Matrix Multiplication Tutor

Using CTAT with SimStudent and Jess to create a cognitive tutor

Liz Blankenship
Clearsighted, Inc. and Iowa State University
PSLC Summer School 2007
Task Domain

How can we teach students to multiply matrices?

\[
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
\times
\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
= ?
\]
About Me

• B.S. in Computer Science and Math

• Currently employed as a cognitive modeler at Clearsighted, Inc., which has a partnership with Carnegie Learning, enabling cognitive modeling using the Cognitive Tutor SDK
Project Goals

• Learn to use CTAT and associated tools
• Help test SimStudents on a new problem
• Think about cognitive modeling from another perspective
Interface Design

Using NetBeans, designing a Java interface was easy.
CTAT makes creating an example tracing tutor easy, but it is time consuming to create many problems.
Cognitive Tutor Design

To create a better tutor that can be reused for multiple problems, a cognitive tutor (or model-tracing tutor) is necessary.

Steps for creating a model-tracing tutor:

1) Use Simulated Students (SimStudents) with CTAT to automatically create predicates

2) Write any other necessary production rules by hand in JESS
Using SimStudent, the correct steps to solve the matrix multiplication problem were demonstrated.
Auto-generated Rules

SimStudents made rules...

some of them correct...

(defrule add-simp
    (matrix-unsimplified nil) (matrix-simplified nil) (matrix-answer nil)  )
?var766 <- (table (columns $?m783 ?var767 $?) (rows ))
?var767 <- (column (cells $?m782 ?var768 $?))
?var768 <- (cell (name ?foa0) (value ?val0&~nil)   )
    (matrix-unsimplified nil) (matrix-simplified nil) (matrix-answer nil)  )
?var759 <- (table (columns $?m795 ?var760 $?) (rows ))
?var760 <- (column (cells $?m794 ?var761 $?)  )
?var761 <- (cell (name ?selection) (value ?input&nil)   )

(test (distinctive ?var768 ?var761))
(test (same-row ?var768 ?var761))
(test (same-column ?var768 ?var761))

=>

(bind ?input (eval-arithmetic ?val0))
(here-is-the-list-of-foas ?foa0)
(predict-algebra-input ?selection UpdateTable ?input )
(modify ?var761 (value ?*sInput*)))

This part correctly identifies a cell (say, with 24+10 in it) and the cell that the simplified equation (34) should be entered into

This rule correctly identifies which cells need to be used in multiplication but you can see by the use of add-term and no mult-term that it inferred the wrong rule.

With more trials SimStudent would generate the correct rules

others, not so much...

(defrule 1-1-unsimplified
  ...
  ;; cut the code for identification of the cells in the row and column
  =>
  (bind ?input (add-term ?val3 ?val4))
  (here-is-the-list-of-foas ?foa0 ?foa1 ?foa2 ?foa3)
  (predict-algebra-input ?selection UpdateTable ?input )
  (modify ?var31 (value ?*sInput*)))

This part correctly identifies a cell (say, with 24+10 in it) and the cell that the simplified equation (34) should be entered into
Problems Encountered

• Running SimStudents required a lot of debugging effort since it had not been thoroughly tested outside of example circumstances

• The learning curve for writing Jess production rules was too steep under the time constraints

• Although SimStudents was on the right track to creating production rules for multiplying matrices, its interface with CTAT prevented me from being able to test the production
Outcomes

• Gained confidence in creation of example-tracing tutors with CTAT
• Gained basic understanding of SimStudents, but it needs more work
• Learned the basics of Jess for implementation of production rules
• Experienced a different perspective of cognitive modeling
Thanks

Thanks to the PSLC and CMU for hosting this summer school, and an extra thanks to Noboru Matsuda, Vincent Aleven, and Jonathan Sewall for many hours of help and interesting discussion in the lab.