

Berücksichtigung von Folgefehlern im ONYX Aufgabensystem

Stand 2018-6-14, ONYX Testsuite 7.5.1

Es gibt Codebeispiele (erkennbar durch diese Schriftart), welche über eine Zeile lang sind und deshalb hier im Dokument einen Zeilenumbruch haben. Dieser Zeilenumbruch muss bei Verwendung von Copy-Paste wieder entfernt werden. Wurde eine lange Zeile hier im Dokument umgebrochen, so ist das durch Einrückung der Zeile gekennzeichnet.

Dieses Beispiel entspricht einer Zeile, welche ohne Zeilenumbruch dargestellt werden müsste, wenn genug Platz vorhanden wäre.

1 tl;dr

Aufgabentyp „Formelvergleich“ nutzen; Expertenmodus der Lücke aktivieren; Zugriff auf Inhalt der aktuellen Lücke mit LEARNERRESPONSE, auf andere Lücken mit z.B. LEARNERRESPONSE_GAP_1 (Nummerierung nach Erstellungsreihenfolge, ID steht rechts unten im Bearbeitungsfenster)

Beispiel, bei welchem entweder die korrekte Antwort oder die Hälfte der in der ersten Lücke eingegebenen Zahl als korrekt gewertet wird (Verwendung von Komma oder Punkt als Dezimaltrennzeichen möglich; absolutes Fehlerintervall für noch korrekt bewertete Antworten von 0.01):

```
is(abs(parse_string(ssubst(".",",","LEARNERRESPONSE"))-
CORRECTRESPONSE)<=0.01)
or is(abs(parse_string(ssubst(".",",","LEARNERRESPONSE"))-
LEARNERRESPONSE_GAP_1/2)<=0.01)
```

2 Allgemeine Voraussetzungen

Die Berücksichtigung von Folgefehlern ist nur mit dem Aufgabentyp „Formelvergleich“ möglich. Im Vergleich zu einer normalen Aufgabe des Types „Berechnung“ müssen daher einige Funktionen manuell berücksichtigt werden. Das betrifft die Möglichkeit der Eingabe von Zahlen mit Komma oder Punkt sowie die Berücksichtigung von Rundungsfehlern mittels Toleranz der Antworten.

Die Beispielaufgabe und die Bearbeitungsfenster für die beiden vorhandenen Variablen sind auf der nächsten Seite zu finden.

Die korrekte Lösung für die erste Lücke ist die Variable {Ergebnis_GAP_1} (10), das korrekte Ergebnis für die 2. Lücke {Ergebnis_GAP_2} (5). Der mögliche Folgefehler bezieht sich also auf die zweite Lücke, in der korrekt die Zahl 5 eingetragen werden müsste, aber auch die Eingabe von Lücke 1, geteilt durch zwei als korrekt gewertet werden soll.

Der Quelltext zur Berücksichtigung von Folgefehlern muss im entsprechenden Feld (Bewertung MAXIMA) eingetragen werden. Das Feld ist nur im Expertenmodus sichtbar (blauer Link rechts im Bearbeitungsfenster).

3 Namen von Variablen und Lücken

Neben den frei wählbaren Variablen (im Reiter Variablen) werden für die Auswertung der Aufgaben weitere Variablen verwendet. Zum besseren Verständnis der Codebeispiele sind diese in der folgenden Tabelle kurz erklärt.

Variablenname	Beschreibung
LEARNERRESPONSE	vom Student in die Lücke eingetragener Text
CORRECTRESPONSE	korrekte Lösung einer Lücke (Feld Lösung)
MAXSCORE	maximale Punkte für eine Lücke
MINSCORE	minimale Punkte für eine Lücke (i.d.R. 0)

3.1 Zugriff auf Lücken

Auf den Inhalt der Lücke welche bearbeitet wird, kann mittels der Variable `LEARNERRESPONSE` zugegriffen werden. Um auf die Inhalte anderer Lücken in der gleichen Aufgabe zuzugreifen, wird an die Variable z.B. ein `_GAP_1` angefügt. Die Lücken sind nummeriert, nach der Reihenfolge wie sie erstellt wurden (nicht die Reihenfolge in der Aufgabe). Den Namen einer Lücke kann man herausfinden, indem man im Bearbeitungsmodus auf die Lücke klickt. Unten rechts steht dann fett geschrieben z.B. „ID: GAP_2“. Auf den Inhalt dieser Lücke kann also mit `LEARNERRESPONSE_GAP_2` zugegriffen werden.

4 Beispiele ohne Folgefehler

Die folgenden Beispiele ohne Folgefehler zeigen die verschiedenen Möglichkeiten der Fehlertoleranz und erklären, welche Befehle zur Auswertung durchgeführt werden.

4.1 ohne Folgefehler, ohne Fehlertoleranz

```
is(parse_string(ssubst(".",",","LEARNERRESPONSE"))=CORRECTRESPONSE)
```

In der Antwort des Studenten werden alle Kommas durch Punkte ersetzt. Nur bei exakter Übereinstimmung wird die Antwort als korrekt gewertet.

4.2 ohne Folgefehler, absolute Fehlertoleranz

```
is(abs(parse_string(ssubst(".",",","LEARNERRESPONSE"))-CORRECTRESPONSE)<=0.01)
```

Ist die (nicht vorzeichenbehaftete) Differenz zwischen der korrekten Lösung und der Antwort des Studenten kleiner gleich 0.01, wird die Antwort als korrekt gewertet.

4.3 ohne Folgefehler, relative Fehlertoleranz

```
is(abs(parse_string(ssubst(".",",","LEARNERRESPONSE"))-CORRECTRESPONSE)<=abs(0.01*CORRECTRESPONSE))
```

Bei einer Abweichung kleiner gleich 1% wird die Antwort als korrekt gewertet.

5 Beispiele mit Folgefehler

```
is(<Vergleich_1>)  
or is(<Vergleich_2>)  
or is(<Vergleich_3>)  
...
```

Wenn <Vergleich_1>, <Vergleich_2> oder <Vergleich_3> korrekt ist, wird die Antwort als korrekt gewertet. Die unterschiedlichen Lösungen/Vergleiche werden mit dem logischen Operator `or` verknüpft. Es können beliebig viele Vergleiche verknüpft werden. In einem Vergleich wird die Lösung des Studenten einmal mit der korrekten Lösung verglichen (<Vergleich_1>). Alle weiteren Vergleiche sind mögliche Folgefehler, in denen die korrekte Lösung durch Varianten mit falschen Zwischenergebnissen ersetzt werden muss.

5.1 ohne Fehlertoleranz

```
is(parse_string(ssubst(".",",",", "LEARNERRESPONSE"))=CORRECTRESPONSE)  
or is(parse_string(ssubst(".",",",", "LEARNERRESPONSE"))=  
    LEARNERRESPONSE_GAP_1/2)
```

In der zweiten Zeile (für den Folgefehler) wurde `CORRECTRESPONSE` ersetzt. Dadurch wird nun auch die Hälfte der ersten Lücke als korrekt gewertet.

5.2 absolute Fehlertoleranz

```
is(abs(parse_string(ssubst(".",",",", "LEARNERRESPONSE"))-  
    CORRECTRESPONSE)<=0.01)  
or is(abs(parse_string(ssubst(".",",",", "LEARNERRESPONSE"))-  
    LEARNERRESPONSE_GAP_1/2)<=0.01)
```

5.3 relative Fehlertoleranz

```
is(abs(parse_string(ssubst(".",",",", "LEARNERRESPONSE"))-  
    CORRECTRESPONSE)<=abs(0.01*CORRECTRESPONSE))  
or is(abs(parse_string(ssubst(".",",",", "LEARNERRESPONSE"))-  
    LEARNERRESPONSE_GAP_1/2)<=abs(0.01*LEARNERRESPONSE_GAP_1/2))
```

6 Beispiele Teilpunkten bei Folgefehler

Für die folgenden Beispiele muss im Feld „Bewertung (MAXIMA)“ „Punkte“ statt „Richtig/-Falsch“ ausgewählt werden.

```
if is(<Vergleich_1>) then MAXSCORE
elseif is(<Vergleich_2>) then float(MAXSCORE*0.8)
elseif is(<Vergleich_3>) then float(MAXSCORE*0.7)
else MINSORE
```

Die Möglichkeit nur Teilpunkte zu vergeben, ist ebenso ohne, mit absoluter oder mit relativer Fehlertoleranz möglich. Es besteht die Möglichkeit, abhängig von der Antwort unterschiedlich viele Punkte zu geben. Wenn <Vergleich_1> zutrifft, gibt es die volle Punktzahl, trifft <Vergleich_2> zu (z.B. ein Vorzeichenfehler) gibt es nur 80% der möglichen Punkte, bei <Vergleich_3> entsprechend 70%. Trifft keiner der Vergleiche zu, so wird die minimale Anzahl von Punkten vergeben (0 Punkte).

6.1 absolute Fehlertoleranz

```
if is(abs(parse_string(ssubst(".",",",", "LEARNERRESPONSE"))-
CORRECTRESPONSE)<=0.01) then MAXSCORE
elseif is(abs(parse_string(ssubst(".",",",", "LEARNERRESPONSE"))-
LEARNERRESPONSE_GAP_1/2)<=0.01) then float(MAXSCORE/2)
else MINSORE
```