Gleichungen an, die erste Zeile der Antwort als Syntax-Hinweis einzugeben.
Wichtig: Dieser Hinweis verschwindet nie automatisch, sondern muss manuell gelöscht werden, wenn er nicht Teil der Antwort sein soll.
- **Verbotene Worte** sind Zeichenketten, die in der Eingabe verboten sind. Dies wird beispielsweise wichtig, sobald Maxima-Befehle bei der Lösung helfen könnten, denn es ist möglich, Maxima-Ausdrücke in das Eingabefeld zu schreiben, die für die Antwort interpretiert werden. Damit die Antwort nicht mit Hilfe von Befehlen erzielt werden kann, sollten eben diese Befehle/Wörter/Zeichen unter **Verbotene Worte** eingegeben werden. Mehrere Wörter werden mit Kommata voneinander getrennt.
- Ist **Gleitkomma verbieten** aktiviert, wird jede Gleitkommazahl in der eingegebenen Antwort als falsch ausgewiesen. Falls Sie diese Option aktivieren, benötigen die Studierenden den Hinweis, dass sie Zahlen als Bruch bzw. ganze Zahl schreiben müssen.
- **Wörter zulassen** ist eine Funktion aus älteren STACK-Versionen. Sie wird nicht mehr benötigt.
- Ist **Verlange vollständige Kürzung** aktiviert, muss die Eingabe als kleinstmöglicher Term geschrieben werden. Ansonsten wird sie als falsch abgewiesen. Diese Option funktioniert jedoch ausschließlich bei Brüchen.
- Ist **Antwort-Typ überprüfen** aktiviert, werden Antworten mit einem anderen Typ als erwartet direkt als ungültig abgewiesen, bevor der Rückmeldebaum versucht diese auszuwerten.
- **Verifizierung erforderlich** funktioniert momentan nur in Kombination mit **Validierung anzeigen**.
- **Validierung anzeigen** funktioniert momentan nur in Kombination mit **Verifizierung erforderlich**. Diese Einstellung bewirkt, dass angezeigt wird, wie das System die Eingabe interpretiert. Diese Einstellung ist besonders hilfreich für Antworten, die lange Terme erfordern.
- **Zusätzliche Optionen** werden für einige Eingabetypen benötigt. Für gewöhnlich muss hier nichts eingetragen werden.

5.3.3 Rückmeldebäume (PRTS)

Im Abschnitt Rückmeldebäume (engl. potential response trees) werden die Rückmeldebäume der Aufgabe erstellt und bearbeitet. Sie machen STACK besonders, da sie dazu dienen, basierend auf der Eingabe der Studierenden passendes und individuelles Feedback zu geben, verschiedene Aufgabenteile auszuwerten und Punkte zu vergeben.

Rückmeldebäume bestehen aus einem oder mehreren Knoten, die miteinander verknüpft werden können. In jedem Knoten werden SAns und TAns miteinander abgeglichen. Stimmen sie überein, wird der grüne Pfad weitergegangen, ansonsten der rote, was zu unterschiedlichen Rückmeldungen führt. Die Struktur eines Rückmeldebaumes kann in der **Grafischen Anzeige** betrachtet werden.

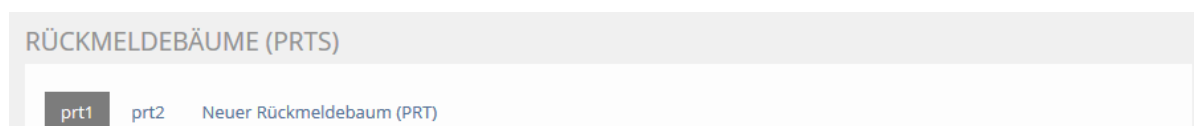


Wichtig: Der Rückmeldebaum beginnt immer bei dem Knoten, der als Startknoten eingestellt ist. Knoten, die in der Reihenfolge vor dem Startknoten kommen, werden nicht durchlaufen. In diesem Fall ist es notwendig den Startknoten zu ändern. Knoten, die nicht mit dem Startknoten verbunden sind, werden ebenfalls nicht durchlaufen. Hier ist es sinnvoll einen zweiten Rückmeldebaum anzulegen, da dieser unabhängig überprüft wird. Auch für separate Rückmeldungen zu unterschiedlichen Teilaufgaben müssen Sie für jede Teilaufgabe einen eigenen Rückmeldebaum erstellen.

Der Umfang der Rückmeldebäume ist abhängig von dem Detailreichtum der von Ihnen erstellten Rückmeldungen. Damit individuelle Rückmeldungen gegeben werden können, müssen häufige Fehler im Vorhinein antizipiert werden. Es ist jedoch auch möglich zunächst eine STACK-Frage mit wenig komplexen Rückmeldungen zu erstellen und später Rückmeldungen zu ergänzen.

Beispiele zum Aufbau von Bäumen und den damit verbundenen Möglichkeiten finden Sie in diesem Dokument unter [Übungsaufgaben](#).

Mittels der Reiter können Sie zwischen den verschiedenen Bäumen der Aufgabe umschalten. Ein Klick auf **Neuer Rückmeldebaum (PRT)** erstellt einen neuen Rückmeldebaum.



Sobald Sie einen Namen eingegeben *und* gespeichert haben, erscheint der neue Baum als Reiter in der Auswahl.

- **Erster Knoten:** Legt den ersten Knoten des Baumes fest. Alle gespeicherten Knoten können der erste Knoten sein. Achten Sie darauf, dass alle Knoten vom ersten aus erreichbar sind, so dass der Baum vollständig durchlaufen werden kann.

Durch das Aufklappen von **Einstellungen** erscheinen folgende Optionen:

- **Aufgabenwert:** Gibt die Gewichtung des Rückmeldebaumes an. Eine ausführliche Erklärung finden Sie in diesem Dokument unter [Punktevergabe → Rückmeldebäume gewichten](#).
- **Vereinfachung:** Falls Sie Vereinfachung auf **Ja** setzen, werden alle Variablen und Ausdrücke, die im Rückmeldebaum definiert und aufgerufen werden, vereinfacht. Die Einstellung hier ist unabhängig von der Einstellung unter **Optionen → Zeige Frageoptionen → Aufgabenweite-Vereinfachung**.
Hinweis: Ist **Aufgabenweite-Vereinfachung** aktiviert, kann es sein, dass die Variablen im Rückmeldebaum im Nachhinein dennoch vereinfacht werden. Stellen Sie deswegen im Zweifelsfall vorzugsweise die Aufgabenweite-Vereinfachung aus.
- **Feedback-Variablen:** Hier werden Variablen oder Funktionen definiert. Dabei ist es auch möglich, Variablen und Funktionen aus den **Fragen-Variablen** zu verwendet.
Wichtig: Die Verwendung hier definierter Variablen ist auf den jeweiligen Rückmeldebaum beschränkt.
- **Lösche Rückmeldebaum** löscht den Baum mitsamt aller darin enthaltenen Knoten.
Achtung: Dieser Button löscht den Rückmeldebaum sofort, ohne dass eine weitere Bestätigung erforderlich ist! Dieser Schritt ist nicht rückgängig zu machen!

Die Knoten des aktuell ausgewählten Baumes können ebenfalls über einen Reiter ausgewählt werden.

Der letzte Reiter ist dazu da, einen neuen Knoten zu erstellen.



Um einen **Neuen Knoten** zu erstellen, geben Sie etwas in **SAns** und **TAns** ein und klicken Sie **Speichern**. Danach kann der Knoten verbunden werden.

Folgende Einstellungen können für die einzelnen Knoten vorgenommen werden:

- **Antwortüberprüfung:** Hier wird die Antwortüberprüfungsoption für den Knoten ausgewählt. Mit der Antwortüberprüfung stellen Sie ein, durch welche

(mathematischen) Eigenschaften **SAns** und **TAns** verglichen werden sollen. Entsprechend dem, was überprüft werden soll, müssen Sie den passenden Antwortüberprüfungstypen auswählen. Die wichtigsten Optionen finden Sie im Kapitel [Antwortüberprüfungen](#).

- **SAns** und **TAns** sind (inkl. Test-Option) die Argumente, die an die Funktion der ausgewählten Antwortüberprüfung übergeben werden. In den üblichsten Fällen gilt: Stimmen **SAns** und **TAns** überein, ist der Knoten wahr. Weitere Einstellungen für diesen Fall werden dann in dem grün unterlegten Feld getroffen. Stimmen **SAns** und **TAns** nicht überein, ist der Knoten falsch. Die weiteren Einstellungen für diesen Fall sind in dem roten Feld zu tätigen.

Im einfachsten Fall wird in **SAns** der Name des Eingabefeldes (z.B. ans1) eingetragen und in **TAns** wird die Musterlösung kopiert. Für die Rückmeldung wird dann überprüft, ob die Studierendenantwort mit der Musterlösung übereinstimmt. Die Antwortüberprüfung gibt dann aus, ob der Knoten wahr (SAns und TAns stimmen im Sinne der Antwortüberprüfung überein) oder falsch (SAns und TAns stimmen nicht überein) ist.

Ob ein Knoten wahr oder falsch ist, sagt jedoch nicht unbedingt etwas darüber aus, ob die Aufgabe richtig beantwortet wurde oder nicht. Beispielsweise können Sie mit einem Knoten prüfen, ob ein bestimmter Fehler in der Aufgabe gemacht wurde. Wurde ebendieser Fehler gemacht, ist der Knoten wahr. Die Aufgabe selbst wurde allerdings falsch beantwortet.

- **Test-Optionen:** Hier geben Sie je nach Antwortüberprüfung Werte ein, die von der Überprüfungsfunktion verwendet werden. Ein Beispiel dafür können Sie in [Runden und Funktionen](#) sehen.

Hinweis: Wenn eine Antwortüberprüfung eine Test-Option benötigt, steht dies im Abschnitt [Antwortüberprüfungen](#) dabei.

- **Feedback unterdrücken:** Wenn diese Einstellung auf **Ja** eingestellt ist, wird automatisch generiertes Feedback unterdrückt, um ein mehrfaches Anzeigen verschiedener Feedbacks vermieden. In wenigen Fällen kann das automatische Feedback hilfreich sein. Standardmäßig ist aber zu empfehlen diese Einstellung auf **Ja** eingestellt zu lassen.
- **Lösche Knoten:** Löscht den aktuellen Knoten, sofern er nicht mit einem anderen Knoten verbunden ist. Um einen Knoten zu löschen, ist es also notwendig, zuerst all seine Verbindungen zu entfernen.

Die Antwortüberprüfung kann wahr oder falsch ausgehen. Dies bedeutet nicht, dass die Antwort der Studierenden wahr oder falsch ist, sondern nur, ob **SAns** und **TAns** übereinstimmen (wahr) oder verschieden sind (falsch). Stimmen SAns und TAns überein, wird der grüne Pfad weiter gegangen. Stimmen SAns und TAns nicht überein, wird der rote Pfad gegangen. Entsprechend werden die Punkte vergeben, das Feedback angezeigt und der Folgeknoten bestimmt.

Mod <div style="display: flex; align-items: center;"> <input style="width: 30px;" type="text" value="="/> <input style="width: 20px;" type="button" value="v"/> </div> <p style="font-size: small;">Wie die Bepunktung angepasst wird. "=" setzt die Punkte auf einen bestimmten Wert. "+/-" addieren oder subtrahieren Punkte von der aktuellen Summe.</p>	Mod <div style="display: flex; align-items: center;"> <input style="width: 30px;" type="text" value="="/> <input style="width: 20px;" type="button" value="v"/> </div> <p style="font-size: small;">Wie die Bepunktung angepasst wird. "=" setzt die Punkte auf einen bestimmten Wert. "+/-" addieren oder subtrahieren Punkte von der aktuellen Summe.</p>
--	--

- **Mod:** Gibt an, wie die Punkte verrechnet werden, falls dieser Ast des Baumes gegangen wird. Der Modus „+“ bewirkt, dass die Punkte, die in diesem Ast vergeben werden, zu den bisherigen Punkten addiert werden. Der Modus „-“ bewirkt, dass die in dem Ast vergebenen Punkte von den bisherigen Punkten angezogen werden. Der Modus „=“ bewirkt, dass die in vorherigen Knoten bereits gesammelten Punkte durch die neue Punktzahl ersetzt werden.

Tipp: Wir empfehlen den Modus „+“ zu wählen.

- **Punkte wenn wahr/falsch:** Hier können Sie die Punktzahl für den Knoten eingeben, welche Sie abhängig von der Rückmeldung der Überprüfungsfunktion geben möchten.
Anmerkung: Abzüge werden von ILIAS nicht unterstützt. Diese Funktion stammt aus STACK in moodle und wurde hier übernommen, damit die Kompatibilität zwischen den beiden Plattformen gegeben ist.
- **Knoten wenn wahr/falsch:** Hier geben Sie den Knoten an, der durchlaufen werden soll, nachdem der ausgewählte Knoten als wahr bzw. falsch bewertet wurde.
- **Antworthinweis wenn wahr/falsch:** Gibt den Pfad an, der bei einer Rückmeldung im Baum genommen wurde. Dieser Antworthinweis wird in einem zusammen mit der Antwort gespeichert, um nachvollziehen zu können, welche Rückmeldung angezeigt wurde. In diesem Feld müssen Sie nichts verändern.
- **Feedback wenn wahr/falsch:** Hier können Sie das Feedback eintragen, welches vom Rückmeldebaum ausgegeben werden soll, wenn der entsprechende Pfad in der Rückmeldung durchlaufen wird.

Hinweis: Sie können in dem Feedback zuvor definierte Variablen verwenden. Dies funktioniert analog zum Fragetext über {@Variablenname@}.

- **Feedback-Style:** Unterhalb des Eingabefeldes für das Feedback können vordefinierte Feedback-Styles ausgewählt werden. Die Benennungen geben einen Hinweis darauf, in welchen Situationen der entsprechende Feedback-Style verwendet werden kann. Diese sind jedoch lediglich eine Empfehlung. Abgesehen von dem Erscheinungsbild der Rückmeldung ändert diese Einstellung nichts und die Bewertung wird dadurch nicht beeinflusst. Sie können also den Feedbackstyle wählen, der Ihnen für die entsprechende Situation am meisten zusagt.

Hinweis: Bei der Verteilung der Punkte ist es wichtig, dass die Punkte so verteilt werden, dass es einen Pfad gibt, der maximal 1 ergibt. Dies ist wichtig, denn die Studierenden erhalten die volle Punktzahl für den Rückmeldebaum, wenn sie im Rückmeldebaum einen Punkt (100%) erhalten haben. Zudem ist es wichtig, dass die Pfade (in der Summe) maximal 1 ergeben. Falls Sie den Studierenden mehr Punkte für die gesamte Aufgabe geben möchten, stellen Sie unter „Bewertung“ die Punkte für die Aufgabe ein. Soll außerdem ein Rückmeldebaum mehr gewichtet werden als ein anderer, so sollten Sie den Aufgabenwert dementsprechend einstellen.

5.3.4 Punktevergabe

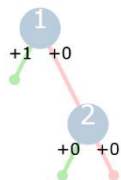
Die Punktevergabe wird an drei unterschiedlichen Stellen eingestellt. Dies wirkt auf den ersten Blick möglicherweise viel, doch dies ermöglicht es, die Bepunktung einfach und flexibel anzupassen. In den Rückmeldebäumen wird eingestellt, wie viel Prozent der Punkte des Rückmeldebaumes für welche Antworten geben werden. Die Gewichtung der Rückmeldebäume bestimmt, wie groß der Anteil der Gesamtpunkt ist, die in jedem Rückmeldebaum maximal erreicht werden können. Die Gesamtpunktzahl wird zentral für die gesamte Aufgabe eingestellt.

Bewertung innerhalb eines Rückmeldebaumes

Innerhalb eines Rückmeldebaumes wird festgelegt, wie viel Prozent der Punkte des Rückmeldebaumes für welche Antwort gegeben werden. Die Punkte entlang eines Pfades können in der **Grafischen Anzeige** betrachtet werden.

Hinweis: Bei der Verteilung der Punkte ist es wichtig darauf zu achten, dass es einen Pfad gibt, der insgesamt 1 ergibt. Nur so ist es möglich, dass die Studierenden die volle Punktzahl erreichen können, denn 1 entspricht hierbei 100% der Punkte. Zudem ist es wichtig, dass die Punkte entlang eines Pfades (in der Summe) maximal 1 ergeben. Alles was über 1 (100%) hinaus geht wird auf 100% abgerundet.

prt1 Neuer Rückmeldebaum (PRT)
Grafische Anzeige

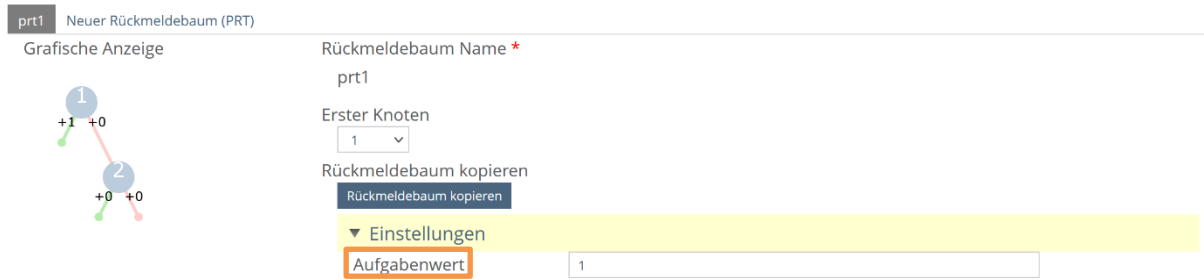


Beispiel: In dem nebenstehenden Bild werden 100% der Punkte vergeben, wenn die Studierenden die Antwort eingegeben haben, die im ersten Knoten zu dem grünen Pfad führt. Für alle anderen Pfade werden 0% der Punkte vergeben. Sie werden lediglich für ein individuelles Feedback verwendet.

Rückmeldebäume gewichten

Die Gewichtung eines Rückmeldebaumes wird über das Feld **Aufgabenwert** in den Einstellungen des entsprechenden Rückmeldebaumes angepasst. Der Aufgabenwert gibt an, wie die Rückmeldebäume in der Bepunktung im Verhältnis zueinander gewichtet werden. Haben zwei Rückmeldebäume den gleichen Aufgabenwert, wird ihnen die gleiche maximale Punktzahl zugeordnet. Hat ein Baum einen doppelt so hohen Aufgabenwert wie ein anderer, ist die Maximalpunktzahl dort entsprechend doppelt so hoch. Die absoluten Zahlen haben hier keinen Einfluss, nur das Verhältnis zwischen den Bäumen ist entscheidend.

Hinweis: Insbesondere bestimmt der Aufgabenwert der Rückmeldebäume nicht die Gesamtpunktzahl für die STACK-Aufgabe. Er beeinflusst lediglich, wie die Gesamtpunktzahl auf die Rückmeldebäume verteilt wird.



The screenshot shows the 'Neuer Rückmeldebaum (PRT)' configuration page. On the left, a 'Grafische Anzeige' (Graphical View) shows a tree with two nodes: node 1 (blue circle) with a green arrow pointing to it labeled '+1' and a red arrow pointing away labeled '+0'; node 2 (blue circle) with a green arrow pointing to it labeled '+0' and a red arrow pointing away labeled '+0'. On the right, the 'Rückmeldebaum Name' is 'prt1'. The 'Erster Knoten' (First Node) is set to '1'. There is a 'Rückmeldebaum kopieren' button. Below this is a yellow 'Einstellungen' (Settings) section with a sub-section 'Aufgabenwert' (Task Value) set to '1'.

Beispiel: prt1 hat den Aufgabenwert 1, prt2 hat den Aufgabenwert 2. Dies bedeutet, dass prt2 doppelt so viele Punkte gibt, wie prt1. Prt1 gibt dementsprechend maximal $\frac{1}{3}$ der Gesamtpunkte und prt2 gibt maximal $\frac{2}{3}$ der Gesamtpunkte.

Gesamtpunktzahl

Die Gesamtpunktzahl für eine STACK-Aufgabe wird über den Reiter **Bewertung** oben auf der Seite angepasst.



The screenshot shows the top navigation bar of the STACK interface. The tabs are: 'Fragenpool für Tests', 'Seite bearbeiten', 'Vorschau', 'Eigenschaften', 'Rückmeldung', and 'Statistik'. The 'Bewertung' tab is selected and highlighted with an orange border. Below the tabs, there is a row of links: 'Frage bearbeiten', 'Bewertung' (highlighted with an orange border), 'Vorbereitete Zufallswerte', 'Testfälle', 'Fragen aus MoodleXML importieren', and 'Export nach MoodleXML'.

Zuoberst wird die **Aktuelle Punktzahl** angezeigt. Darunter kann im Feld **Neue Punktzahl** die Gesamtpunktzahl angepasst werden.

Besonders interessant an dieser Ansicht ist, dass die maximal zu erhaltenden Punkte in den einzelnen Rückmeldebäumen sowie den Knoten zu sehen sind. Somit ist schnell zu erkennen, ob die Gesamtpunktzahl oder die Gewichtung schöne Werte ergeben. Zudem kann mit Hilfe des Button **Zeige Vergleich** eine Gegenüberstellung der aktuellen Punktzahl mit der neu eingegebenen Punktzahl angezeigt werden.

Fragenpool für Tests
Seite bearbeiten
Vorschau
Eigenschaften
Rückmeldung
Statistik

Frage bearbeiten
Bewertung
Vorbereitete Zufallswerte
Testfälle
Fragen aus MoodleXML importieren
Export nach MoodleXML

BEWERTUNGS-STATUS

Aktuelle Punktzahl

6

Dies sind die aktuell maximal erreichbaren Punkte für diese Frage.

Neue Punktzahl

Hier können sie einen neuen Wert eingeben, um die Berechnung aus den Teilpunkten zu vergleichen oder eine neue maximale Punktzahl festzulegen.

Diese Seite zeigt die Fragenbewertung in Punkten. Für jeden Knoten eines Ergebnisbaums werden die Punkte des positiven Antworttests in grün und die des negativen Antworttests in rot angezeigt. Für jeden Ergebnisbaum zeigen die maximal erreichbaren Punkte seine Gewichtung innerhalb der Frage

Zeige Vergleich
Punktzahl speichern

Name des Auswertungsbaums: prt1	Maximale Punktzahl für diesen Auswertungsbaum: 2	
Knotenname: 1	=2	=0
Knotenname: 2	+0	+0
Name des Auswertungsbaums: prt2	Maximale Punktzahl für diesen Auswertungsbaum: 4	
Knotenname: 1	=4	=0

Beispiel: Die Gesamtpunktzahl von 6 wird durch die Gewichtung der Rückmeldebäume so verteilt, dass prt1 maximal 2 Punkte ergibt. Prt2 ergibt maximal 4 Punkte.

5.3.5 Eingabetypen

Dieser Abschnitt bietet eine kurze Erklärung zu den einzelnen Eingabetypen. Der Eingabetyp entscheidet darüber, wie das Eingabefeld für die Studierenden angezeigt wird und welche Eingaben möglich sind. Jedes Eingabefeld muss anschließend mit Hilfe einer [Antwortüberprüfung](#) im Rückmeldebaum überprüft werden. Wir werden zu jedem Eingabetypen kurz angeben, welche Antwortüberprüfungen die gängigsten sind. Welche Antwortüberprüfung in Ihrem Fall die passende ist hängt jedoch davon ab, wie Sie Ihre STACK-Frage gestalten möchten. Lesen Sie sich hierzu auch den Abschnitt zu den [Antwortüberprüfungen](#) durch. Sollten noch Fragen offenbleiben, beraten wir Sie gerne.

▼ Einstellungen der Eingabe ans1 *

Eingabetyp *

Musterlösung *

Eingabebreite

Strikte Syntax

Sterne einfügen

Algebraische Eingabe

Algebraische Eingabe

Wahr/Falsch

Matrix

Einzelzeichen

Mehrzeilig

Kontrollkästchen

Drop-Down-Liste

Gleichungen

Notizen

Radio-Auswahl

Einheiten

Zeichenkette

Numerisch

Wenn aktiviert, werden autom

Algebraische Eingabe

Mit diesem Eingabetyp können Antworten in Form von Formeln, Zahlen oder Buchstaben ohne Leerzeichen gegeben werden. Dieser Eingabetyp eignet sich daher besonders gut für mathematische Fragen. In die Musterlösung wird die richtige Lösung eingegeben.

Eingabetypen: Algebraische Eingabe

Leiten Sie folgende Funktion ab:

$f(x) = x^3 + 4 \cdot x$

$f'(x) =$ ✓

$3 \cdot x^2 + 4$ wurde wie folgt interpretiert: $3 \cdot x^2 + 4$

Die folgenden Variablen wurden gefunden: $[x]$

In ihrer Antwort fehlen eine oder mehrere Variablen!

Rückmeldung anfordern

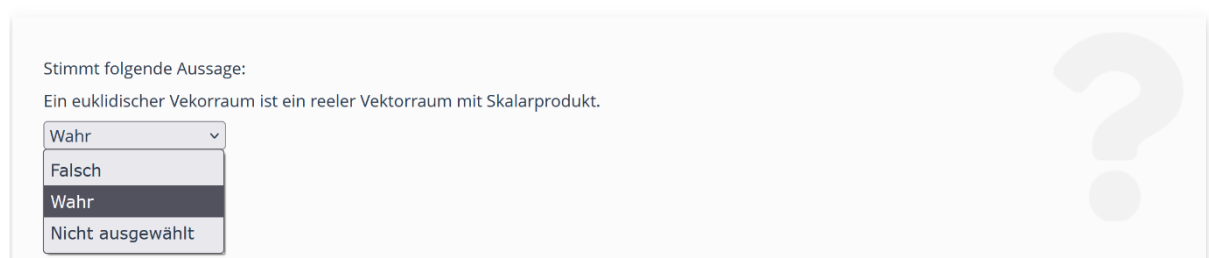
Für die Überprüfung wird der Name des Eingabefeldes (z.B. ans1) in **SAns** eingegeben. In **TAns** wird die richtige Lösung oder ein antizipierter Fehler eingegeben, auf den überprüft werden soll, eingegeben.

Gängige Antwortüberprüfungen: AlgEquiv, EqualComAss, NumAbsolute, NumRelative, String

Wahr/Falsch

Mit diesem Eingabetypen erscheint in der STACK-Frage ein Drop-Down Menü, in dem **Wahr** oder **Falsch** ausgewählt werden kann. Dieser Eingabetyp eignet sich daher gut für Entscheidungsfragen. In die Musterlösung wird entweder **Wahr** oder **Falsch** eingegeben.

Eingabetypen: Wahr / Falsch



Stimmt folgende Aussage:

Ein euklidischer Vektorraum ist ein reeller Vektorraum mit Skalarprodukt.

Wahr

Falsch

Wahr

Nicht ausgewählt

Für die Überprüfung wird der Name des Eingabefeldes (z.B. ans1) in **SAns** eingetragen. In **TAns** wird entsprechend *true* oder *false* eingegeben.

Gängige Antwortüberprüfungen: AlgEquiv, String

Matrix

Mit diesem Eingabetypen wird die Antwort in Form einer Matrix ermöglicht. Die Musterlösung spielt bei diesem Eingabetypen eine besondere Rolle. Sie gibt nicht nur die bestmögliche Lösung an, sondern wird auch verwendet, um festzulegen, wie viele Zeilen und Spalten das Eingabefeld haben wird. Das Eingabefeld wird so viele Zeilen und Spalten haben, wie die Musterlösung. Geben Sie dazu die Musterlösung als Matrix an. Das machen Sie, indem Sie den Maxima-Befehl `matrix([..., ...],[..., ...],[..., ...] ...)` Dabei sind die Einträge in den eckigen Klammern jeweils eine Zeile. Die Einträge in den Zeilen werden mit Komma getrennt. Eine 2x2-Matrix wird beispielsweise eingegeben als `matrix([a,b],[c,d])`.

Verwendung fester Zahlenwerte:

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 2 & 1 \\ 0 & 2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 2 & 0 \\ 1 & 1 & 0 \\ 2 & 0 & 0 \end{pmatrix}$$

Berechnen Sie $C = A * B$

In ihrer Antwort fehlen eine oder mehrere Variablen!

Verwendung von Zufallsvariablen:

$$D = \begin{pmatrix} 2 & 0 & 2 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \quad E = \begin{pmatrix} 0 & 2 & 2 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \end{pmatrix}$$

Berechnen Sie $F = D - E$

In ihrer Antwort fehlen eine oder mehrere Variablen!

[Rückmeldung anfordern](#)

Hinweis: Sie erleichtern sich die Überprüfung der Antworten, indem Sie die richtige Lösung als Variable in den Fragen-Variablen hinterlegen (z.B. `MusterMatrix: matrix([a,b],[c,d])`) und diese dann in der Musterlösung und für die Antwortüberprüfung aufrufen.

Für die Überprüfung kann die Matrix als Gesamtes überprüft werden. Es ist jedoch auch möglich, einzelne Zeilen, Spalten oder Einträge zu überprüfen. Zur Überprüfung der ganzen Matrix geben Sie den Namen des Eingabefeldes (z.B. `ans1`) in **SA**ns ein. In **TA**ns wird die richtige Lösung in Form einer Matrix angegeben (z.B. `matrix([a,b],[c,d])` oder `MusterMatrix`). Wenn Sie eine Zeile der Matrix überprüfen möchten, geben Sie den Namen des Eingabefeldes zusammen mit der Zeilennummer *i* (z.B. `ans1[i]`) in **SA**ns ein. In **TA**ns wird die entsprechend richtige Zeile (z.B. `MusterMatrix[i]` oder `[a,b]`) eingegeben. Für eine Spalte *j* müssen Sie die Matrix transponieren und dann wie bei der Überprüfung einer Zeile vorgehen (z.B. `transpose(ans1)[j]` für **SA**ns und `transpose(MusterMatrix)[j]` für **TA**ns). Einzelne Einträge überprüfen Sie, indem Sie z.B. `ans1[i,j]` ins **SA**ns und `MusterMatrix[i,j]` in **TA**ns eingeben.

Gängige Antwortüberprüfungen: AlgEquiv, StringSloppy

Einzelzeichen

Mit diesem Eingabetypen kann in das Eingabefeld nur ein einzelnes Zeichen eingegeben werden. Die Eingabebreite lässt sich nicht verändern. Dieser Eingabetyp hat den Vorteil, dass es nicht möglich ist mehr Zeichen einzugeben, was mögliche Fehlerquellen minimiert. In die Musterlösung wird das richtige Zeichen eingegeben.

Eingabetypen: Einzelzeichen

Was kennzeichnet einen euklidischen Vektorraum?

- a) Ein euklidischer Vektorraum ist ein topologischer Raum über einem topologischen Körper
- b) Ein euklidischer Vektorraum ist ein komplexer Vektorraum auf dem ein inneres Produkt definiert ist.
- c) Ein euklidischer Vektorraum ist ein reeller Vektorraum mit Skalarprodukt.
- d) Keine der Antworten ist richtig.

Tragen Sie den Buchstaben zur richtigen Lösung in das Feld.

In ihrer Antwort fehlen eine oder mehrere Variablen!

[Rückmeldung anfordern](#)

Für die Überprüfung wird der Name des Eingabefeldes (z.B. ans 1) in **SAns** eingetragen. In **TAns** wird das Zeichen eingegeben, das überprüft werden soll.

Gängige Antwortüberprüfungen: AlgEquiv, String

Mehrzeilig

Mit diesem Eingabetypen wird in der Frage ein Textfeld eingefügt, in dessen Zeilen algebraische Ausdrücke geschrieben werden können. Zeichenkennungen können mit Anführungszeichen als Antwort eingegeben und in der Musterlösung verwendet werden.

Eingabetypen: Mehrzeilig

Berechnen Sie die Nullstelle(n) der folgenden Parabel:



Informationen zur Syntax: Schreiben Sie die Lösungen bitte untereinander.

$$f(x) = x^2 - 1$$

x=1
x=-1



x=1
x=-1

wurde wie folgt interpretiert:

$x = 1$
 $x = -1$

Die folgenden Variablen wurden gefunden: $[x]$

In ihrer Antwort fehlen eine oder mehrere Variablen!

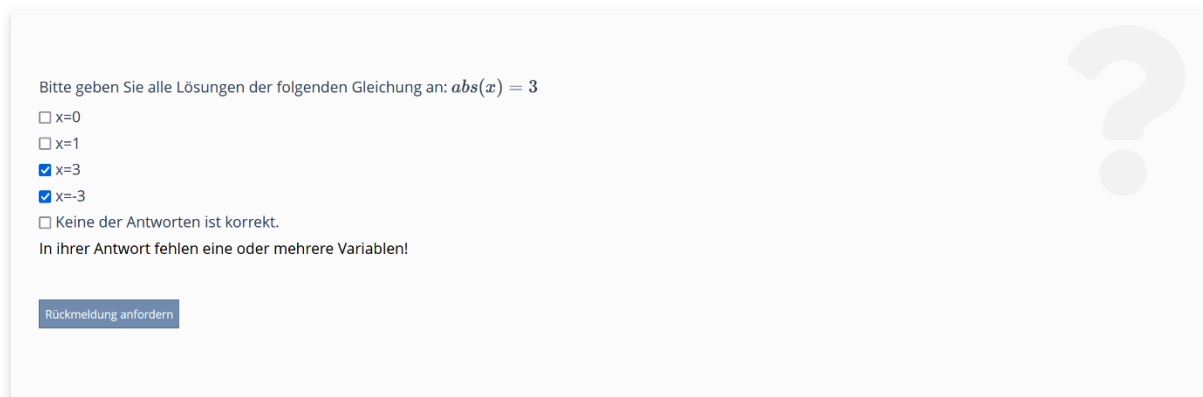
[Rückmeldung anfordern](#)

Hinweis: Wir beschreiben diesen Eingabetypen nicht näher, da er weitestgehend durch den Eingabetypen [Gleichungen](#) ersetzt wurde.

Kontrollkästchen (Multiple-Choice-Aufgabe)

Mit diesem Eingabetyp können Sie Multiple-Choice-Aufgaben erstellen. Hierbei ist es möglich, den Studierenden Antwortmöglichkeiten zu bieten, aus denen sie für ihre Antwort auswählen können.

Eingabetypen: Kontrollkästchen



Bitte geben Sie alle Lösungen der folgenden Gleichung an: $\text{abs}(x) = 3$

☐ $x=0$

☐ $x=1$

☒ $x=3$

☒ $x=-3$

☐ Keine der Antworten ist korrekt.

In Ihrer Antwort fehlen eine oder mehrere Variablen!

[Rückmeldung anfordern](#)

Die Musterlösung spielt bei diesem Eingabetypen eine besondere Rolle. Die Musterlösung legt fest, welche Antwortmöglichkeiten den Studierenden angezeigt werden und welche davon in der Bestmöglichen Lösung als richtig angezeigt werden. Die Musterlösung hat hierbei eine spezielle Form. Sie geben die Musterlösung an, indem Sie folgende Syntax verwenden: `[[internerName1,true/false,"Anzeigetext"],...,[internerNameN,true/false,"Anzeigetext"]]`

Jeder Eintrag in einer eckigen Klammer entspricht einer möglichen Antwortoption der Aufgabe. Jede Antwortoption besteht aus drei Elementen.

1. *internerName* ist für die interne Weiterverarbeitung der Antwortoption relevant. Er wird vor allen Dingen für die Überprüfung der Antwort im Rückmeldebaum verwendet. Es ist wichtig, dass der interne Name keine Lehrzeichen enthält und nach Zahlen keine weiteren Buchstaben folgen.
2. *true* oder *false* gibt an, ob die entsprechende Antwortmöglichkeit in der Bestmöglichen Lösung als richtig (*true*) oder falsch (*false*) angezeigt wird.
3. "*Anzeigetext*" entscheidet darüber, wie die Antwortoption den Studierenden angezeigt wird. Der Anzeigetext kann Leerzeichen und LaTeX-Code beinhalten. Dabei ist es wichtig, dass der Anzeigetext in Anführungszeichen geschrieben wird. Der Anzeigetext ist optional. Wird kein Anzeigetext eingegeben, wird der interne Name angezeigt.

Hinweis: Wenn der interne Name gut verständlich für die Studierenden ist, kann der Anzeigetext weggelassen werden. Wenn die Antwortoption aus mehreren Wörtern besteht oder LaTeX-Code beinhalten soll, ist es notwendig, einen Anzeigetext zu verwenden. Für Antwortoptionen, die [LaTeX-Eingaben](#) beinhalten sollen, verwenden Sie `"\\(LaTeX-Formel\\)"` als „Anzeigetext“. Achten Sie hierbei darauf, bei jedem Befehl „`\\`“ zu verwenden, anstelle von „`\`“.

Hinweis: Bitte verwenden Sie in der Musterlösung außerhalb des Anzeigetextes keine Leerzeichen. Wenn Sie Leerzeichen zwischen den Listenelementen verwenden möchten, um die Musterlösung übersichtlicher zu gestalten, ist es notwendig, dass Sie in den [Einstellungen der Eingabefelder](#) den Haken hinter **Strikte Syntax** entfernen.

Wenn Sie überprüfen möchten, ob die **gegebene Antwort vollständig richtig** ist, tragen Sie den Namen des Eingabefeldes z.B. ans1 in **SAns** ein. In **TAns** wird die Liste der richtigen internen Namen z.B. [internerName1,internerName4] eingegeben.

Achtung: Listen sind in Maxima nur identisch, wenn alle Elemente in der gleichen Reihenfolge angeordnet sind. Deswegen ist die Reihenfolge der Listenelemente für die Überprüfung wichtig. Wir empfehlen, den Maxima-Befehl *sort(list)* zu nutzen, um **SAns** und **TAns** zu vergleichen. Geben Sie also beispielsweise *sort(ans1)* in SAns ein und verwenden Sie beispielsweise *sort([internerName1,internerName4])* in TAns.

Es ist auch möglich, **einzelne Elemente der Antwort zu überprüfen**. Hierbei ist es wichtig zu wissen, dass der Name des Eingabefeldes (z.B. ans1) eine Liste ist. Diese Liste beinhaltet alle internen Namen der Antwortoptionen, die von den Studierenden ausgewählt wurden. Um zu überprüfen, ob ein bestimmtes Element von den Studierenden ausgewählt wurde, schreiben Sie *member(internerNameX,ans1)* in **SAns** und *true* in **TAns**. Wenn die Antwortoption *internerNameX* von den Studierenden ausgewählt wurde, wird im Rückmeldebaum der grüne Pfad weiter gegangen, wenn die Antwortoption nicht gewählt wurde, wird der rote Pfad weiter gegangen.

Gängige Antwortüberprüfungen: AlgEquiv, String

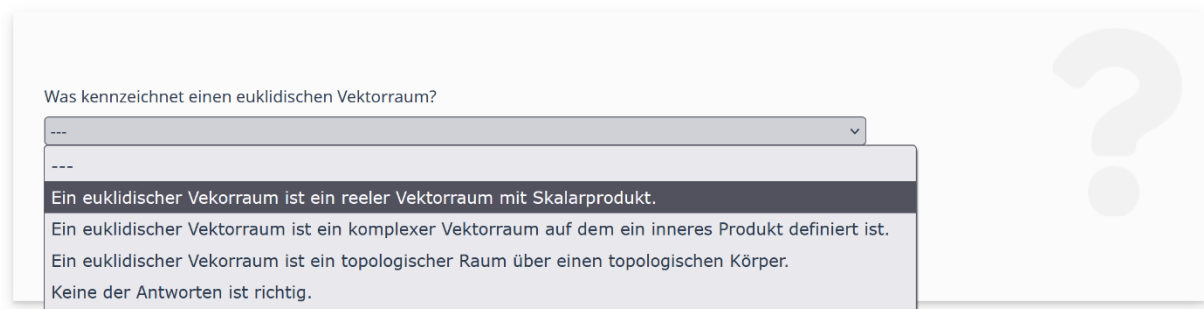
Besondere Einstellungen:

- Im Rückmeldebaum befindet sich die Einstellung **Feedback unterdrücken**. Diese Option verhindert, dass vom System automatisch generierte Rückmeldungen wie „Ihre Liste sollte 2 Elemente enthalten, sie hat aber 1.“ oder „Die rot unterstrichenen Einträge sind falsch.“ nicht angezeigt werden.
- Wenn Sie den Befehl *nonotanswered* in **Test-Eingaben** → **Einstellung der Eingabe ansN** → **Zusätzliche Optionen** eingeben, wird die Darstellung der Antwortmöglichkeit „Nicht ausgewählt“ verhindert.
- Mit Hilfe des Maxima-Befehls *random_permutation(MusterLoesung)* können die Antwortmöglichkeiten in eine zufällige Reihenfolge gebracht werden. Wenn Sie dies verwenden möchten, ist es notwendig, dass Sie SAns und TAns mit Hilfe des Befehls *sort(liste)* sortieren.
- Mittels *mcq_correct(MusterLoesung)* und *mcq_incorrect(MusterLoesung)* kann eine Liste an korrekten/inkorrekten der Musterlösung ausgegeben werden. Diese können Sie für die Überprüfung der Antworten verwenden.

Drop-Down-Liste (Single-Choice-Aufgabe)

Mit diesem Eingabetyp können Sie Single-Choice-Aufgaben erstellen. Die Antwortmöglichkeiten werden hierbei als Drop-Down-Liste angezeigt.

Eingabetypen: Drop-Down-Liste



Die Musterlösung spielt bei diesem Eingabetypen eine besondere Rolle. Die Musterlösung legt fest, welche Antwortmöglichkeiten den Studierenden angezeigt werden und welche davon in der Bestmöglichen Lösung als richtig angezeigt werden. Die spezielle Form der Musterlösung ist hierbei analog zu dem Eingabetypen [Kontrollkästchen](#). Schauen Sie für die Syntax der Musterlösung in diesen Abschnitt.

Achtung: Für den Eingabetypen Drop-Down-Liste ist es wichtig, dass in der Musterlösung nur eine Antwortmöglichkeit als *true* ausgewiesen werden kann.

Hinweis: Die Möglichkeit LaTeX in der *Drop-Down-Liste* zu verwenden gibt es leider nicht. Das geht nur mit der Single-Choice Option *Radioauswahl* und der Multiple-Choice Option *Kontrollkästchen*.

Für die Überprüfung wird der Name des Eingabetypen (z.B. ans1) in **SA**ns eingegeben. In **TA**ns geben Sie den internen Namen der Antwortoption, auf die Sie überprüfen möchten, ein.

Gängige Antwortüberprüfungen: AlgEquiv, String


Besondere Einstellungen:

- Im Rückmeldebaum befindet sich die Einstellung **Feedback unterdrücken**. Diese Option verhindert, dass vom System automatisch generierte Rückmeldungen wie „Ihre Liste sollte 2 Elemente enthalten, sie hat aber 1.“ oder „Die rot unterstrichenen Einträge sind falsch.“ nicht angezeigt werden.
- Der Befehl *nonotanswered* in **Test-Eingaben** → **Einstellung der Eingabe ansN** → **Zusätzliche Optionen** verhindert die Darstellung der Antwortmöglichkeit „Nicht ausgewählt“.
- Mit Hilfe des Maxima-Befehls *random_permutation(MusterLoesung)* können die Antwortmöglichkeiten in eine zufällige Reihenfolge gebracht werden. Wenn Sie dies verwenden möchten, ist es notwendig, dass Sie SAns und TAns mit Hilfe des Befehls *sort(liste)* sortieren.

Dieser Eingabetyp ermöglicht es, einzelne Äquivalenzumformungen zu überprüfen. Daher ist er besonders für Aufgaben, in denen Gleichungen oder Ungleichungen gelöst werden sollen, geeignet. Den Studierenden wird über das **Automatisch generierte Feedback** direkt zurückgemeldet, welche Äquivalenzumformungen fehlerhaft sind.


Eingabetypen: Gleichungen

Formen Sie die Gleichung $x^2 - x = 30$ so um, dass Sie die Nullstellen herausbekommen. Starten Sie mit der gegebenen Gleichung als erste Zeile und benutzen Sie für jeden Schritt der Umformung eine neue Zeile.



Informationen zur Syntax: Jede Zeile sollte eine Gleichung beinhalten, genauso wie jede Zeile eine neue Umformung darstellen soll. Das mathematische "Oder" wird als "or" und das mathematische "Und" als "and" geschrieben. Potenzen schreiben Sie, indem sie die Exponenten mit einem "^" hinter die Basis schreiben, also z.B. so: 2^3

$x^2 - x = 30$
 $x^2 - x - 30 = 0$
 $x - 6 = 0$ nounor $x + 5 = 0$
 $x = 6$ nounor $x = -5$



wurde wie folgt interpretiert:

$x^2 - x = 30$
 $x^2 - x - 30 = 0$
 $x - 6 = 0$ nounor $x + 5 = 0$
 $x = 6$ nounor $x = -5$

$x^2 - x = 30$
 $\leftrightarrow x^2 - x - 30 = 0$
 $\leftrightarrow x - 6 = 0$ or $x + 5 = 0$
 $\leftrightarrow x = 6$ or $x = -5$

In ihrer Antwort fehlen eine oder mehrere Variablen!

Rückmeldung anfordern

Die Musterlösung wird hierbei als Liste von Äquivalenzumformungen angegeben. Sie geben die Musterlösung an, indem Sie folgende Syntax verwenden: [Gleichung1,...,GleichungN]
Hierbei sollte *Gleichung1* die in der Frage gestellte Anfangsgleichung sein. *GleichungN* sollte die Lösung sein.

Für die Überprüfung wird last(ans1) in **SAns** eingetragen. In **TAns** wird last(Muster) eingetragen.

Die einzelnen Zeilen der Antwort werden als Liste in `ans1` gespeichert und können somit auch einzeln abgeglichen werden. Nutzen Sie zur Antwortüberprüfung im **Rückmeldebaum** den Befehl `last(...)`, um auf den letzten Eintrag der Liste mit Äquivalenzumformungen zuzugreifen.

Gängige Antwortüberprüfungen: EquivReasoning, EquivFirst

Besondere Einstellungen:

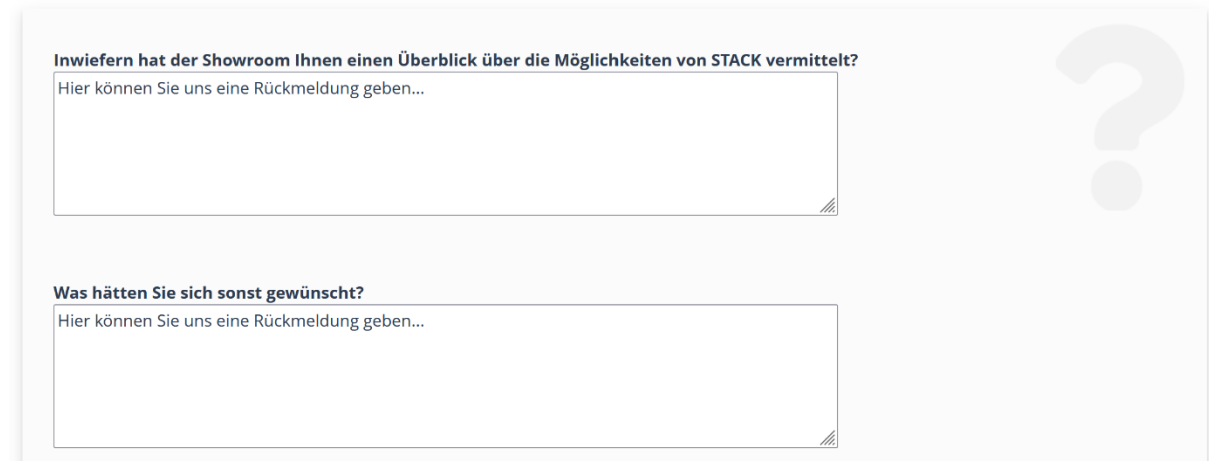
- Wir empfehlen, die **Eingabebreite**, unter *Einstellungen der Eingabe ansX*, etwas breiter als den Defaultwert von 15 einzustellen.
- Logische Operatoren werden in einer speziellen Form in den Gleichungen verwendet. Das logische und wird geschrieben als *nounand*. Das logische oder wird geschrieben als *nounor*. Beispielsweise $x = -5 \wedge x = 6$ wird in STACK als „x=-5 nounand x=6“

geschrieben. Dies gilt nur für die Autor*innen. Studierende können *or* oder *and* verwenden.

Notizen

Mit diesem Eingabetypen können Freitexte als Antwort gegeben werden. Dieser Eingabetyp eignet sich daher beispielsweise dazu, Rückmeldungen von Studierenden zu erfragen. Hierzu empfiehlt es sich, die Eingabebreite größer als den Default-Wert von 15 einzustellen.

Eingabetypen: Notizen



Für die Überprüfung ist wichtig zu wissen, dass der Variablen für das Eingabefeld (z.B. ans1) der Wert *false* zugeordnet wird, wenn keine Antwort in das Eingabefeld eingegeben wurde. Sobald von den Studierenden irgendetwas in das Eingabefeld eingegeben wurde, wird der Variablen für das Eingabefeld (z.B. ans1) der Wert *true* zugeordnet. Im Rückmeldebaum kann also nur überprüft werden, ob eine Antwort gegeben wurde oder nicht. Geben Sie hierzu den Namen des Eingabefeldes (z.B. ans1) in **SAns** ein und true oder false in **TAns**.

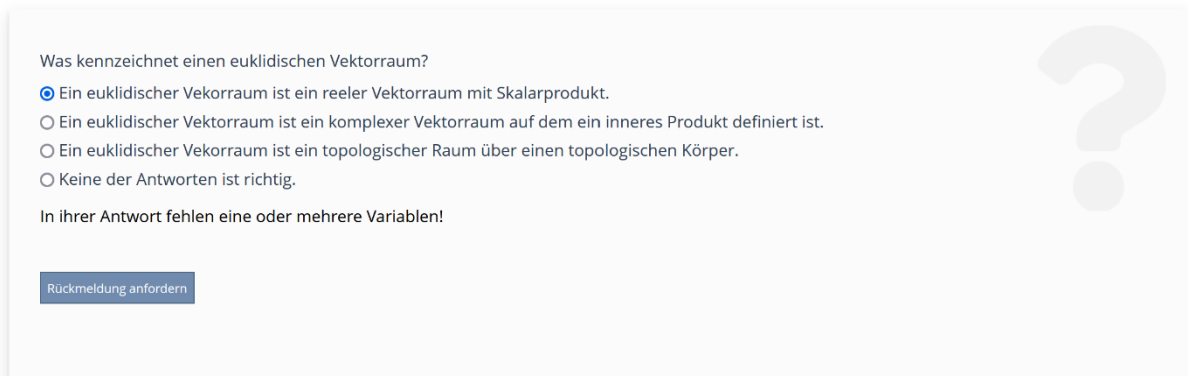
Soll die Antwort bewertet werden, kann dies innerhalb eines Tests über den Reiter **Manuelle Bewertung** vorgenommen werden. Die Manuelle Bewertung ermöglicht es Dozierenden, die entsprechende Frage für alle Teilnehmenden einzusehen und im Nachhinein zu bewerten. Sollten Sie dies planen, kann es hilfreich sein, im Rückmeldebaum darauf hinzuweisen, dass die Bewertung im Nachhinein geplant ist.

Gängige Antwortüberprüfungen: AlgEquiv, String

Radio-Auswahl (Single-Choice Aufgabe)

Mit diesem Eingabetyp können Sie Single-Choice-Aufgaben erstellen. Die Antwortmöglichkeiten werden hierbei untereinander angezeigt.

Eingabetypen: Radio-Auswahl




Die Fragenerstellung und die Antwortüberprüfung funktioniert vollkommen analog zum Eingabetypen [Drop-Down-Liste](#). Schauen Sie sich diesen Abschnitt an, wenn Sie Fragen zu Fragenerstellung oder Antwortüberprüfung haben.

Hinweis: Im Gegensatz zu dem Eingabetyp Drop-Down-Liste sind bei diesem Eingabetyp jedoch Eingaben in LaTeX möglich. Für Antwortoptionen, die LaTeX-Eingaben beinhalten und mit Hilfe von [MathJax](#) gerendert werden sollen, verwenden Sie `"\\(LaTeX-Formel\\)"` als „Anzeigetext“. Achten Sie hierbei darauf bei jedem Befehl „\\“ zu verwenden, statt „\“.

Einheiten

Mit diesem Eingabetypen können Antworten mit physikalischen Einheiten eingegeben und anschließend überprüft werden. Die Musterlösung wird hierbei definiert als *stackunits(Wert, Einheit)*.

 Einheiten werden wie folgt eingegeben: Zahl*Einheit z.B. 10*N oder 3*m*s^(-2). Eine Liste mit den vorhandenen Einheiten und ihrer STACK-Schreibweise finden Sie hier in der Moodle-Beschreibung

Geben Sie 100g in Kilogramm an:

☒

0.1*kg wurde wie folgt interpretiert: 0.1 kg

Die Einheiten in deiner Antwort sind: [kg]

In ihrer Antwort fehlen eine oder mehrere Variablen!

Eine Athletin läuft 1000 Meter in 20 Minuten. Mit welcher Geschwindigkeit ist sie unterwegs?

$v =$ ☒

3000*m/h wurde wie folgt interpretiert: 3000 m/h

Die Einheiten in deiner Antwort sind: [h, m]

In ihrer Antwort fehlen eine oder mehrere Variablen!

Eine Athletin läuft 1000 Meter in 20 Minuten. Mit welcher Geschwindigkeit ist er unterwegs? Geben Sie die Antwort in km/h an

$v =$ ☒

3*km/h wurde wie folgt interpretiert: 3 km/h

Die Einheiten in deiner Antwort sind: [h, km]

In ihrer Antwort fehlen eine oder mehrere Variablen!

[Rückmeldung anfordern](#)

Für die Überprüfung wird der Name des Eingabefeldes (z.B. ans1) in **SA**ns eingetragen. In **TA**ns wird die Lösung eingegeben, die überprüft werden soll. Durch die entsprechenden Antwortüberprüfungen wird es möglich, zu überprüfen, ob die Antwort den richtigen Wert und die richtige Einheit hat. Es ist jedoch auch möglich, physikalische äquivalente Ergebnisse als gleichermaßen richtig zu werden. So ist es beispielsweise möglich, dass 1000m und 1km beides als richtig gewertet werden.

Um diesen Eingabetypen richtig zu verwenden ist wichtig, dass Sie unter **Rückmeldebäume** die **Antwortüberprüfung** auf **Units** stellen. Es gibt verschiedene Antwortüberprüfungen je nachdem, ob Sie den exakten Wert überprüfen möchten (**-SigFig**, **-StrictSigFigs**), eine absolute Toleranz (**-Absolute**, **-StrictAbsolute**) oder relative Toleranz (**-Relative**, **-StrictRelative**) berücksichtigen möchten.

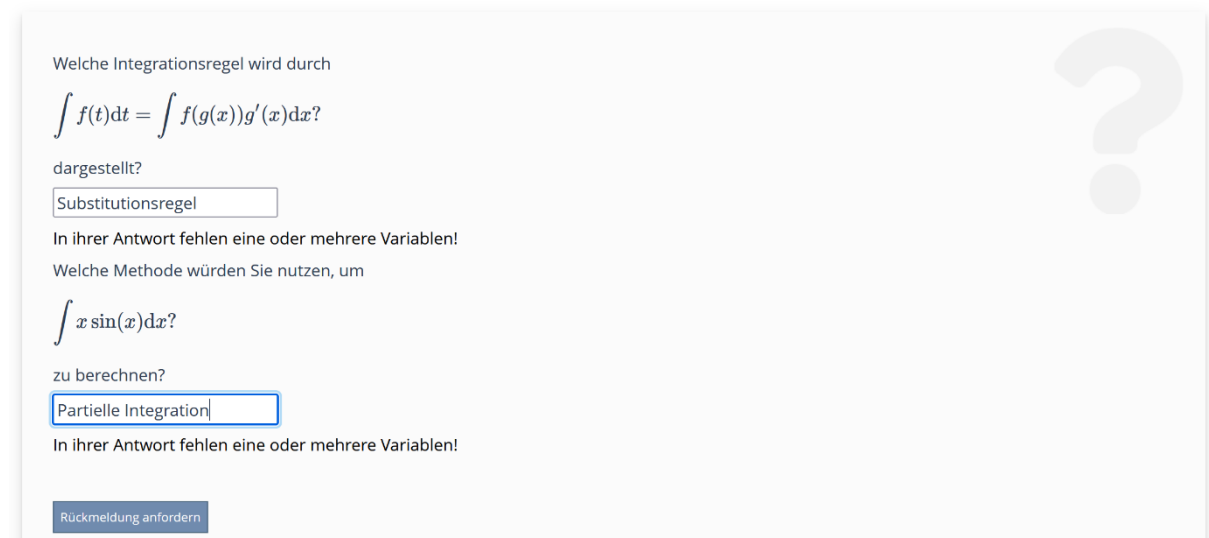
Hinweis: Möchten Sie nur überprüfen, ob die von den Studierenden eingegebene Einheiten mit der gewünschten Einheit übereinstimmen, verwenden Sie `stack_units_units(ans1)` als **SA**ns, `stack_units_units(MusterLoesung)` als **TA**ns und die Antwortüberprüfung AlgEquiv oder String.

Gängige Antwortüberprüfungen: UnitsAbsolute, UnitsStrictAbsolute, UnitsRelative, UnitsStrictRelative

Zeichenkette

Dieser Eingabetyp ermöglicht die Eingabe von Zeichenketten. Im Unterschied zur algebraischen Eingabe sind hier Leerzeichen möglich. Somit ist es möglich, dass Studierende in mehreren Wörtern oder ganzen Sätzen antworten. In die Musterlösung wird die richtige Lösung in Anführungszeichen angegeben.

Eingabetypen: Zeichenkette



Welche Integrationsregel wird durch

$$\int f(t)dt = \int f(g(x))g'(x)dx?$$

dargestellt?

In ihrer Antwort fehlen eine oder mehrere Variablen!

Welche Methode würden Sie nutzen, um

$$\int x \sin(x)dx?$$

zu berechnen?

In ihrer Antwort fehlen eine oder mehrere Variablen!

[Rückmeldung anfordern](#)

Für die Überprüfung wird der Name des Eingabefeldes (z.B. ans1) in **SA**ns eingetragen. In **TA**ns wird die Lösung eingegeben, die überprüft werden soll. Auch hierbei ist es wichtig, dass TAns in Anführungszeichen angegeben wird. Mit Hilfe der Feedback-Variablen und Maxima-Befehlen kann hierbei gut darauf geprüft werden, ob beispielsweise bestimmte Wörter in der Studierendenantwort enthalten sind. Wie das konkret umgesetzt werden kann hängt stark von den individuellen Wünschen bei der Fragerstellung ab. Kontaktieren Sie uns gerne, falls Sie Fragen dazu haben.

Gängige Antwortüberprüfungen: String, StringSloppy

Numerisch

Mit diesem Eingabetypen können in das Eingabefeld ausschließlich Zahlen eingegeben werden. Dies minimiert mögliche Fehlerquellen. In die Musterlösung wird die richtige Zahl eingegeben.

Eingabetypen: Numerisch



Rechnen Sie $3 + 4/2$ aus: ✓

Rechnen Sie $6 - 2 * 2$ aus: ✓

[Rückmeldung anfordern](#)

Für die Überprüfung wird der Name des Eingabefeldes (z.B. ans1) in **SA**ns eingetragen. In **TA**ns wird die Zahl eingegeben, die überprüft werden soll.

Gängige Antwortüberprüfungen: AlgEquiv, NumAbsolute, NumRelative, NumDecPlaces

Weitere Ausführungen zum Zusammenspiel von dem Eingabetyp Numerisch und verschiedenen Antwortüberprüfungen finden Sie unter folgendem Link:



https://github.com/mathsmoodle-gtype_stack/blob/master/doc/en/Authoring/Answer_tests_numerical.md

5.3.6 Antwortüberprüfungen

Antwortüberprüfungen werden in Rückmeldebäumen verwendet, um die Eingabe der Studierenden zu überprüfen und entsprechende Punkte zu vergeben. Dazu werden **SAns** und **TAns** miteinander abgeglichen. Stimmen sie überein, wird im Rückmeldebaum der grüne Pfad weitergegangen, ansonsten der rote, was zu unterschiedlichen Rückmeldungen führt (siehe Abschnitt [Rückmeldebäume \(PRTS\)](#)). Welche Kriterien erfüllt sein müssen, damit **SAns** und TAns als übereinstimmend gelten, wird hierbei durch die **Antwortüberprüfung** festgelegt.

Grundsätzlich sind Eingabetypen und Antwortüberprüfungen unabhängig voneinander, doch bei bestimmten Eingabetypen sind bestimmte Antwortüberprüfungen sinnvoll. Die gängigen Antwortüberprüfungen zu den Eingabetypen finden Sie im Abschnitt [Eingabetypen](#). Im Folgenden beschreiben wir die wichtigsten Antwortüberprüfungen und deren Funktionen. Zusätzlich verlinken wir zu jeder Antwortüberprüfung die ausführliche Erklärung des STACK-Erfinders Chis Sangwin.

Eine vollständige Liste mit den Antwortüberprüfungstypen und deren Funktionsweisen finden Sie unter:



https://github.com/mathsmoodle-gtype_stack/blob/master/doc/en/Authoring/Answer_tests.md

AlgEquiv

AlgEquiv überprüft, ob SAns und TAns algebraisch Äquivalent zueinander sind. Das werden SAns und TAns zunächst vereinfacht und anschließend miteinander verglichen. Bei mathematischen Aussagen spielt die Schreibweise somit keine Rolle. Es ist lediglich relevant, ob die Aussagen algebraisch miteinander übereinstimmen.

Diese Antwortüberprüfung kann für vielfältige mathematische Objekte wie numerische Eingaben, Terme mit Variablen, Gleichungen, Ungleichungen, Listen und Mengen verwendet werden. Er eignet sich jedoch auch um Zeichenketten (Strings) auf exakte Übereinstimmung zu überprüfen.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-gtype_stack/blob/master/doc/en/Authoring/Answer_tests.md#algequiv-algequiv

EqualComAss

EqualComAss überprüft, ob SAns und TAns bis auf Kommutativität (Vertauschung der Ausdrücke) und Assoziativität (Klammersetzung) mathematisch miteinander übereinstimmen. So werden beispielsweise $x+x$ und $2*x$ nicht als gleich angesehen, was bei der AlgEquiv noch

als gleichwertig gelten würde. Dadurch bietet diese Antwortüberprüfung weniger Spielraum für die Eingabe von Studierenden, als dies bei AlgEquiv der Fall ist.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Answer_tests.md#equalcomass

EquivReasoning

EquivReasoning wird in Verbindung mit dem Eingabetyp Gleichungen verwendet. Diese Antwortüberprüfung prüft, ob einzelnen Zeilen der Studierendenantwort algebraisch äquivalent zueinander sind. Diese Antwortüberprüfung ermöglicht es dadurch Studierenden anzuzeigen, an welcher Stelle der Antwort sie einen Fehler in der Umformung gemacht haben. Um die dadurch automatisch generierte Rückmeldung anzuzeigen ist es wichtig, dass im entsprechenden Knoten des Rückmeldebaumes **Feedback unterdrücken** auf *Nein* steht.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/CAS/Equivalence_reasoning.md#answer-tests

EquivFirst

EquivFirst wird genau wie EquivReasoning in Verbindung mit dem Eingabetyp Gleichungen verwendet. Zunächst wird überprüft, ob alle Zeilen der Studierendenantwort algebraische äquivalent zueinander sind. Anschließend wird überprüft, ob die erste Zeile von SAns bis auf Kommutativität (Vertauschung) und Assoziativität (Klammersetzung) mit TAns übereinstimmt.

Um zu überprüfen, ob die letzten Zeilen miteinander übereinstimmen können Sie die Antwortüberprüfung AlgEquiv oder EqualComAss und den Maxima-Befehl *last(Liste)* verwenden. Tragen Sie dazu beispielsweise *last(ans1)* in SAns ein und *last(Musterlösung)* in TAns.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/CAS/Equivalence_reasoning.md#answer-tests

Num-GT(E)

Num-GT (Num-GTE) überprüft, ob **SAns** größer als (oder gleich zu) **TAns** ist.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Answer_tests_numerical.md#gt--gte

NumAbsolute

NumAbsolut überprüft, ob SAns und TAns innerhalb einer gewissen Toleranz miteinander übereinstimmen. Dabei wird in die Test-Option eingetragen, wie weit SAns von TAns voneinander abweichen dürfen. Gilt $|\mathbf{SAns} - \mathbf{TAns}| < \mathbf{Test-Option}$ wird die Überprüfung als wahr ausgewiesen. Die maximal erlaubte Abweichung wird also als absolute Zahl angegeben.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Answer_tests_numerical.md#numrelative--numabsolute

NumRelative

NumRelativ überprüft ebenfalls, ob SAns und TAns innerhalb einer gewissen Toleranz miteinander übereinstimmt. Jedoch ist hier die relative Abweichung von TAns entscheidend. In den Test-Optionen wird eingetragen, wie weit SAns und TAns voneinander abweichen dürfen. Gilt $|\mathbf{SAns} - \mathbf{TAns}| < \mathbf{Test-Option} * |\mathbf{TAns}|$ wird die Überprüfung als wahr ausgewiesen. Die Test-Option ist hierbei der prozentuelle Anteil (angegeben als Dezimalzahl zwischen 0 und 1), den **SAns** von **TAns** abweichen darf.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Answer_tests_numerical.md#gt--gte

NumDecPlaces

NumDecPlaces überprüft, ob SAns die gewünschte Anzahl Nachkommastellen hat. Wie viele Nachkommastellen gefordert werden, wird in der Test-Option festgelegt. Hat SAns nicht die passende Anzahl Nachkommastellen wird eine entsprechende Fehlermeldung gegeben. Zusätzlich wird überprüft, ob SAns numerisch mit TAns übereinstimmt. Dazu werden die Zahlenwerte zunächst auf die in der Test-Option eingestellte Anzahl Nachkommastellen gerundet und anschließend miteinander verglichen. Um die automatisch generierte

Rückmeldung anzuzeigen ist es wichtig, dass im entsprechenden Knoten des Rückmeldebaumes **Feedback unterdrücken** auf *Nein* steht.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Answer_tests_numerical.md#numdecplaces

Units(Strict)SigFigs

UnitsSigFigs wird in Kombination mit dem Eingabetyp Einheiten verwendet. Deswegen ist es für diese Antwortüberprüfung wichtig, dass TAns auch Einheiten beinhaltet. Dabei ist es auch möglich, TAns mit Hilfe der Funktion *stackunits(Wert, Einheit)* einzugeben.

Mit UnitsSigFigs wird überprüft, ob SAns in der gewünschten Genauigkeit angegeben wurde. Wie viele signifikante Stellen SAns haben soll, wird hierbei in die Test-Option eingetragen. Zudem wird überprüft, ob der SAns unabhängig von den Einheiten äquivalent zu TAns ist, indem die Antwort umgeformt wird. So sind beispielsweise 1 km das gleiche wie 1000 m . Sind die Einheiten von SAns nicht passend, wird also beispielsweise eine Leistung angegeben, obwohl eine Kraft gefordert wurde, wird ein entsprechender automatisch generierter Hinweis gegeben.

Mit dem Zusatz **-Strict** muss **SAns** in derselben Einheit wie **TAns** angegeben sein. Dies ermöglicht, dass nur Eingaben in der richtigen Einheit als richtig gewertet werden. Ohne diesen Zusatz werden alle zu **TAns** äquivalenten Angaben als richtig bewertet.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Units.md#answer-tests

UnitsStrictAbsolute

UnitsStrictAbsolut wird in Kombination mit dem Eingabetyp Einheiten verwendet. Deswegen ist es für diese Antwortüberprüfung wichtig, dass TAns auch Einheiten beinhaltet. Dabei ist es auch möglich, TAns mit Hilfe der Funktion *stackunits(Wert, Einheit)* einzugeben.

UnitsStrictAbsolut überprüft, ob SAns und TAns im Rahmen einer Toleranz miteinander übereinstimmen. Die Höhe der Toleranz wird im Rückmeldebaum in die Test-Optionen eingetragen. Der eingegebene Wert ist hierbei eine absolute Toleranz. Die SAns darf daher, analog zu NumAbsolute in beide Richtungen maximal um diesen Wert von TAns abweichen.

Mit dem Zusatz **-Strict** muss SAns in derselben Einheit wie TAns angegeben sein. Dies ermöglicht, dass nur Eingaben in der richtigen Einheit als richtig gewertet werden. Ohne diesen Zusatz werden alle zu **TAns** äquivalenten Angaben als richtig bewertet.

Hinweis: Von der Antwortüberprüfung UnitsAbsolute (ohne den Zusatz Strict) raten wir ab, weil die Toleranz auf die Einheit von SAns gerechnet wird. Dies führt dazu, dass abhängig davon, welche Einheit die Studierenden wählen, unterschiedlich große Abweichungen in Bezug auf die Einheit der TAns möglich sind. Dies führt dazu, dass die Eingaben verschiedener Studierender möglicherweise nicht mehr vergleichbar sind. Die Antwortoption UnitsStrictAbsolute hingegen können wir sehr empfehlen.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Units.md#answer-tests

Units(Strict)Relative

UnitsRelative wird in Kombination mit dem Eingabetyp Einheiten verwendet. Deswegen ist es für diese Antwortüberprüfung wichtig, dass TAns auch Einheiten beinhaltet. Dabei ist es auch möglich, TAns mit Hilfe der Funktion *stackunits(Wert, Einheit)* einzugeben.

UnitsRelative überprüft, ob SAns und TAns im Rahmen einer Toleranz miteinander übereinstimmen. Die Höhe der Toleranz wird im Rückmeldebaum in die Test-Optionen eingetragen. Der eingegebene Wert ist hierbei eine relative Toleranz. Diese wird als Wert zwischen 0 und 1 angegeben, wobei die 0 keine und die 1 eine 100% Abweichung toleriert. Die SAns darf daher, analog zu NumRelative in beide Richtungen maximal um diesen Wert von TAns abweichen.

Mit dem Zusatz **-Strict** muss SAns in derselben Einheit wie TAns angegeben sein. Dies ermöglicht, dass nur Eingaben in der richtigen Einheit als richtig gewertet werden. Ohne diesen Zusatz werden alle zu **TAns** äquivalenten Angaben als richtig bewertet.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Units.md#answer-tests

Diff

Diff überprüft, ob SAns und TAns algebraisch miteinander übereinstimmen und gibt automatisches Feedback, falls SAns den Anschein macht, als wurde integriert anstellen von differenziert. Wir finden es für Differenzationsaufgaben schöner, die Antwortüberprüfung AlgEquiv zu wählen, in SAns den Platzhalter für die Studierendenantwort (z.B. ans1) einzutragen und in TAns die Maxima Funktion **diff(F(x), x)** zu verwenden, wobei F(x) die Funktion ist, die differenziert werden sollte und x die Variable nach der differenziert werden soll. So ist es möglich mit Zufallswerten zu arbeiten und dennoch gezieltes Feedback zu geben.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-gtype_stack/blob/master/doc/en/Authoring/Answer_tests.md#calculus-diff

Int

Int ist sehr voraussetzungsreich in der Anwendung. Wir gehen hier nicht näher auf die Funktionsweise dieser Antwortüberprüfung ein, denn wir finden es für Integrationsaufgaben schöner, die Antwortüberprüfung AlgEquiv zu wählen, in SAns den Platzhalter für die Studierendenantwort (z.B. ans1) einzutragen und in TAns die Maxima Funktion **int(f(x), x)** zu verwenden, wobei f(x) die Funktion ist, die integriert werden sollte und x die Variable nach der integriert werden soll. So ist es möglich mit Zufallswerten zu arbeiten und dennoch gezieltes Feedback zu geben.

Eine Erklärung der Antwortüberprüfung Int finden Sie unter:



https://github.com/mathsmoodle-gtype_stack/blob/master/doc/en/Authoring/Answer_tests.md#calculus-int-int

String

String wird verwendet, um Zeichenketten Zeichen für Zeichen miteinander zu vergleichen. abzugleichen. Nur wenn alle Zeichen (inkl. Leerzeichen sowie Groß- und Kleinschreibung) in SAns und TAns miteinander übereinstimmen, gibt diese Antwortüberprüfung zurück, dass SAns und TAns miteinander übereinstimmen.

Hiermit lassen sich auch algebraische Eingaben überprüfen, wenn wichtig ist, dass alle Zeichen übereinstimmen. Besonders gut eignet sich diese Antwortüberprüfung jedoch für den Eingabetyp Zeichenkette, da dabei Antworten aus mehreren Wörtern bestehen und dementsprechend Leerzeichen enthalten können.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/c1040ea3f640871ad1239d247145111b22923e5c/doc/en/Authoring/Answer_tests.md#string-match

StringSloppy

StringSloppy funktioniert analog zu String. Der Unterschied besteht darin, dass alle Groß- und Kleinschreibung, sowie Leerzeichen nicht beachtet werden.

Eine weitere Erklärung finden Sie unter:



https://github.com/mathsmoodle-qtype_stack/blob/c1040ea3f640871ad1239d247145111b22923e5c/doc/en/Authoring/Answer_tests.md#string-match

5.3.7 Zufallswerte

Zufallswerte stellen eine wichtige Komponente der STACK-Frage dar, denn sie können verwendet werden, um Abschreiben zu verhindern und eine Wiederholung zu Übungszwecken zu ermöglichen.

Der häufigste Einsatz für Zufallswerte ist das Ersetzen von festen Zahlen durch Zufallszahlen. Doch auch nicht-mathematische Anwendungen sind möglich. Zufallswerte können neben Zahlen auch Funktion oder Zeichenkette (z.B. Wörter) sein.

Die Zufallswerte werden durch [Maxima](#) bestimmt und für jeden Testdurchlauf und jeden User neu festgelegt. Wird ein Testdurchlauf unterbrochen, bleiben die Zufallswerte demnach gleich. Wird der Test beendet und ein neuer Testdurchlauf gestartet, werden die Zufallswerte neu festgelegt.

Fragen-Variablen definieren

Fragen-Variablen werden mit Hilfe von Doppelpunkten definiert. Sie können festgelegte Werte (z.B. Var1:5) oder eine Zufallsfunktion (z.B. Var1:rand(9)) beinhalten. Außerdem ist es möglich, ganze Aufgaben (z.B. A1: x^3+3x^2) oder Lösungen (z.B. Loes1:diff(A1)) in den Fragen-Variablen zu definieren. Dies kann hilfreich sein, um alle für die Aufgabe relevanten Informationen an einer Stelle zu haben. Insbesondere erleichtert dies die Abwandlung von Fragen.

Fragen-Variablen

Dieses Feld erlaubt es, CAS-Variablen zu definieren und zu verändern, z.B. um Zufallsversionen zu ermöglichen. Diese Variablen sind in allen anderen Teilen der Aufgabe verfügbar.
Inhalt: CAS-Ausdruck

Achtung: Es ist wichtig, darauf zu achten, dass die Fragen-Variablen in der richtigen Reihenfolge definiert werden. Soll eine Variable weiterverwendet werden, muss sie zunächst in einer der darüber liegenden Zeilen definiert worden sein.

Hinweis: Variablennamen können aus Buchstaben und Zahlen bestehen. Wichtig ist dabei jedoch, dass auf Zahlen keine weiteren Buchstaben folgen. So ist beispielsweise *Var1a* kein gültiger Variablen-Name.

Hinweis: Grundsätzlich gibt es zwei unterschiedliche Arten, Variablen aufzurufen. In Feldern, in denen Maxima-Code verwendet werden darf (Fragen-Variablen, Musterlösung, Feedback-Variablen, SAns, TAns) wird einfach nur der Variablenname eingegeben. In Textfeldern (Fragetext, Aufgabenhinweis und alle Rückmeldungsfelder) müssen die definierten Variablen über {@Variablennamen@} aufgerufen werden.

Zufallswerte definieren

Zufallswerte werden im Feld **Fragen-Variablen** definiert. Schreiben Sie dazu *Variablenname:Zufallsfunktion*.

Fragen-Variablen

Var1:rand(9)

Dieses Feld erlaubt es, CAS-Variablen zu definieren und zu verändern, z.B. um Zufallsversionen zu ermöglichen. Diese Variablen sind in allen anderen Teilen der Aufgabe verfügbar.
Inhalt: CAS-Ausdruck

Zufallsfunktionen

Die meisten Zufallswerte können mit Hilfe einiger weniger Befehle erzeugt werden.

- **rand(n)**
 - **n ist ganze Zahl:** Erzeugt eine zufällige ganze Zahl zwischen 0 und n-1
 - z.B. Var1:rand(9) ergibt eine ganze Zahl zwischen 0 und 8
 - **n Dezimalzahl:** Erzeugt eine zufällige Dezimalzahl zwischen 0 und n
 - z.B. Var2:rand(5.3) gibt eine rationale Zahl zwischen 0 und 5.3
- **rand_with_step(Start,Ende,Schrittweite)**
 - gibt eine zufällige Zahl aus der Liste [Start, Start+Schrittweite, ..., Ende] aus. Wird das Ende nicht exakt mit der Schrittweite erreicht, wird die obere Grenzen nicht mit eingeschlossen.
 - z.B. Var3:rand_with_step(1,7,1.5) gibt eine Zahl aus [1,2.5,4,5.5,7] (Die obere Grenze ist mit eingeschlossen.)
 - z.B. Var4:rand_with_step(1,6,1.5) gibt eine Zahl aus [1,2.5,4,5.5] (Die obere Grenze ist nicht mit eingeschlossen, da sie nicht durch die gewählte Schrittweite erreicht wird und die nächst höhere Zahl außerhalb des Bereiches liegt.)
- **rand_with_prohib(Start,Ende,[c,d,e,...])**
 - gibt eine zufällige Zahl aus der Liste [Start, Start+1, ..., Ende] ohne die Elemente [c, d, e, ...] aus.
 - z.B. Var5:rand_with_prohib(1,6,[3,4]) ergibt eine Zahl aus [1,2,5,6]
- **rand([liste])**
 - gibt ein zufällig ausgewähltes Element aus der gegebenen Liste aus.
 - z.B. Var6:rand([a,3,4,t,x^2]) ergibt ein Element aus [a,3,4,t,x^2]
 - z.B. Var7:rand([cat,dog,elefant]) ergibt eines der Worte

Eine Liste mit Erklärungen der Zufallszahlenfunktionen in Maxima gibt es unter:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/CAS/Random.md

(Hinweis: Dieser Link bezieht sich auf STACK in moodle. Vieles davon kann jedoch direkt übernommen werden, da es bei STACK in ILIAS genauso funktioniert. Dies trifft jedoch nicht auf alles zu.)

Zufallswerte verwenden

Nachdem die Zufallsvariablen definiert sind, können sie über {@Variablenname@} in den Fragetext eingefügt werden.

Zudem ist es wichtig, dass alle definierten Zufallswerte über {@Variablenname@} in das Feld **Aufgabenhinweis** eingetragen werden.

Aufgabenhinweis (notwendig)

Der Aufgabenhinweis wird in einem Test gemeinsam mit der bearbeiteten Aufgabe gespeichert.

Aufgabenhinweis

Mit dem Aufgabenhinweis soll zwischen den verschiedenen zufälligen Versionen einer Frage unterschieden werden. Zwei Fragen sind gleich, genau dann wenn die Aufgabenhinweise gleich sind. Für die spätere Analyse ist es hilfreich sinnvolle Antworthinweise zu erstellen.
Inhalt: CAS-Text

Deswegen ist es wichtig, dass wirklich jeder Zufallswert dort aufgerufen wird. So können Sie nachvollziehen, welche Zufallswerte die Studierenden in der Aufgabe erhalten haben, also welche Aufgabenversion sie angezeigt bekamen. Dies ist wichtig, falls die Studierenden nach einem Test mit Ihnen über die Aufgaben sprechen möchte und um Fehler nachvollziehen zu können.

Tipp: Wir empfehlen die einzelnen Variablenamen über *Variablenname* ist {@Variablenname@} in den Aufgabenhinweis einzutragen. Enthält eine Aufgabenstellung mehrere Zufallsvariablen, kann es auch hilfreich sein, die Aufgabestellung hier einzufügen.

Frage *

Rechnen Sie $3+{@Var1@}/2$ aus:

Pfad: p

Bearbeitungsdauer Stunden: Minuten: Sekunden:

Punkt(e) 1

Fragen-Variablen Var1:rand(9)

Aufgabenhinweis Var1 ist {@Var1@}, Deshalb ist die angezeigte Funktion $3+{@Var1@}/2$

Vorbereitete Zufallswerte (optional)

Vorbereitete Zufallswerte können bei der Verwendung von Zufallswerten benutzt werden, um die Auswirkung unterschiedlicher Zufallswerte auf die Aufgabe zu betrachten. Zudem können sie dazu genutzt werden, in einem Test nur Zufallswerte bzw. Kombinationen aus Zufallswerten anzuzeigen, die Sie für sinnvoll erachten.

Unter dem Reiter **Einstellungen** finden Sie den Abschnitt **Vorbereitete Zufallswerte**.

[Fragenpool für Tests](#)
[Seite bearbeiten](#)
[Vorschau](#)
[Eigenschaften](#)
[Rückmeldung](#)
[Statistik](#)

[Frage bearbeiten](#)
[Bewertung](#)
[Vorbereitete Zufallswerte](#)
[Testfälle](#)
[Fragen aus MoodleXML importieren](#)
[Export nach MoodleXML](#)

NEUEN ZUFALLSWERT ERZEUGEN

Zufallswert

[Neuen Wert erzeugen](#)

Zufallswert	Fragen-Notiz für den Zufallswert	Aktionen
Keine Einträge		

Das Feld **Neuen Wert erzeugen** generiert einen Satz Zufallswerte und somit eine Version der Aufgabe. So können Sie mehrere mögliche Versionen der Aufgabe erzeugen. Alle so erzeugte Versionen werden anschließend in einer Tabelle dargestellt.

Zufallswerte, die Ihnen nicht passend erscheinen, können Sie wieder löschen. Dies bewirkt, dass in einem Test nur Zufallswerte angezeigt werden, die sich für sinnvoll erachten.

Der Tabelleneintrag *Fragen-Notiz für den Zufallswert* gibt hierbei den Aufgabenhinweis für den entsprechenden Satz an Zufallswerten wieder. Somit ist ein sinnvoller Aufgabenhinweis auch für die Verwendung von vorbereiteten Zufallswerten von großer Bedeutung. Haben Sie dort

die Aufgabenstellung und die Lösung eingegeben, können Sie hier direkt sehen, wie sich unterschiedliche Zufallswerte auf Ihre Frage auswirken.

Hinweis: Der Aufgabenhinweis dient dazu, vorbereitete Zufallswerte zu unterscheiden. Zwei vorbereitete Zufallswerte gelten als gleich, wenn der Aufgabenhinweis gleich ist. Gleiche Werte werden nicht doppelt angelegt. Dies kann bei wenigen möglichen Zufallswerten dazu führen, dass ein Klicken auf **Neuen Wert erzeugen** keinen neuen Wert in die Tabelle hinzufügt. Sollte dies passieren, erzeugen Sie einfach erneut einen Wert. Auch hierfür ist es wichtig, dass der Aufgabenhinweis aussagekräftig ist und alle Zufallswerte enthält.

Achtung: Sobald Sie Vorbereitete Zufallswerte angelegt haben, wird in einem Test nur noch aus diesen ausgewählt. Möchten Sie, dass wieder alle Kombinationen der Zufallswerte möglich sind, müssen Sie die vorbereiteten Zufallswerte wieder entfernen.

5.3.8 Maxima

Maxima ist das Computer Algebra System (CAS), welches in STACK verwendet wird. Maxima wird unter anderem zum Testen auf algebraische Äquivalenz und zur Verwendung von Zufallswerten eingesetzt. Doch auch komplexere Anwendungen sind mit Maxima möglich. Besonders für mathematische Fragen bietet Maxima viele Möglichkeiten.

STACK ist auch ohne Kenntnisse in Maxima verwendbar, doch Maxima erweitert den Einsatzmöglichkeiten von STACK erheblich. Deswegen ist es hilfreich, grundlegende Befehle aus Maxima zu kennen. Wie auch bei jeder anderen Programmiersprache ist das Internet hierbei ein sehr guter Lehrmeister.

Hilfreiche Links hierzu sind:



[https://de.wikipedia.org/wiki/Maxima_\(Computeralgebrasystem\)](https://de.wikipedia.org/wiki/Maxima_(Computeralgebrasystem))

<http://maxima.sourceforge.net/docs/manual/maxima.html>

https://maxima.sourceforge.io/docs/manual/maxima_369.html

https://github.com/mathsmoodle-qtype_stack/tree/master/doc/en/CAS

(*Hinweis:* Dieser Link bezieht sich auf STACK in moodle. Vieles davon kann direkt übernommen werden, da es bei STACK in ILIAS sehr ähnlich funktioniert, doch das trifft nicht auf alles zu.)

Funktionen aus Maxima-Paketen sind allerdings nur begrenzt nutzbar, einige können jedoch zusätzlich installiert werden. [Schreiben Sie uns gerne eine Mail](#), falls Sie Funktionen aus Maxima-Paketen verwenden möchten, die noch nicht installiert sind.

Nicht jede Maxima-Version funktioniert gleich, die Unterschiede sind jedoch in der Regel gering. Sollte es für Sie wichtig sein, die aktuell verwendete Maxima-Version zu kennen, [schreiben Sie uns gerne eine Mail](#).

Eingabe in STACK: In STACK müssen Maxima-Befehle einzeilig (d.h. ohne Enter) geschrieben werden, weil am Ende jeder Zeile automatisch ein Semikolon „;“ eingefügt wird. Dieses Zeichen gibt in Maxima das Ende eines Befehls an und muss normalerweise nach jeder Eingabe explizit eingegeben werden. Dies ist in STACK nicht zwingend notwendig.



Unter <http://maxima.cesga.es/> können Ausdrücke zur Probe eingegeben werden.

Hinweis: Hier müssen Sie nach jeder Zeile ein Semikolon „;“ einfügen, um einzelne Befehle voneinander zu trennen.

Grundlegende Befehle

Im Folgenden sind die Maxima-Befehle aufgeführt, welche in den Musterlösungen der Übungsaufgaben verwendet wurden. Für eine umfangreichere Dokumentation können Sie den folgenden Link nutzen:

Maxima-Einführung für STACK-Nutzer*innen:



https://github.com/mathsmoodle-qtype_stack/tree/master/doc/en/CAS

Sollten Sie die Erklärung zu einem bestimmten Befehl vermissen, können Sie uns dies gerne [per Mail mitteilen](#).

- Der Doppelpunkt „:“ wird genutzt, um einer Variablen einen Wert zuzuweisen. Zum Beispiel um der Variable n den Wert 5 zuzuweisen, nutzen wir die Syntax „ $n:5$ “.
- Doppelpunkt-Gleich „:=“ wird verwendet, um eine Funktion zu definieren. Der Unterschied zu einer Variablen ist, dass in einer Funktion allgemeine Variablen enthalten sein können, die später beim Aufruf der Funktion ersetzt werden. Weitere Informationen finden Sie im Abschnitt [Hinweise zur Erstellung eigener Funktionen](#). Beispiele finden Sie im Abschnitt [Workarounds](#).
- **diff(<expr>, <x>)** differenziert die Funktion <expr> nach <x>.
- **int(<expr>, <x>)** integriert die Funktion <expr> nach <x>.
- **is (<expr>)** prüft, ob eine Aussage <expr> wahr oder falsch ist und gibt entsprechend true oder false aus.
- **and** ist der logische Operator „und“. Hiermit können beispielsweise durch **is (<expr>)** erzeugte Aussagen miteinander verknüpft werden.
- **simp** ist der Befehl für Vereinfachung. Mit simp:true bzw. simp:false kann die Vereinfachung an bzw. aus gestellt werden.
- **lmax(L)** gibt den maximalen Wert der Liste L zurück.
- **lmin(L)** gibt den minimalen Wert der Liste L zurück.
- Die **genmatrix** Funktion generiert mit Hilfe einer angegebenen Funktion eine Matrix. **gematrix(f(n,m),n,m)** bedeutet, dass eine Matrix mit n Zeilen und m Spalten generiert wird. Die Funktion f entscheidet darüber, wie die einzelnen Einträge belegt werden. Damit dies möglich ist muss f von n und m abhängen, denn diese beiden Werte werden an die Funktion f weitergegeben. f könnte beispielsweise so aussehen: $f(m,n):=n^2-m^4$.

Hinweis: Einträgen mit gleichen Indizes wird immer der gleiche Werte zugewiesen werden. Dies gilt auch wenn $f(m,n):=\text{rand}(9)$. Dies kommt dadurch zustande, dass der Seed für die Zufallszahlen zum Zeitpunkt der Definition der Funktion festgelegt wird. Dieser Seed wird ab dann verwendet und führt bei gleichen Argumenten zu gleichen

Ergebnissen. Die Funktion muss also neu definiert werden, wenn eine weitere Matrix mit neuen Zufallszahlen generiert werden soll.

Eine Anwendung dieser Funktion finden Sie in den Beispielen unter [08 Matrizen](#).

STACK-spezifische Maxima-Befehle

- **plot([f1,f2,...],[x,x0,x1],[y,y0,y1])** gibt einen Plot der Funktionen f1, f2,... aus, welcher zwischen x0 und x1 bzw. y0 und y1 dargestellt wird.

In den Übungsaufgaben gibt es in der Frage [19 Plot-Optionen](#) weitere Optionen zum Anschauen.

- **solve(<expr>)** löst die Gleichung <expr>.
- **float(<expr>)** wandelt <expr> in eine Dezimalzahl („Fließkommazahl“) um.
- **round(<expr>*10^n)/10^n** rundet <expr> auf n Nachkommastellen.

- Beispiel:

```
x:%pi;
```

```
n:3;
```

```
u:float(round(x*10^n)/10^n);
```

Workarounds

In diesem Abschnitt finden Sie von uns geschriebene Funktionen, die kopiert und verwendet werden können. Kopieren Sie die entsprechende Funktion an den Anfang des Feldes *Fragen-Variablen*, um sie in der gesamten Aufgabe aufrufen und verwenden zu können.

Die folgenden Funktionen sind zur Anschauung in Aufgaben eingebaut. Diese Aufgaben finden Sie im Fragenpool für **Workarounds mit Maxima-Befehlen** unter:



https://www.selfassessment.uni-koeln.de/goto.php?target=qpl_1945&client_id=uzk_selfea

- **Runden**

Diese Funktion rundet eine Zahl ZTV auf ZTN Nachkommastellen.

- $\text{round2}(\text{ZTV}, \text{ZTN}) := (\text{if } \text{ZTV} * 10^{\text{ZTN}} - \text{floor}(\text{ZTV} * 10^{\text{ZTN}}) < 0.5 \text{ then } \text{float}(\text{floor}(\text{ZTV} * 10^{\text{ZTN}}) * 10^{-\text{ZTN}}) \text{ else } \text{float}(\text{floor}(\text{ZTV} * 10^{\text{ZTN}} + 1) * 10^{-\text{ZTN}}))$
- Beispiel: $\text{round2}(4.5678, 2) = 4.57$

Die Maxima-Funktion *round()* ermöglicht das Runden ebenfalls. Jedoch verwendet diese Funktion das symmetrische Runden. Die Funktion *round2()* ermöglicht kaufmännisches Runden, wie man es in den meisten Fällen verwenden möchte.

- **Trennzeichen in große Zahlen einfügen**

Diese Funktion fügt Leerzeichen als Trennzeichen in große Zahlen ein. So wird die Ansicht für Studierende übersichtlicher. Die Zahl wird hierbei in einen String umgewandelt. Deswegen ist es notwendig für Rechnungen die ursprüngliche Zahl zu verwendet, falls mathematische Eigenschaften überprüft werden sollen. Diese Funktion dient lediglich der Darstellung.

- $\text{trennzeichen}(\text{zahl}) := (\text{if } \text{numberp}(\text{zahl}) = \text{true} \text{ then } \text{zahl}:\text{string}(\text{zahl}), \text{if } \text{ssearch}(".", \text{zahl}) = \text{false} \text{ then } (l:\text{length}(\text{zahl}), \text{while } l > 3 \text{ do } (l:(l-3), (\text{zahl}:\text{insert}(".", \text{zahl}, l+1)))) \text{ else } (\text{nachkommastellen}:\text{sconcat}(".", \text{part}(\text{split}(\text{zahl}, "."), 2)), \text{vorkommastellen}:\text{part}(\text{split}(\text{zahl}, "."), 1), l:\text{length}(\text{vorkommastellen}), \text{while } l > 3 \text{ do } (l:(l-3), (\text{vorkommastellen}:\text{insert}(".", \text{vorkommastellen}, l+1))), \text{zahl}:\text{sconcat}(\text{vorkommastellen}, \text{nachkommastellen})), \text{zahl})$
- Beispiel: $\text{trennzeichen}(398675643) = 398\ 675\ 643$

- **Kreuzprodukt**

Diese Funktion berechnet das Kreuzprodukt von zwei Vektoren **u** und **v**. Dabei ist es wichtig, dass die Vektoren als Liste angegeben werden.

- $\text{cross}(u, v) := [u[2] * v[3] - v[2] * u[3], v[1] * u[3] - u[1] * v[3], u[1] * v[2] - v[1] * u[2]]$

Hinweise zur Erstellung eigener Funktionen

Funktionen werden mittels `:=` definiert.



Wir empfehlen, die Funktionen unter <http://maxima.cesga.es/> zu schreiben und zu testen.

Denken Sie hierbei daran, ein Semikolon „`;`“ nach jeder Zeile einzufügen. Um die Funktion anschließend in STACK zu verwenden ist es wichtig, dass Sie alle Zeilenumbrüche entfernen und nur die Semikolons „`;`“ stehen lassen.

Zur Verwendung von Schleifen: Um **Schleifen** in STACK verwenden zu können, müssen sie immer auf eine Variable geschrieben werden.

`for i:1 thru 6 do z:z*2;` funktioniert nicht, aber

`x:for i:1 thru 6 do z:z*2;` funktioniert.



Grundlegende Informationen zu Schleifen finden Sie unter:
[https://de.wikipedia.org/wiki/Schleife_\(Programmierung\)](https://de.wikipedia.org/wiki/Schleife_(Programmierung))

5.3.9 MathJax

MathJax ermöglicht eine anschauliche Darstellung der Frage- und Feedbacktexte mit Hilfe ausgewählter LaTeX-Befehle. MathJax kann in STACK verwendet werden, um beispielsweise Formeln ästhetisch ansprechend anzuzeigen.



<http://docs.mathjax.org/en/latest/>

Allerdings sind dabei nur die LaTeX-Befehle verwendbar, die verwendet werden, um Formeln darzustellen. LaTeX-Befehle, die darüber hinaus die Ansicht gestalten (wie z.B. `\vspace`) können nicht verwendet werden. Verwenden Sie stattdessen [HTML](#) um den Fragetext oder das Feedback zu gestalten.

LaTeX lernen Sie am besten, indem Sie sich bereits existierende Dokumente mit Erläuterungen anschauen oder indem Sie die Befehle mit Hilfe einer Online-Suchmaschine herausfinden. Für eine umfangreiche Nutzung von STACK sind Basis-Kenntnisse in der Formelgestaltung mit LaTeX bereits ausreichend.

Der TeX-Knopf im Kopf des Eingabefeldes **Frage** fügt an der entsprechenden Stelle den Formelbefehl ein. Dort sehen Sie in einer Vorschau auch, wie die Formel den Studierenden angezeigt werden wird. So können Sie verschiedene Eingaben ausprobieren, bis Ihnen die Anzeige gefällt.

Neben dem TeX-Knopf ist es auch möglich LaTeX-Befehle einzugeben, indem man in dem Fragetext eine Zeichenkombination für Beginn und Ende des LaTeX-Befehls (Mathmode) verwendet, die sogenannten Parser. Davon gibt es verschiedene. Inline-Eingaben in LaTeX beginnt man mit „`\(`“ und beendet sie mit „`\)`“. Newline-Eingaben in LaTeX beginnt man mit „`\[`“ und beendet sie mit „`\]`“.

Angewendet könnte das beispielsweise so aussehen: `\(3+\frac{4}{2}\)`, was in der Ausgabe ungefähr so aussieht: $3 + \frac{4}{2}$

LaTeX-Befehle lassen sich sehr gut durch Googlen herausfinden. Viele wichtige LaTeX-Befehle finden Sie zudem unter den folgenden Links:



[https://de.wikibooks.org/wiki/LaTeX-Kompodium: F%C3%BCr Mathematiker](https://de.wikibooks.org/wiki/LaTeX-Kompodium:_F%C3%BCr_Mathematiker)

<http://www.physki.de/PhysKi/index.php?title=MathJax-Formelsatz>

https://de.wikipedia.org/wiki/Liste_mathematischer_Symbole

Grundlegende Befehle

Im Folgenden sind die MathJax-Befehle angeführt, welche in den Musterlösungen der Übungsaufgaben verwendet wurden. Geben Sie die Befehle in eine STACK-Frage ein, um auszuprobieren, wie sie in einer Frage aussehen würden.

Für eine umfangreichere Dokumentation können Sie den folgenden Link nutzen:



<http://docs.mathjax.org/en/latest/>

- Brüche: $\frac{\text{Zähler}}{\text{Nenner}}$
- Produkte: $a \cdot b$ oder $a \times b$
- Integrale: $\int_{\text{Untergrenze}}^{\text{Obergrenze}} \text{Funktion } dx$

Für Doppelintegrale setzt man einfach ein weiteres i vor das int. Für drei noch ein i und so weiter.

- Wurzeln: \sqrt{x} oder $\sqrt[n]{x}$
- Potenzen: a^x

5.3.10 HTML

HTML können Sie verwenden, um die Frage ansprechender zu gestalten. Hierzu klicken Sie auf den HTML-Knopf über dem Eingabefeld.



Beispielsweise können Sie HTML verwenden, um [Tabellen zu gestalten](#). Neben der Hintergrundfarbe der Zellen können Sie auch beliebige andere Eigenschaften einer Tabelle anpassen, wie z.B. die Zeilen- und Spaltenhöhe und -breiten, die Linienbreiten und die Tabelle dazu nutzen um beispielsweise Infoboxen zu erstellen und Bilder zu platzieren.



Wichtige Befehle finden Sie im Internet beispielsweise auf www.w3schools.com oder Sie kopieren sich den HTML-Text von anderen Fragen in Ihre eigenen und wandeln ihn passend ab. Beispiele dazu gibt es in den [Übungsaufgaben](#).

Tabellen

Eine manuelle Gestaltung von Tabellen über HTML-Befehle ist möglich und kann eine gute Alternative zu den voreingestellten Optionen sein, wenn Sie spezielle Wünsche haben. Um eine Tabelle individuell zu gestalten, klicken Sie, wie oben beschrieben, auf den HTML-Knopf und fügen die entsprechenden HTML-Befehle ein.

Zuerst folgt der grundsätzliche Aufbau einer Tabelle, wobei es sich um eine Tabelle mit einer Zeile und zwei Spalten handelt. Es können beliebig viele Zeilen und Spalten ergänzt werden. Der linke Text entspricht dem HTML-Code, die rechte Seite ist die Erklärung des Befehls.

<code><table></code>	Beginn Tabelle
<code><tr></code>	Beginn Zeile
<code><td>Inhalt</td></code>	Erste Spalte in erster Zeile, Inhalt zwischen <td> und </td>
<code><td>Inhalt</td></code>	Erste Spalte in erster Zeile, Inhalt zwischen <td> und </td>
<code></tr></code>	Ende erste Zeile
<code></table></code>	Ende Tabelle

Die nun folgenden Befehle müssen immer an entsprechender Stelle platziert werden, je nachdem, ob sie sich auf die ganze Tabelle, eine Zeile oder nur eine Zelle beziehen, z.B. `<table align="center">` (Einstellung für ganze Tabelle), `<tr style="height: 50px;" >` (Einstellung für eine Zeile) oder `<td style="width: 50%;">` (Einstellung für eine Spalte).

Da es sehr viele Optionen gibt, wie man die Tabelle individuell gestalten kann, werden wir hier nur die geläufigsten Optionen vorstellen.

- **Ausrichtung Tabelle:** `align="left/center/right"`
- **Linienbreite:** `border="1"`
- **Breite der Tabelle/Spalte/Zeile:** `style="width: 100%;"`
- **Höhe der Tabelle/Spalte/Zeile:** `style="height: 64px;"`
- **Abstand zur Zellenbegrenzung:** `style="padding: 10px 10px 10px 10px;"`
- **Hintergrundfarbe:** `style="background-color: #c8d1db;"`

Die eingetragenen Werte sind hier natürlich nur beispielhaft und können angepasst werden.

Wenn verschiedene Optionen verwendet werden, die unter `style=" "` anzuordnen sind, werden diese hintereinander aufgezählt und mit einem Semikolon ; getrennt.

Um sehen zu können, wie das ganze am Ende aussehen kann und wie der Code dahinter aussieht, schauen Sie gerne in die Musterlösung der Übungsaufgabe [17 HTML Attribute](#).

Infobox

Tabellen können auch dafür genutzt werden, um Bilder zu platzieren oder Info- und Hinweisboxen zu gestalten. Auf letzteres gehen wir im Folgenden etwas näher ein.



Diese **Infobox** ist eigentlich eine Tabelle mit zwei Spalten und einer Zeile. Hier ist der Rand auf 0 gesetzt und in der einen Spalte ist das Infobild und die andere Spalte kann mit freiem Text gefüllt werden.

Zum Beispiel kann ich hier auf einen **Link** mit weiteren Informationen zu HTML verweisen, wenn Sie sich gerne weiter Informieren möchten!

Der folgende und vorrausgehende **große Absatz** entsteht durch einen weißen Punkt im HTML Code.

Das entscheidende bei dieser Art von Boxen ist, dass es eine Zeile und zwei Spalten gibt. Die Einstellungen wie z.B. die Breite der Spalten werden hier an diesem konkreten Beispiel anhand von dem gesamten Code kommentiert.

```
<table border="0">
```

```
<tr style="background-color: #e3e8eb;">
```

```
<td height="64" width="64" style="vertical-align: middle;">
```

```
<td style="vertical-align: middle; font-size: 14px; padding: 0 16px 0 0;">Beliebiger Text</td>
```

```
</tr>
```

```
</table>
```

```
<p style="color: white;".</p>
```

Beginn Tabelle, Zellenrand auf 0

Beginn Zeile, Zeilenhintergrundfarbe

Beginn Spalte, Spaltenhöhe und -breite, Ausrichtung des Inhalts, Link zum verwendeten Bild (dieser kann kopiert und so verwendet werden oder ein beliebig anderes Bild hochgeladen und genutzt werden), Ende Spalte

Beginn Spalte, Ausrichtung des Inhalts, Schriftgröße, Abstände zur Zellenwand, Inhalt, Ende Spalte

Ende Zeile

Ende Tabelle

Erzeugt den in der Infobox verwendeten größeren Absatz durch Einfügen eines weißen Punktes. (Die Farbe muss an den jeweiligen Hintergrund angepasst werden.)

6 Verwendung von STACK in Tests

STACK kann, wie andere Fragetypen in ILIAS, in Tests eingebunden werden, um Studierenden die Fragen zur Verfügung zu stellen. Prinzipiell funktioniert dies genau gleich, wie bei anderen Fragetypen. Wir empfehlen, die Fragen zunächst in einem Fragenpool zu erstellen und anschließend in einen Test einzufügen.

Die aus dem Fragenpool eingefügten Fragen können Sie direkt im Test bearbeiten. Die Änderungen können dann in den Fragenpool übernommen werden. Andersherum ist dies nicht möglich.

Hinweis: In einem Test können Änderungen an den Fragen nur vorgenommen werden, wenn keine Teilnehmerdaten im Test gespeichert sind. Sind die Fragen zusätzlich in einem Fragenpool gespeichert, können sie dort jederzeit bearbeitet werden. Ist die Bearbeitung des Tests durch Studierende vorbei, können die verbesserten Fragen anschließend im Test ersetzt werden.

Welche Einstellungen Sie für den Test verwenden hängt hierbei von Ihrem Einsatzszenario und Ihren didaktischen Überlegungen ab. Dies erfolgt analog zu anderen Fragetypen in ILIAS. Eine Einstellung ist jedoch besonders für STACK relevant. Möchten Sie, dass das individuelle Feedback der Rückmeldebäume angezeigt wird, muss in den Einstellungen des Tests im Abschnitt *Durchführung: Verhalten der Frage* die Option *Direkte Rückmeldung* aktiviert werden. In den dann erscheinenden Optionen wählen Sie anschließend die Option *Unterschiedliche Rückmeldungen pro gegebener Antwort* aus.

Teilnehmerergebnisse

Wenn Sie eine Ergebnisdatei erzeugen, erhalten Sie in der .xlsx-Datei folgende STACK-spezifische Informationen, die Ihnen im Nachhinein helfen die Eingaben der Studierenden nachzuvollziehen:

- Den **Fragetext** inkl. der Platzhalter für die Eingabefelder, das Feedback und für die (Zufalls-)Variablen.
- Den **Aufgabenhinweis**. Dieser ist in der Zeile „Fragen-Notiz“ hinterlegt und ermöglicht es nachzuvollziehen, welche Aufgabenversion den Studierenden angezeigt wurde. Deswegen ist wichtig, dass der Aufgabenhinweis alle Zufallsvariablen enthält und aus ihm klar hervor gehen, wie sich diese auf die angezeigte Aufgabe auswirken.
- Die **gegebenen Antworten** der Studierenden.

Wichtig: Die Antworten der Studierenden werden nur gespeichert, wenn das entsprechende Eingabefeld in einem Rückmeldebäum aufgerufen wird. Für den Fall, dass Sie Eingabefelder haben, welche nicht ausgewertet, aber dennoch gespeichert

werden sollen, empfehlen wir die Variable für das Eingabefeld in den Feedback-Variablen aufzurufen. z.B. NichtAusgewerteteAntworten:[ans1,ans2]. Alternativ ist es auch möglich, diese Input-Felder in einem separaten Rückmeldebaum zu verwenden, diesem Rückmeldebaum eine Gewichtung von 0 (Aufgabenwert 0) zu geben und ihn den Studierenden nicht zu zeigen, in dem Sie den Feedbackplatzhalter löschen.

7 Weiterführende Links

Es gibt darüber hinaus noch viele weitere Seiten, auf denen man nützliche Infos findet.
Eine **Special Interest Group** auf **Ilias.de** beschäftigt sich zudem mit STACK.

STACK-Dokumentation für moodle (englisch):



https://github.com/mathsmoodle-qtype_stack/tree/master/doc

https://github.com/mathsmoodle-qtype_stack/tree/master/doc/en/Authoring

Ausführliche Behandlung der Antwortüberprüfungsmöglichkeiten:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/Authoring/Answer_tests.md

Maxima-Einführung für STACK-Nutzer*innen:



https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/CAS/Maxima.md

Maxima-Dokumentation:



<http://maxima.sourceforge.net/docs/manual/maxima.html>

Hier kann unter [Index] nach bestimmten Maxima-Funktionen gesucht werden:



http://maxima.sourceforge.net/docs/manual/maxima_89.html#SEC422

MAXIMA-Onlinekurs des ACDCA:



<http://www.austromath.at/daten/maxima/>

Maxima Online Shell:



<http://maxima.cesga.es/>

MathJax:



<http://docs.mathjax.org/en/latest/>

8 Übungsmaterial

In diesem Kapitel werden verschiedene Aufgaben gegeben, mit denen Sie die Erstellung von STACK-Fragen üben und unterschiedliche Anwendungen ausprobieren können. Versuchen Sie zuerst selbstständig die STACK-Aufgaben zu erstellen. Orientieren Sie sich dabei an dem [Flowchart](#) zu Beginn dieses Dokuments und den Erklärungen in den darauffolgenden Kapiteln. Bei Unklarheiten können Sie auf die verlinkten Hilfestellungen sowie die Musterlösung zurückgreifen. Die Musterlösung finden Sie in dem Kurs „Einstieg STACK 3“ unter



[Einstieg STACK 3 -> Einstieg für AutorInnen -> Musterlösungen zur „Einführung STACK3“¹](#)

Die Übungsaufgaben sind unterteilt in drei Abschnitte. Im Abschnitt zur algebraischen Eingabe lernen Sie die Grundlagen der STACK-Frage und im zweiten Abschnitt verschiedene Eingabetypen, die für bestimmte Aufgabentypen äußerst nützlich sein können. Der letzte Abschnitt beschäftigt sich mit einer ansprechenden Darstellung der Frage und der Rückmeldungen.

Falls Sie versehentlich Beispielfragen verändern, melden Sie uns das bitte zurück, damit wir diese für künftige Nutzer*innen wieder in den Ausgangszustand versetzen können.

Sie erreichen uns unter: ilias-stack@uni-koeln.de



In dem [Kurs der Self-Assessment-Instanz](#) haben Sie die Möglichkeit, selbst Fragen zu schreiben und die Aufgaben zu lösen. Bitte erstellen Sie Ihre Fragen im **Sandkasten zum Ausprobieren**.

¹ Eine Anleitung, wie Sie dem Kurs beitreten, finden Sie in diesem Dokument unter [Beitritt zum Kurs](#).

8.1 Übungsaufgaben

Bearbeiten Sie die Übungsaufgaben zunächst selbstständig. Wenn Sie Fragen zur Umsetzung haben oder Ihre Lösung vergleichen möchten, können Sie dies mit der verlinkten Hilfestellung tun.

Wichtig: Zum Erstellen einer Frage gehört es immer auch dazu, ihre Funktionalität zu überprüfen. Schauen Sie sich dazu die Vorschau der Frage an und geben Sie testweise Antworten ein, um die unterschiedlichen Rückmeldungen auszulösen und so zu überprüfen, ob die Frage wie gewünscht funktioniert.

Übungsaufgaben zur algebraischen Eingabe

In diesem Abschnitt zur algebraischen Eingabe wird eine STACK-Frage nach und nach aufgebaut und erweitert. Erstellen Sie für die erste Aufgabe eine STACK-Frage und erweitern Sie diese mit den darauffolgenden Aufgaben.

1. Erstellung der ersten STACK-Frage

Erstellen Sie die erste Stack-Frage mit dem Fragetext:

„Rechnen Sie $3 + \frac{4}{2}$ aus.“ Geben Sie eine passende Rückmeldung und Punkte im Rückmeldebaum.

[Hilfestellung – Schritt für Schritt](#)

2. Einfügen von weiteren Knoten

Fügen Sie der STACK-Aufgabe einen zweiten Rückmeldeknoten hinzu, der überprüft, ob die Punkt-vor-Strich-Regel missachtet wurde und geben Sie eine hilfreiche Rückmeldung, falls der Fehler gemacht wurde.

[Hilfestellung – Schritt für Schritt](#)

3. Einfügen einer weiteren Teilaufgabe

a) Fügen Sie der Stack-Frage eine weitere Teilaufgabe hinzu:

„Rechnen Sie $6 - 2 * 2$ aus“

Fügen Sie entsprechend weitere Rückmeldeknoten und passende Rückmeldungen hinzu.

b) Überprüfen Sie mit Hilfe eines neuen Rückmeldebaums, ob ein Wiederholungsfehler vorliegt, also die Punkt-vor-Strich-Regel in beiden Teilaufgaben missachtet wurde.

- c) Verschieben Sie anschließend die Platzhalter der Rückmeldebäume aus *Spezifisches Feedback* an die entsprechende Stelle in das *Frage-Feld* und schauen Sie sich die Veränderung der Darstellung in der Vorschau an.

[Hilfestellung – Schritt für Schritt](#)

4. Nutzen von Feedbackvariablen

Nutzen Sie nun Maxima-Befehle, um zu überprüfen, ob ein Wiederholungsfehler vorliegt. Dies ist eine gute Alternative zu aufwendigen Rückmeldebäumen und kann mit etwas Erfahrung die Rückmeldungen vereinfachen.

Definieren Sie hierfür eine Feedbackvariable „WdhF“ mithilfe der Maxima-Befehle **is(<expr>)** und **<expr>and<expr>**. Nutzen Sie die Variable zum Abgleich mit **TAns** und **SAns**.

[Hilfestellung – Schritt für Schritt](#)

5. Verwendung zufälliger Variablen

Ersetzen Sie die Zahlen in der Aufgabenstellung durch Zufallsvariablen. Nutzen Sie dafür die Befehle **rand**, **rand_with_step** und **rand_with_prohib**.

Hinweis: Eine Erklärung zu der [Verwendung von Variablen](#) und eine [Erklärung der Befehle](#) finden Sie in diesem Dokument.

[Hilfestellung – Schritt für Schritt](#)

6. Vorbeugende Maßnahmen

Erstellen Sie für diese Aufgabe eine **neue** STACK-Frage mit dem folgenden Fragetext:

„Differenzieren Sie $x^{a1} * e^{x*a2}$ nach x . Berechnen Sie den Bruch $\frac{a1}{a2}$.“

$a1, a2$ sollen Zufallsvariablen sein. Achten Sie darauf entsprechende Maxima-Befehle zu verbieten, sodass sich die Lösung nicht durch Eingabe des Befehls „erschummeln“ lässt.

[Hilfestellung – Schritt für Schritt](#)

7. Runden und Funktionen

Erstellen Sie für diese Aufgabe eine **neue** STACK-Frage mit den folgenden Fragetext:

„Geben Sie π auf 4 Nachkommastellen genau an.“

Überprüfen Sie, ob die angegebene Lösung korrekt und auf exakt 4 Stellen gerundet wurde. Benutzen Sie **round2()** aus den [Maxima Workarounds](#).

Hinweis: Die Maxima-Funktion **round()** ermöglicht das Runden ebenfalls. Jedoch verwendet diese Funktion das symmetrische Runden. Die Funktion **round2()**, die wir selbst

geschrieben haben, ermöglicht kaufmännisches Runden, wie man es in den meisten Fällen verwenden möchte.

[Hilfestellung – Schritt für Schritt](#)

Übungsaufgaben zu anderen Eingabetypen

In den bisherigen Aufgaben wurde unter *Einstellungen der Eingabe* immer der *Eingabetyp Algebraische Eingabe* verwendet. In den folgenden Aufgaben wird es darum gehen, einige der anderen *Eingabetypen* innerhalb von STACK kennenzulernen. Wählen Sie diese entsprechend der Aufgabenstellungen unter *Einstellungen der Eingabe* aus.

8. Matrizen

Erstellen Sie die Matrix

$$A = \begin{pmatrix} -2 & 4 & 1 \\ 1 & 0 & 2 \\ -2 & -5 & 3 \end{pmatrix}$$

sowie zwei 3x3 Zufallsmatrizen B und C und stellen Sie folgende Aufgabe:

„Berechnen Sie $D = B - A$

Berechnen Sie $E = B * C$ “

Verwenden Sie dazu den *Eingabetyp Matrix*.

[Hilfestellung – Schritt für Schritt](#)

9. Kontrollkästchen (Multiple-Choice)

Erstellen Sie die folgende Frage als Multiple Choice-Aufgabe mit Hilfe des Eingabetyps **Kontrollkästchen**:

„Bitte geben Sie alle Lösungen der folgenden Gleichung an: $|x| = 3$ “

[Hilfestellung – Schritt für Schritt](#)

10. Drop-Down-Liste (Single-Choice)

Erstellen Sie die folgende Aufgabe mit Hilfe des Eingabetyps **Drop-Down-Liste**:

„Die Funktion $f(x) = x^3 - 2 * x^2 - 5 * x + 6$ ist eine

- a) Ganzrationale Funktion
- b) Exponential Funktion
- c) Gebrochenrationale Funktion
- d) Trigonometrische Funktion

Die maximale Anzahl der Nullstellen dieser Funktion kann man abschätzen

- a) nur durch eine genaue Rechnung
- b) mit Hilfe des niedrigsten Exponenten
- c) mit Hilfe des höchsten Exponenten
- d) erst wenn man Mathe studiert hat

Die Anzahl der Nullstellen von der Funktion $f(x)$ lautet:

- a) 3
- b) 1
- c) 0
- d) 2

Die Funktion hat den Schnittpunkt mit der y-Achse bei

- a) 6
- b) -6
- c) -2
- d) 5

[Hilfestellung – Schritt für Schritt](#)

11. Radio-Auswahl (Single-Choice)

Erstellen Sie außerdem die folgende Aufgabe mit Hilfe des Eingabetyps **Radio-Auswahl**:

„Welche Aussage ist richtig:

- a) eins plus vier ist fünf
- b) zwei plus fünf ist zwölf
- c) drei plus sechs ist einundzwanzig
- d) fünf plus acht ist vierunddreißig

[Hilfestellung – Schritt für Schritt](#)

12. Gleichungen Einstieg

Erstellen Sie folgende Aufgabe mit Hilfe des Eingabetyps **Gleichungen**:

„Berechnen Sie die Nullstellen von: $f(x) = x^2 - 8x$ “

[Hilfestellung – Schritt für Schritt](#)

13. Notizen

Erstellen Sie mit Hilfe des Eingabetyps **Notizen** eine Frage, die es den Studierenden am Ende eines Tests ermöglicht eine Rückmeldung zum Test zu geben.

[Hilfestellung – Schritt für Schritt](#)

14. Einheiten

Erstellen Sie mit Hilfe des Eingabetyps **Einheiten** die folgende Aufgabe:

„Geben Sie die Erdbeschleunigung in SI-Einheiten an.“

[Hilfestellung – Schritt für Schritt](#)

15. Zeichenkette

Erstellen Sie mit Hilfe des Eingabetyps **Zeichenkette** die folgende Aufgabe:

„Welche Integrationsregel würden Sie nutzen, um $\int x \sin(x) dx$ zu berechnen?“

[Hilfestellung – Schritt für Schritt](#)

16. Numerisch

Erstellen Sie mit Hilfe des Eingabetyps **Numerisch** die folgende Aufgabe: „Geben Sie die ersten vier Stellen von π an.“

[Hilfestellung – Schritt für Schritt](#)

Übungsaufgaben für eine ansprechende Rückmeldung

17. HTML zur Darstellung verwenden

Versuchen Sie Ihre Frage ansprechend zu gestalten und nutzen Sie dafür HTML. In der Musterlösung finden Sie hierzu etwas Inspiration und Ideen.

[Hilfestellung – Schritt für Schritt](#)

18. Plots für Rückmeldungen verwenden

Erstellen Sie eine Frage, in der die Studierenden eine Funktion mit Zufallsvariablen ableiten sollen. Als Rückmeldung sollen die richtig und die falsch abgeleitete Funktion in einem Plot verglichen werden.

[Hilfestellung – Schritt für Schritt](#)

19. Plot Attribute Beispielsammlung

Nutze Sie verschiedene Optionen, um Plots zu erstellen. Lassen Sie sich von den Darstellungsbeispielen aus der Frage [19 Plot Optionen](#) in dem ILIAS-Kurs inspirieren.

[Hilfestellung – Schritt für Schritt](#)

8.2 Schritt-für-Schritt-Hilfestellungen

Hilfestellung zur algebraischen Eingabe

Erstellen der ersten STACK-Frage (01_Erste Frage)

Inhalt: Frage erstellen

Zu beachtende neue Einstellungen:

- **Titel**
- **Frage**
- **Test-Eingaben**
 - **Einstellungen der Eingabe ans1**
 - **Musterlösung**
- **Rückmeldebäume**
 - **SAns**
 - **TAns**
 - **Mod (wenn wahr/wenn falsch)**
 - **Punkte (wenn wahr/wenn falsch)**
 - **Feedback (wenn wahr/wenn falsch)**
 - **Feedback-Style (wenn wahr/falsch)**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 01 Erste Frage](#)

Einfügen von weiteren Knoten (02_Zweiter Knoten - Punkt vor Strich)

Inhalt: Knoten erstellen und verknüpfen

Man kann sich vorstellen, dass Studierende die Punkt-vor-Strich Regel missachten. Daher ergibt es Sinn, einen zweiten Knoten zu erstellen, welcher, falls das Ergebnis falsch ist, prüft, ob die Punkt-vor-Strich Regel missachtet wurde.

In dem Beispiel *02_Zweiter Knoten* wird zuerst (in Knoten 1) abgefragt, ob das Ergebnis richtig oder falsch ist. Danach (in Knoten 2) wird erst geprüft, ob das Ergebnis richtig ist.

Die Reihenfolge der beiden Knoten ist allerdings egal, da beide Abfragen sich gegenseitig ausschließen.

Zu beachtende neue Einstellungen:

- **Rückmeldebäume**
 - **Neuer Knoten**
 - **Ersten Knoten bearbeiten**
 - **Knoten wenn falsch und Feedback wenn falsch anpassen**



Einfügen einer weiteren Teilaufgabe (03_Zweites Eingabefeld a) & 03_Zweites Eingabefeld b))

Inhalt: Mehrere Input-Felder, Bewertung anpassen

Fügen Sie analog zur ersten Frage ein weiteres Eingabefeld ein, in welchem nach dem Ergebnis von $6 - 2 * 2$ gefragt wird. Speichern Sie ab und gehen analog zur ersten Frage für die „Test-Eingabe“ für die zweite Eingabe und einen neuen Rückmeldebaum vor. Benennen Sie diesen sinnvoll und speichern ab, bevor Sie den Rückmeldebaum weiter ausfüllen. Erstellen Sie auch hier einen zweiten Knoten zur Überprüfung der Punkt-vor-Strich Regel.

Zu beachtende neue Einstellungen:

- **Frage**
- **Test-Eingabe**
 - **Einstellungen der Eingabe ans2**
- **Rückmeldebäume**
 - **Neuer Rückmeldebaum (PRT)**
- **Spezifisches Feedback**
 - **Kopiere die Feedback Blöcke in das Frage Feld, hinter die entsprechende Fragestellung**

Hiermit kann allerdings nicht festgestellt werden, ob ein Wiederholungsfehler aufgetreten ist. Dafür müssen die Fälle, dass zwei Mal die Regel missachtet wurde und, dass die Regel nur einmal beachtet wurde, unterschieden werden. Hierfür erstellen Sie eine Frage wie *03_Zweites Eingabefeld b)*.

Hier wird ein dritter Rückmeldebaum erstellt, der überprüft, ob bei beiden Eingaben die Regel missachtet wurde. Hierbei ist es wichtig, dass Sie den Rückmeldebaum aus der Bewertung herausnehmen.

Zu beachtende neue Einstellungen:

- **Rückmeldebäume (PRTS)**
 - **Einstellungen**
 - **Aufgabenwert auf 0**



Nutzen von Feedbackvariablen (04_Maxima)

Inhalt: Maxima, Logische Verknüpfungen, Antwortüberprüfungen, Feedbackvariablen

Bisher wurde die Programmiersprache Maxima, mit der man Variablen global und lokal in STACK definieren kann, in den Beispielen noch nicht benutzt. Diese ist jedoch hilfreich, um logische Verknüpfungen, Berechnungen und Zufallswerte einzufügen. Im Folgenden werden diese Feedbackvariablen benutzt, um die Aufgabe umzuschreiben.

- Starten Sie mit dem gleichen Aufgabentext wie zuvor.
- Klicken Sie im Rückmeldebaum auf **Einstellungen**, im Feld Feedback-Variablen können Sie nun in Maxima programmieren. Hierbei fragt **is(<expr>)** ab, ob der Inhalt der Klammern wahr ist. Der Befehl **<expr>and<expr>** verknüpft die beiden logischen Aussagen zu einem **true** oder **false**.

Feedback-Variablen

WdhF: is(ans1=3,5) and is(ans2=8)

Die Feedback-Variablen erlauben es die Eingabe zusammen mit den Aufgabenvariablen zu manipulieren, bevor der Rückmeldebaum durchlaufen wird. Variablen, die hier definiert werden, können überall im Rückmeldebaum benutzt werden.
Inhalt: CAS-Ausdruck

- Die so erhaltene Variable kann nun in einem Knoten verwendet werden. Hierfür schreiben Sie den Variablennamen *WdhF* in **TAns** und *true* in **SAns**.

Beachten Sie, dass das Abfragen von mehr Aufgabenteilen ohne die Nutzung von Maxima-Befehlen sehr viel aufwendiger ist.

Zu beachtende neue Einstellungen:

- **Rückmeldebäume**
 - **Einstellungen**
 - **Feedback-Variablen**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 04_Maxima](#)

Verwendung zufälliger Variablen (05_Zufallszahlen)

Inhalt: Zufallszahlen, simp, individuelle Aufgaben

Um Aufgaben verschieden zu gestalten oder um bspw. Abschreiben vorzubeugen, sollten Sie Zufallszahlen oder Zufallsausdrücke verwenden. In unserem Beispiel wird die vermutlich häufigste Verwendung von Zufallsfunktionen in Aufgaben gezeigt. Wir ersetzen feste Zahlenwerte durch Zufallszahlen.

Sie müssen die Zufallszahlen im Feld **Aufgabenhinweis** notieren. Wenn Sie Funktionen mit Zufallswerten definieren, stellen Sie außerdem die **Aufgabenweite-Vereinfachung** aus, um zu verhindern, dass Brüche automatisch gekürzt werden.

Achten Sie darauf, dass Sie sowohl die Musterlösung als auch den Rückmeldebaum an die neuen Einstellungen anpassen.

Hierfür können Sie außerdem vorbereitete Zufallswerte nutzen, um für den Tests Zufallswerte auszugeben, die Sie passend finden.

Zu beachtende neue Einstellungen:

- **Fragen-Variablen**
- **Aufgabenhinweis**
- **Optionen**
 - **Zeige Fragenoptionen**
 - **Aufgabenweite-Vereinfachung**
- **Test-Eingaben**
 - **Einstellung der Eingabe**
 - **Musterlösung**
- **Rückmeldebäume**
 - **TAns**
 - **Einstellungen**
 - **Feedback-Variablen**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 05 Zufallszahlen](#)

Vorbeugende Maßnahmen (06_Verbotene Wörter)

Inhalt: Verbotene Wörter, Zufällige Operatoren

Um Wörter oder Maxima-Befehle (z.B. differenzieren, integrieren, etc.) zu verbieten, müssen Sie die entsprechenden Befehle oder Operatoren in **Test-Eingaben** → **Einstellungen der Eingabe** → **Verbotene Wörter** eintragen.

In der Beispielaufgabe sind diese unerwünschten Wörter der Maxima-Begriff „diff“ und die Rechenzeichen „*“ und „/“. Letztere Maßnahme verhindert, dass die Aufgabenstellung in das Antwortfeld eingetippt werden kann, statt selbstständig zu rechnen.

Zu beachtende neue Einstellungen:

- **Test-Eingaben**
 - **Einstellungen der Eingabe (ans1/ans2)**
 - **Verbotene Wörter**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 06 Verbotene Wörter](#)

Runden und Funktionen (07_Runden)

Inhalt: Runden, Funktionen (selbst definieren)

Die bei Maxima eingebaute Rundungsfunktion verwendet symmetrisches Runden. Diese führt jedoch in manchen Fällen zu unerwünschten Ergebnissen, da in der Regel das sogenannte kaufmännische Runden gewünscht ist.

Hierzu haben wir die Funktion **round2()** geschrieben, welche im Abschnitt [9.3 Workarounds](#) beschrieben wird. Sie rundet einen Wert **ZTV** auf **ZTN** Nachkommastellen.

```
round2(ZTV,ZTN):=(if      ZTV*10^ZTN-floor(ZTV*10^ZTN)<0.5      then  
float(floor(ZTV*10^ZTN)*10^-ZTN) else float(floor(ZTV*10^ZTN+1)*10^-ZTN));
```

Fügen Sie diese Funktion in die **Fragen-Variablen** ein, um sie zu verwenden.

Diese Funktion wird auf die Studierendenantwort und auf die Musterlösung angewendet, bevor diese verglichen werden. Wenn diese nicht übereinstimmen, wird nachgeschaut, ob die Antwort falsch gerundet wurde. Hierfür kann man **NumAbsolute** verwenden und als Intervall (in den **Test-Optionen**) **10^-ZTN** als Dezimalzahl eingeben.

Zu beachtende neue Einstellungen:

- **Fragen-Variablen/ TAns und SAns**
 - **round2()**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 07_Runden](#)

Hilfestellungen zu anderen Eingabetypen

Matrizen (08_Matrizen)

Inhalt: Matrizen erstellen, Rechnen mit Matrizen

Um Matrizen eingeben zu lassen, stellen Sie den Eingabetyp auf **Matrix**. Die Rechenoperationen in Maxima Syntax zur Verwendung innerhalb der Aufgabe können unter folgendem Link gefunden werden:



http://maxima.sourceforge.net/docs/manual/maxima_118.html#Functions-and-Variables-for-Matrices-and-Linear-Algebra

Herauszustellen ist:

- Das nicht kommutative Matrixprodukt wird durch **<Matrix1>.<Matrix2>** beschrieben, die anderen Grundrechenarten sind durch die gewohnten Zeichen aufzurufen.
- Die **genmatrix** Funktion generiert mit Hilfe einer angegebenen Funktion eine Matrix. Die im Beispiel benutzte **genmatrix** Funktion generiert eine Matrix aus Integer-

Zufallszahlen. Eine ausführliche Erklärung zu dieser Funktion finden Sie in diesem Dokument unter [Maxima → Grundlegende Befehle](#).

Zu beachtende neue Einstellungen:

- **Fragen-Variablen**
- **Einstellungen der Eingabe**
 - **Eingabetyp**
- **Zeige Fragenoptionen**
 - **Matrix Klammern**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 09 Matrizen](#)

Kontrollkästchen (Multiple-Choice) (09_Kontrollkästchen)

Inhalt: Multiple-Choice Fragen mit Kontrollkästchen, Maxima Syntax Liste

Beachten Sie die [Syntaxhinweise](#) aus den Erläuterungen zum Eingabetyp **Kontrollkästchen**.

Arbeitsschritte:

- Erstellen Sie eine STACK Aufgabe und schreiben Sie in das Feld *Frage*: „Bitte geben Sie alle Lösungen der folgenden Gleichung an: $\backslash(\text{abs}(x)=3\backslash)$ `[[input:ans1]] [[validation:ans1]]`“
- Geben Sie in *Fragen-Variablen* die eine Liste mit allen Antwortmöglichkeiten an, die angezeigt werden sollen und geben sie entsprechend *false* oder *true* an.
- Vermischen Sie die Antwortmöglichkeiten zufällig mit dem Befehl *random_permutation(list)* und hängen Sie mit dem Befehl *append(list1,list2)* „Keine der Antworten ist korrekt.“ an.
- Werten Sie die Eingabe im PRT mit Hilfe des Feldes Feedback-Variablen aus.
 - Nutzen Sie dafür *member(x,list)* und logische Verknüpfungen wie *or*.
 - Nutzen Sie *sort(list)* um Listen zu sortieren und somit vergleichen zu können
- Geben Sie sinnvolle Rückmeldungen im PRT ein und legen Sie eine Bewertung fest.

Zu beachtende neue Einstellungen:

- **Fragen-Variablen**
- **Test-Eingabe**
 - **Einstellung der Eingabe**
 - **Eingabetyp**
 - **Kontrollkästchen**
 - **Musterlösung**
- **Rückmeldebäume**

- **Einstellungen**
 - **Feedback-Variablen**
- **SAns**
- **TAns**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 09 Kontrollkästchen](#)

Drop-Down-Liste (Single-Choice) (10_Drop-Down-Liste)

Inhalt: Drop-Down-Listen

Das Vorgehen ist ähnlich wie bei der Erstellung einer Multiple-Choice-Aufgabe mit Hilfe des Eingabetyps **Kontrollkästchen**. Beachten Sie dementsprechend die [Syntaxhinweise](#) aus den Erläuterungen der Eingabetypen. Es ist allerdings zu beachten, dass beim Eingabetyp **Drop-Down-Liste** nur eine Antwortmöglichkeit als "true" ausgewiesen werden kann.

Zu beachtende neue Einstellungen:

- **Fragen-Variablen**
- **Test-Eingabe**
 - **Einstellung der Eingabe**
 - **Eingabetyp**
 - **Drop-Down-Liste**
 - **Musterlösung**
- **Rückmeldebäume**
 - **SAns**
 - **TAns**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 10 Drop-Down-Liste](#)

Radio-Auswahl (Single-Choice) (11_Radio-Auswahl)

Inhalt: Radio-Auswahl

Das Vorgehen ist ähnlich wie bei der Erstellung einer Single-Choice-Aufgabe mit Hilfe des Eingabetyps **Drop-Down-Liste**. Beachten Sie dementsprechend die [Syntaxhinweise](#) aus den Erläuterungen der Eingabetypen. Es ist allerdings zu beachten, dass auch beim Eingabetyp **Radio-Auswahl** nur eine Antwortmöglichkeit als "true" ausgewiesen werden kann.

Zu beachtende neue Einstellungen:

- **Fragen-Variablen**

- **Test-Eingabe**
 - **Einstellung der Eingabe**
 - **Eingabetyp**
 - **Radio-Auswahl**
 - **Musterlösung**
- **Rückmeldebäume**
 - **SAns**
 - **TAns**



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 12 Radio-Auswahl](#)

Gleichungen Einstieg (12_Gleichungen)

Inhalt: Gleichungen

Beachten Sie die [Syntaxhinweise](#) im Abschnitt Eingabetypen.

Hier werden die Studierendenantwort, sowie die Musterlösung als Liste von allen Zwischenschritten gespeichert. Im Rückmeldebaum werden also zur Überprüfung die beiden letzten Schritte, also das Ergebnis, mit dem Maxima Befehl *last(list)* verglichen. Dieser Befehl gibt das letzte Element einer Liste aus.

Außerdem wird unter *Syntax-Hinweis* die erste Linie eingegeben, um sicher zu sein, dass mit der Gleichen Ausgangsposition gestartet wird.

Zu beachtende neue Einstellungen:

- **Frage**
- **Fragen-Variablen**
- **Test-Eingabe**
 - **Einstellung der Eingabe**
 - **Eingabetyp**
 - **Gleichungen**
 - **Musterlösung**
 - **Eingabebreite**
 - **Syntax-Hinweis**
 - $x^2-8*x=0$
- **Rückmeldebäume**
 - **SAns**

- TAns



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 12 Gleichungen](#)

Notizen (13_Notizen)

Inhalt: Notizen, Feedback, Rückmeldungen

Zu beachtende neue Einstellungen:

- Frage
- Test-Eingabe
 - Einstellung der Eingabe
 - Eingabetyp
 - Notizen
 - Musterlösung
 - *true*
 - Eingabebreite
- Rückmeldebäume
 - SAns
 - TAns



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 13 Notizen](#)

Einheiten (14_Einheiten)

Inhalt: SI-Einheiten

Beachten Sie bei der Eingabe der Musterlösung, dass Sie die Maßeinheiten mit Multiplikationszeichen verbinden müssen.

Bei der Antwortüberprüfung gibt es verschiedene Möglichkeiten, um eine Fehlertoleranz einzubeziehen. Zum einen gibt es die absolute (...*Absolute*) und die relative (...*Relative*) Abweichung. In beiden Fällen wird die tolerierte Abweichung in das Feld **Test-Optionen** eingetragen, entweder als absoluter Wert der Abweichung in beide Richtungen (*Absolute*) oder als Prozentsatz des richtigen Ergebnisses, wobei der Wert 1 den 100% entspricht (*Relative*).

Zu beachtende neue Einstellungen:

- Frage
- Test-Eingabe
 - Einstellung der Eingabe
 - Eingabetyp

- Einheiten
 - Musterlösung
- Rückmeldebäume
 - Antwortüberprüfung
 - z. B. *UnitsStrictRelative*
 - SAns
 - TAns
 - Test-Optionen
 - z.B. *0.02*



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 14 Einheiten](#)

Zeichenkette (15_Zeichenkette)

Inhalt: Zeichenkette, Begriffe

Beachten Sie die [Syntaxhinweise](#) zu dem Eingabetyp **Zeichenkette**.

Zu beachtende Einstellungen:

- Test-Eingabe
 - Einstellung der Eingabe
 - Eingabetyp
 - Zeichenkette
 - Musterlösung
- Rückmeldebäume
- Rückmeldebäume
 - Antwortüberprüfung
 - *StringSloppy*
 - SAns
 - TAns



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 15 Zeichenkette](#)

Numerisch (16_Numerisch)

Inhalt: Numerische Eingaben, Runden, signifikante Stellen, Rundungsfehler

Sie können die Antwort auf die richtige Anzahl der signifikanten Stellen mit Hilfe der Einstellung **NumSigFigs** unter **Antwortüberprüfung** im **Rückmeldebaum** untersuchen. Unter **Test-**

Optionen können Sie die gewünschte Anzahl eingeben. Beachten Sie, dass hier nicht die Anzahl der Nachkomma-Stellen, sondern aller signifikanter Stellen überprüft wird.

Falls Sie abweichende Ergebnisse aufgrund von Rundungsungenauigkeiten zulassen möchten, können Sie unter **Test-Optionen** eine Liste der Form $[n, n-1]$ angeben. Hierdurch wird die Antwort auf n Stellen geprüft. Die Übereinstimmung mit der Musterlösung wird allerdings nur bis zur $n-1$ -ten Stellen abgeglichen. Die Nutzung von **NumSigFigs** bei einer Eingabe von $[4, 3]$ unter **Testoptionen** bedeutet in diesem Fall, dass alle Angaben von π der Form $3.14x$ als richtig bewertet werden.

Des Weiteren werden mit **NumSigFig** bei Abweichungen der signifikanten Stellen oder der Rundungsgenauigkeit automatische Fehlermeldungen generiert:

„Ihre Antwort hat die falsche Anzahl an Dezimalstellen. Die Genauigkeit Ihrer Antwort ist nicht korrekt. Entweder haben Sie das Endergebnis oder einen Zwischenwert falsch gerundet.“

Zu beachtende Einstellungen:

- **Test-Eingabe**
 - **Einstellung der Eingabe**
 - **Eingabetyp**
 - **Numerisch**
 - **Musterlösung**
- **Rückmeldebäume**
 - **Antwortüberprüfungs**
 - **NumSigFigs**
 - **Test-Optionen**
 - **SAns**
 - **TAns**



[Einstieg STACK-Frage](#) → [Musterlösungen zur „Erklärung STACK 3“](#) → [16 Numerisch](#)

Musterlösungen für eine ansprechende Rückmeldung

HTML zur Darstellung verwenden. (17_HTML Attribute)

Inhalt: HTML verwenden

Die Gestaltung der Felder mit HTML kann die STACK-Frage für Studierende übersichtlicher machen. Dies gilt vor allem für zum Beispiel Tabellen. Wenn Sie hier Farben verwenden, können Studierende auf den ersten Blick sehen, wie die Tabelle aufgebaut ist.

Um HTML zu verwenden, benutzen sie den HTML Knopf über den Eingabefeldern (entweder Frage oder Feedback).



Es öffnet sich ein neues Fenster, in welches Sie nun HTML Befehle eingeben können. Nach einem Klick auf **Aktualisieren** wird das Fenster geschlossen und die Änderungen werden übernommen.



Wenn Sie sich selbst mit den HTML-Befehlen auseinandersetzen wollen, geht dies beispielsweise unter www.w3schools.com. Alternativ können Sie auch aus den Beispielfragen den HTML Code herauskopieren und in Ihre eigenen Fragen einbauen.

In unserem Anschauungsbeispiel wird gezeigt wie man

- eine farbig hinterlegte Infobox gestaltet.
- große Absätze einfügt.
- Hintergrundfarben von Tabellen und Freitexten ändert.
- einzelne Einstellungen der Tabelle anpasst.
- Schriftfarben ändert.
- Hyperlinks einfügt.

Hinweis: Die von uns verwendeten Standardfarben im Feedback sind: rot (#ffe7e7), grün (#e7ffe7) und gelb (#fcf8e3). Unsere Standardfarben in Tabellen sind dunkelblau (#c8d1db) und hellblau (#e3e8eb). Andere Farben finden Sie, wenn Sie HTML Farben im Internet suchen.



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 17 HTML Attribute](#)

Plots für Rückmeldungen verwenden (18_Plot Ableitung)

Inhalt: Plots verwenden

Plots können Sie verwenden, indem Sie im „Frage“-Feld oder im „Feedback wenn...“ Feld @plt@ aufrufen, wobei auf plt in den Frage-Variablen oder den Feedback-Variablen vorher ein Plot-Befehl geschrieben wurden muss.

Feedback-Variablen

```
plt:plot([ans1,f],[x,-5,5],[color,red,blue])
plt2:plot([ans1,fdx],[x,-5,5],[color,red,green])
```

Die Feedback-Variablen erlauben es die Eingabe zusammen mit den Aufgabenvariablen zu manipulieren,

Zu beachtende Einstellungen:

- Der Maxima-Befehl „plot2d“ oder „plot3d“ funktioniert in STACK aus Sicherheitsgründen nicht, jedoch können viele der „plot2d“ Attribute in **plot** verwendet werden.
- **plot([f1,f2,...],[x,x0,x1],[y,y0,y1])** gibt einen Plot der Funktionen f1, f2,... aus, welcher zwischen x0 und x1 bzw. y0 und y1 dargestellt wird. Sie können auch **plot(f1,[x,x0,x1])** verwenden.
- Sie sollten folgenden HTML-Befehl verwenden, wenn Sie @plt@ auf farbigem Hintergrund, zum Beispiel im Feedback, aufrufen. Durch den Befehl wird die Hintergrundfarbe auf Weiß umgestellt und entfernt die Ränder. Auf dem farbigen Hintergrund ist der Plot sonst schwieriger zu erkennen.

```
<div style="margin: -0px -16px -0px; padding: -0px 16px 0px; opacity: 1; background-color: ffcccc70;"><div style="opacity: 1; background-color: white;">@plt@</div></div>
```



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 18 Plot Ableitung](#)

Plot Attribute Beispielsammlung (19_Plot-Optionen)

Inhalt: Beispielplots



Die Beispiele stammen von https://github.com/mathsmoodle-qtype_stack/blob/master/doc/en/CAS/Plots.md.

Zu beachtende Einstellungen:

- Diese Frage wurde ausschließlich zur Darstellung erstellt. Sie prüft nichts ab, soll aber zeigen wie Plots in STACK aussehen können.
- Die Plots sind alle im Feedback-Variablen Feld erstellt worden



[Einstieg STACK-Frage → Musterlösungen zur „Erklärung STACK 3“ → 19 Plot-Optionen](#)

9 Abkürzungsverzeichnis

CAS

Das **Computer Algebra System (CAS)**, welches an STACK angeschlossen ist, ist Maxima.

PRT

Der **Potential Response Tree (PRT)** ist der Rückmeldebaum, in dem Sie die Antworten der Studierenden (**SAns**) mit der Antwort der Dozierenden (**TAns**) abgleichen wird, um Punkte zu vergeben und Feedback geben können.

SAns

Die **Student Answer (SAns)** ist die Antwort der Studierenden, die mit Hilfe des Platzhalters für das Eingabefeld in den Rückmeldebaum (**PRT**) eingetragen wird.

TAns

Die **Teacherr Answer (TAns)** ist die korrekte Lösung der Aufgabe, die als mathematischer Ausdruck in den Rückmeldebaum (**PRT**) eingetragen wird, um sie mit der Antwort der Studierenden (**SAns**) abzugleichen. **TAns** kann hierbei jedoch auch ein antizipierter Fehler sein, für den im Rückmeldebaum überprüft werden soll, ob dieser bestimmte Fehler von den Studierenden gemacht wurde.

10 Kontakt

Falls Sie Interesse an einer STACK-Schulung haben oder Fragen und Anregungen zu diesem Dokument oder zu STACK äußern möchten, können Sie sich gerne jederzeit an uns wenden.

Sie erreichen uns per E-Mail:

llias-stack@uni-koeln.de

STACK-Team

Eva Mix

Tjark Eilts

Annemarie Sich