

Christina Klüver: Online-Lernsystem für Mathematik mit *Computational Intelligence-Methoden*

Durch den Einsatz von Computational Intelligence-Methoden, nämlich einem Booleschen Netz und einem Self-Enforcing Network (ein selbstorganisiert lernendes neuronales Netzwerk)¹ wurde ein Prototyp implementiert, in dem Lernende Mathematikaufgaben webbasiert in JACK (ein automatisiertes Lehrsystem)² lösen und überprüfen können.

Das Besondere an dem Verfahren ist, dass die Lernenden zu Beginn auch die Lösungswege gemäß einem Lösungsalgorithmus angeben sollen, die von dem Self-Enforcing Network (SEN) analysiert werden. Die Eingaben werden in dem Booleschen Netz (BN) anschließend dynamisch visualisiert, d.h. jeder Lösungsschritt wird *nacheinander* angezeigt. Im Falle, dass der Lösungsschritt korrekt ist, ändert sich die Farbe des Knoten in grün, entsprechend in dunkelrot, sofern ein Fehler vorliegt.

Besonders wichtig ist, dass bei unbekanntem Eingaben das System nicht ausgibt, dass es sich um eine falsche Lösung handelt, sondern lediglich, dass die Lösung nicht eingeordnet werden kann. Dadurch werden Frustrationen vermieden im Fall z.B. kreativer Lösungen.

Die Zielsetzung wurde sowohl in technischer als auch in inhaltlicher Hinsicht erreicht, wie im Folgenden gezeigt wird (auf technische Details wird verzichtet):

In Abb. 1 wird zunächst das System aus Anwendersicht gezeigt; die Lernenden erhalten die Aufgabenstellung (in diesem Fall zum *Aufgabentyp* „Anwendung der pq-Formel“ gehörend), ohne anzugeben, wie viele Schritte zu Lösung der Aufgabe benötigt werden. Die Anwender können die Lösungen direkt im vorgesehenen Feld oder mit Hilfe des Formeleditors eingeben.

COBASC-JACK

Aufgabe "Aufgabe 1"
Aufgabenbeschreibung: Lösen Sie bitte die quadratische Gleichung $3x^2-30x+70=-5$ unter Verwendung der pq-Formel.

Schritt 1

Weiterer Schritt **Lösung einreichen**

Calculator Interface:

+	-	·	^	√	cos()	√
<	≤	=	≥	>	sin()	√
π	e	i	∞	tan()	{ }	
-		!	e	ln()	log(10,)	

Abb. 1: Anwendersicht in JACK

¹ Das SEN wurde von der Forschungsgruppe CoBASC entwickelt

² JACK wurde am Lehrstuhl „Spezifikation von Softwaresystemen“ entwickelt

In der Ergebnisansicht werden die einzelnen Schritte im Lösungsalgorithmus nacheinander angezeigt. In Abb. 2 wird visuell dargestellt, dass alle Lösungsschritte korrekt waren, bis auf das Endergebnis:

Ergebnisübersicht

Alle Schritte sind richtig bis auf das Ergebnis. Bitte überprüfen.

Rahmenfarbe	Füllfarbe	Strichstärke
Zugehörigkeit zum Pfad der Musterlösung	Ergebnis des eingegebenen Schritts	Schrittfolge der abgegebenen Lösung
korrekter Pfad	korrekt	Abfolge der Lösung
sonstiger Schritt	unbekannt	Übliche Schrittfolge
	falsch	

Schritt 1
 $3x^2 - 30x + 75 = 0$

Schritt 2
 $x^2 - 10x + 25 = 0$

Schritt 3
 $x_1 = -\frac{-10}{2} + \sqrt{\left(\frac{-10}{2}\right)^2 - 25} \wedge x_2 = -\frac{-10}{2} - \sqrt{\left(\frac{-10}{2}\right)^2 - 25}$

Schritt 4
 $x = 3$

Abb. 2 Ergebnisübersicht: Auf der rechten Seite zeigt die Legende, wie die Visualisierung zu interpretieren ist; die eigenen Lösungen werden ebenfalls angezeigt, wodurch der Fehler direkt identifiziert werden kann.

Wurde ein Lösungsschritt ausgelassen, wird dies von SEN erkannt und entsprechend dem BN übertragen (Abb. 3). An der Darstellung kann erkannt werden, dass die Struktur und die Visualisierung im BN individuell angepasst wird:

Aufgabenbeschreibung: Lösen Sie bitte die quadratische Gleichung $2x^2 + 4x = 70$ unter Verwendung der pq-Formel.

Ergebnisübersicht

Ihre Abgabe ist korrekt. Sie haben allerdings vergessen die "Normalisierung" anzugeben. Geben Sie beim nächsten Mal bitte **alle** Schritte ein. (Konfidenz: 100%)

Rahmenfarbe	Füllfarbe	Strichstärke
Zugehörigkeit zum Pfad der Musterlösung	Ergebnis des eingegebenen Schritts	Schrittfolge der abgegebenen Lösung
korrekter Pfad	korrekt	Abfolge der Lösung
sonstiger Schritt	unbekannt	Übliche Schrittfolge
	falsch	

Schritt 1
 $2x^2 + 4x - 70 = 0$

Schritt 2
 $x_1 = -\frac{2}{2} + \sqrt{\left(\frac{2}{2}\right)^2 + 35} \wedge x_2 = -\frac{2}{2} - \sqrt{\left(\frac{2}{2}\right)^2 + 35}$

Schritt 3
 $x_1 = 5 \wedge x_2 = -7$

Abb. 3 Ergebnis einer korrekten Lösung mit einem fehlenden Lösungsschritt

In Abb. 4 schließlich handelt es sich um eine eingereichte Lösung, die eher dem Prinzip „Versuch und Irrtum“ entspricht. Das Besondere in diesem Fall ist, dass im SEN und entsprechend im generierten

BN erkannt wird, dass ein dem System unbekannter Lösungsschritt vorliegt und das angegebene Endergebnis einer anderen Aufgabe entspricht.

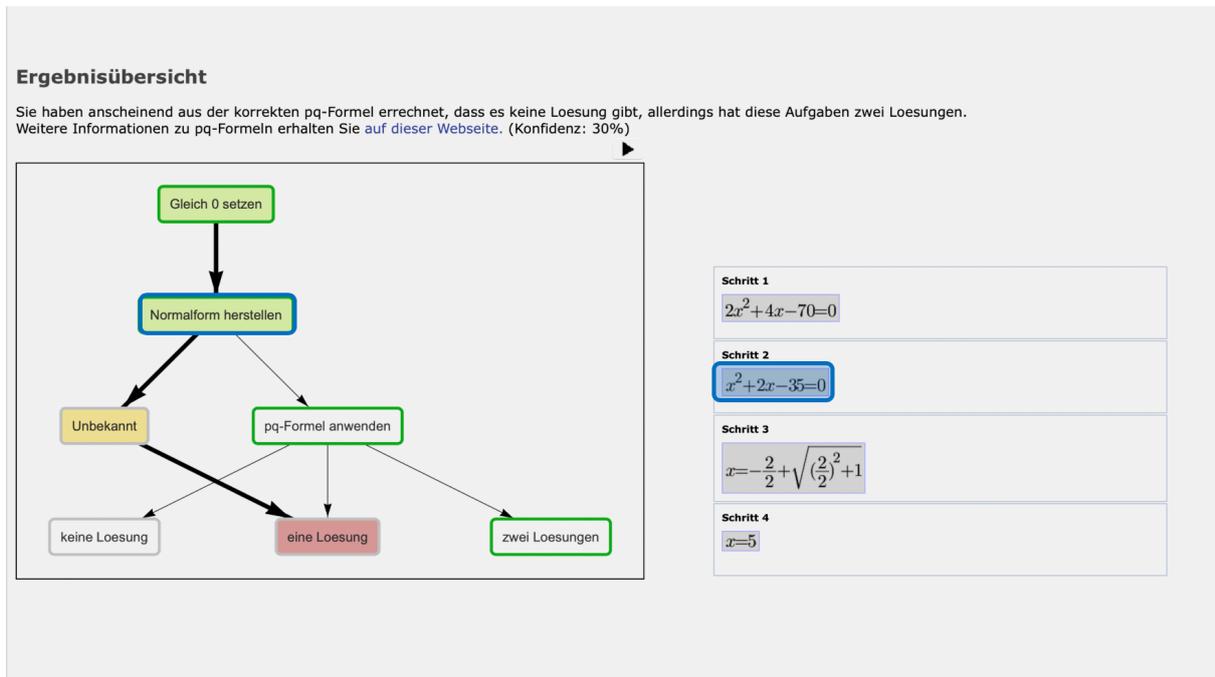


Abb. 4 Visualisierung einer fehlerhaften Lösung. In diesem Fall wurde eine „unbekannte“ Lösung eingegeben. Auf Grund der eingereichten Lösungen wird hier der gesamte Lösungsalgorithmus angezeigt.

Aktuell wird die Option implementiert, die Zeit, die für die Lösung einer Aufgabe benötigt wird, zu erfassen und die Anzahl der Klicks auf angebotene Hilfestellungen. Damit sollen die individuellen Probleme bzw. Wissenslücken eruiert werden.

Zusätzlich wird überprüft, um was für einen Fehler es sich handelt, um Folgefehler bei den eingereichten Lösungen ebenfalls zu berücksichtigen.

Nicht intendiert, im Rahmen der Implementierung jedoch von den Programmierern konzipiert und umgesetzt, ist die Möglichkeit, dass Lehrende ihre eigenen Lösungsalgorithmen und die konkreten Aufgaben im eigenen PC eingeben können, ohne über Programmierkenntnisse verfügen zu müssen. Diese Option wurde durch eine spezielle Schnittstelle zur Webplattform JACK und zu dem Booleschen Netzwerk realisiert, so dass die Eingaben der Lehrenden automatisch im BN generiert werden. Das BN generiert seinerseits die Lerndaten für das SEN, das zunächst die angegebenen Aufgaben sowie deren Lösungen lernt; anschließend werden die Eingaben der Lernenden, wie bereits gezeigt, in JACK ausgewertet.

Abb. 5 zeigt wie die Lösungsalgorithmen und Aufgaben im BN abgebildet werden können:

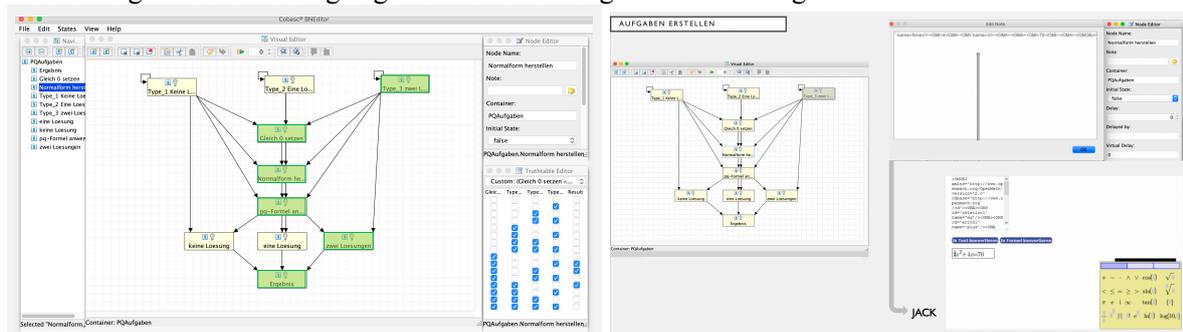


Abb. 5 Auf der linken Seite wird im BN der Lösungsalgorithmus festgelegt, auf der rechten Seite werden über eine Schnittstelle zu JACK die konkreten Aufgaben erstellt.

Die Generierung des SEN wird in Abb. 6 dargestellt. Die Aufgaben werden in die sog. „Semantische Matrix“ des SEN übertragen, die von SEN gelernt werden. Im SEN hat ein Lehrender die Möglichkeit, zusätzlich mögliche und bekannte Fehler zu berücksichtigen und darüber hinaus Hilfestellungen zu definieren:

Abb. 6 Semantische Matrix des SEN, die vom BN generiert wurde, sowie Ergänzungen durch Lehrende. Im Falle eines falschen Ergebnisses wird dies nach dem Lernprozess angezeigt.

Das SEN verfügt über verschiedene Visualisierungsmöglichkeiten, von denen zwei in Abb. 6 unten rechts zu sehen sind. Bei einer vollständig richtigen Lösung ist der sog. Aktivierungswert (Ranking) 1.0 und die Distanz 0.0 (in der zweiten Visualisierung). In diesem Fall zeigt das SEN an, um welchen Aufgabentyp es sich handelt und dass bei einer entsprechenden Eingabe (eine Lösung von Lernenden) nur das Ergebnis nicht korrekt ist, da die Aktivierung um lediglich 0.2 geringer als der Höchstwert ist und die Distanz um 0.28 abweicht. Diese Informationen sind wichtig für Lehrende, um zu überprüfen, ob die angegebenen Lösungen korrekt sind.

Entsprechend werden von SEN die jeweiligen Eingaben der Lernenden analysiert und die Ergebnisse dem BN übertragen. In Abb. 2 wurde gezeigt, dass alle eingereichten Lösungen korrekt waren, bis auf das Endergebnis. In diesem Fall wurde also der Knoten im BN, der für die Darstellung des Endergebnisses steht, nicht aktiviert („wahr“), sondern als „falsch“ dargestellt.

Das System wurde in eigenen Veranstaltungen evaluiert und in einem Webinar präsentiert: „Mathematik in MINT-Fächern: Kreative Lehr-Lern-Konzepte digital umgesetzt“, veranstaltet von eCULT+ (Leibniz Universität Hannover). In Kürze wird die Präsentation online verfügbar sein.

Da Mathematikprobleme in vielen Disziplinen auftreten, ist es geplant, dass die Aufgaben aus anderen Fächern, wie zum Beispiel Chemie oder Physik aufbereitet werden, wodurch jedoch das System für die speziellen Erfordernisse, beispielsweise hinsichtlich der verwendeten Symbole, angepasst werden muss. Insbesondere ist es wünschenswert, andere Entwicklungen, die im Rahmen der Fellowships erfolgt sind, zu integrieren.

Die Kombination zwischen Booleschen Netzwerken und Self-Enforcing Networks kann zudem für Probleme eingesetzt werden, bei denen es um komplexe Abläufe geht (wie zum Beispiel Workflows, die im Booleschen Netz abgebildet werden), und die zugleich eine inhaltliche Überprüfung der Abläufe erfordert (wie beispielsweise die Kenntnis von Normen, Angaben von korrekten

Mischverhältnissen bei Chemikalien, Materialien etc.). Somit kann die Entwicklung in jeder Lehr-/Lernsituation eingesetzt werden, in denen ein bestimmter Ablauf eingehalten werden soll, der zugleich eine korrekte bzw. vollständige inhaltliche Bearbeitung der einzelnen Schritte, Komponenten etc. erfordert.