

Elaborating Models of Algebraic Expression-Writing

Mary A. Mark, Kenneth R. Koedinger, and William S. Hadley

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213
mary.mark@cs.cmu.edu, ken.koedinger@cs.cmu.edu, bh23@andrew.cmu.edu

Abstract. This paper discusses the refinement of the algebraic expression-writing rules for the PAT Algebra I tutor, between 1992 and 1997. Direct observation of students in class, step-by-step PAT tutor protocols tracing student behavior, and statistical analysis of protocol data, have all informed our understanding of students' skills, and our refinement of this part of PAT's production model.

1 Introduction

Since 1992, the Pittsburgh Advanced Cognitive Tutor Center has been developing an intelligent tutoring system for algebraic problem solving, the PAT Algebra I tutor. First used in one grade nine algebra classroom, the PAT Algebra I tutor is now used in about 40 schools. The tutor has been designed to complement the PUMP algebra curriculum used in grade nine algebra classes (cf. [3]). Both are designed to address math reform concerns of the National Council of Teachers of Mathematics [8]. They focus on the mathematical analysis of real world situations, and the use of multiple representations for problem-solving [6].

Much of the PAT development effort has focused on breadth, providing materials for a full curriculum year. The current tutor curriculum includes over 15 lessons of multiple sections, drawing on over 400 problem situations. The complete production model contains 56 production sets, and 623 production rules.

As reported elsewhere, the overall impact of PAT tutoring is positive [6], [7]. We have also examined specific aspects of PAT tutoring in some detail. This paper describes the development of the PAT production rules for expression-writing from 1992 to 1997. A brief description of the PAT tutor's interface and behavior places expression-writing in the context of other algebraic problem solving skills modeled in PAT. We then discuss development of the expression-writing model from our initial modeling as a single skill to our current characterization of expressions in terms of features of operators and arguments which cause students difficulty. These changes in the model are supported by formative analyses of students' expression-writing success. These analyses also provide evidence that students are learning to write expressions as they use the tutor. Finally, we suggest future directions for development.

2 The PAT Algebra I Tutor

Daily, people deal with situations that draw on algebraic and quantitative reasoning skills. Checking the amount of a paycheck, estimating the discount applied to an on-sale purchase, and choosing between long-distance telephone services, are three examples of real-world situations in which algebraic skills are useful. The PAT tutor tries to help people to learn and apply mathematical problem-solving skills in the context of such real-life situations. To solve a PAT problem, students read a textual description of a situation with related questions. They represent the situation using tables, graphs, and symbols as tools to answer questions. Multiple representations are a major focus of the tutor.

Figure 1 shows a partial solution for a problem involving a single linear equation, from Lesson 1 of the PAT Algebra I curriculum for the 1996-1997 school year. The lesson focuses on constructing a table by finding solutions to questions and writing an expression.

The screenshot displays the PAT Algebra I Tutor interface, which is divided into several windows:

- Problem Statement:** Contains the text: "The Sky Rink is a gear-round skating rink. It is located on top of a 16-story skyscraper in downtown Boston, and has a terrific view. When you rent a pair of skates there, for \$1.50, you must also pay a rental fee." followed by five questions:
 - If you spent two hours skating at Sky Rink, how much would it cost you?
 - If you spent the afternoon - 3 hours - at Sky Rink, how much would it cost?
 - If you spent the entire day - 7 1/2 hours - at Sky Rink, how much would it cost?
 - If you spent \$27.30 at Sky Rink, how long did you skate?
 - How long did you skate if you spent \$11.50 at Sky Rink?
 Below the questions, it says: "For the formula, define a variable for the time at Sky Rink and use this variable to write a rule for the cost of going to the Sky Rink."
- Worksheet:** A table with columns "HOURS" and "DOLLARS" under the heading "TIME SKATING COSTS OF SKATING". The first column is labeled "FORMULA" and contains numbers 1 through 5. The second column is labeled "HOURS" and contains 2, 3, 7.5, 5, 5. The third column is labeled "DOLLARS" and contains 19.5, 27.5, 43.5, 27.5, 11.5.

	HOURS	DOLLARS
1	2	19.5
2	3	27.5
3	7.5	43.5
4	5	27.5
5	5	11.5
- Pattern Finder:** Contains the following text:

How would you calculate THE COST OF SKATING AT SKY RINK for 2 hours?
 $2 \times \$3.5 = 19.5$

How would you calculate THE COST OF SKATING AT SKY RINK for 3 hours?
 $3 \times \$3.5 = 27.5$

How would you calculate THE COST OF SKATING AT SKY RINK for 4 hours?
 $4 \times \$3.5 = 35.0$

Write an expression which describes your calculations, using a variable $Y = \$3.5 \times X$.
- Messages:** Contains the text: "You have entered an equation for calculating the cost of skating at sky rink in the Pattern Finder window. What expression does Y equal?"

Fig. 1. The PAT Algebra I Tutor, Lesson 1

The top-left corner of the tutor screen shows a problem situation: the cost of skating at Sky Rink, based on a flat fee for renting skates, and a per-hour fee for the time spent at the rink. Students investigate this situation using tools such as a spreadsheet (the "Worksheet" window) and a tool for identifying algebraic expressions by elaborating patterns (the "Pattern Finder" window).

Students construct the Worksheet (lower-left of Figure 1) by identifying relevant quantities in the situation, labeling columns, entering appropriate units, writing an algebraic expression, and answering the questions in the problem description. The Pattern Finder window (center-right) can be used at any time. Students who have difficulty writing expressions are encouraged to examine progressions of concrete instances of arithmetic expressions, and look for underlying patterns indicating the form of the expression. The Pattern Finder design was informed by previous prototype research in which students who were encouraged to work out concrete instances prior to expression-writing performed better on expression-writing during training, and learned more from pre-test to post-test, than students in a control group who were encouraged to write the expression first and then answer concrete instances by evaluating the expression [5].

On average, students spend about twelve minutes solving such a problem. The tutor monitors their activities and provides feedback through *model tracing* (cf. [1]) to individualize instruction. With both correct and buggy rules, the tutor can often indicate what is wrong with the answer or suggest a better alternative.

Another way of individualizing instruction is to offer help on request. The tutor selects help messages by using the production system to identify a desirable next activity, based on the student's current focus of activity, the overall status of the student's solution, and internal knowledge of interdependencies between problem-solving activities. Multiple levels of help provide more detailed information for repeated requests. The "Messages" window in Figure 1 shows the result of a help request. The current focus of attention is the worksheet cell for the expression, selected under the column entry for 'costs of skating'. Based on information about the problem state, especially other tasks completed, the student is encouraged to relate the Pattern Finder solution to the desired worksheet expression: "You have entered an equation for calculating the cost of skating at Sky Rink in the Pattern Finder window. What expression does Y equal?" If the Pattern Finder had not been used, different help would be given, prompting the direct construction of an expression based on information from the problem statement e.g. "Read the problem statement to see how to calculate the cost of skating at Sky Rink. The expression for the cost of skating at Sky Rink depends on the number of hours skated." or, more explicitly, "To write an expression for the cost of skating at Sky Rink, enter 8 times the number of hours skated, plus 3.50. Use a variable to represent the number of hours skated."

PAT also monitor students' learning across problems, supporting learning through *knowledge tracing*. The Skillometer window displays the tutor's assessment of the student's acquisition of problem solving skills. The tutor can identify individual areas of difficulty and present problems targeting skills which are not mastered [2]. Self-paced advancement through sections and lessons of the curriculum is also accomplished through knowledge tracing.

For model tracing and knowledge tracing to be effective, the cognitive model must capture aspects of the problem that the student finds difficult, and differentiate these from aspects of the problem which the student finds easy. Otherwise, the tutor will be unable to focus remediation on those areas where

the student needs work. The model must also differentiate easy from difficult skills to ensure that a student is not being advanced past areas of difficulty without detection. The cognitive model is the heart of the tutor: its ability to make such discriminations is essential.

3 Models of Expression-Writing

Writing algebraic expressions is an important and difficult skill for students. The ability to capture the complexities of a mathematical relationship and represent them in an elegant algebraic form is seen by teachers as an indicator of students' understanding of the specific situation and the underlying mathematical relationships. The ability to translate a quantitative problem situation into algebraic symbols is important for using real-world tools to solve real-world problems [5].

To experts, writing an algebraic expression like " $8.0 * x + 3.50$ " seemed a fairly obvious process: you read the problem statement, identified the changing quantity and rate of change in the problem, checked for an initial value, and put the appropriate numbers in the appropriate places in the formula. Novice students entered many unexpected answers. Through observations in class and examination of protocol data we examined what students were doing and tried to determine what they were and were not learning about expression-writing.

The data presented here was collected in the 1996-1997 school year, at Langley High School, Pittsburgh, PA. Langley is a typical urban school, mixing lower and middle class families. About 50 % of students are from single parent families, 60% come from families with sufficiently low incomes to qualify for the free reduced lunch program, and 50% are minority students. Student attendance is volatile, with high absenteeism (30%) and student mobility (20%).

Data collection occurred while developing and deploying the tutor in grade nine classes as part of the regular class schedule. The PAT tutor saved protocol files recording student actions and productions fired. The tutor curriculum was designed to meet classroom constraints for instructional time available and material to be covered. This client-centered approach meant that the curriculum was not ideally suited to testing some of our hypotheses about expression-writing.

We have chosen to present the 1996-1997 Lesson 1 data, as representative of the abilities of incoming grade nine students; and to discuss it in terms of cognitive models from present and previous years, to illustrate the expressiveness of the cognitive models developed from 1992 to 1997. Reanalyzing one data set rather than data from each year also excludes the possibility that different analyses reflect changes in the population, teacher, or tutor versions of different years, unrelated to changes in the expression-writing model.

1992-1994: In its first year the PAT tutor was used in selected classes for short periods of time. Development focused on introducing new tutor components and increasing curriculum coverage. Model tracing was used within sections, but knowledge tracing was not yet used. In the second year, we introduced knowledge tracing, and began to analyze student data and review our design. Development involved many target skills besides expression-writing, including

numerical solution of result-unknowns and start-unknowns, quantity and unit identification, table building, graphing skills, equation solving, etc.

Initially, expression-writing was modeled as a single skill. Students were not very successful at writing expressions. They succeeded on their first attempts at writing expressions only 39% of the time, across all problems in Lesson 1. In comparison, when solving result-unknown questions, e.g. "If you skated at Sky Rink for two hours, how much would it cost you?", students were correct 71% of the time on their first attempt at answering the first result-unknown in a problem.

Furthermore, when treated as a unitary skill, student profiles and accuracy curves for groups of students showed that students did not succeed consistently in writing expressions. In the following student profile, "Student AB" succeeds in 11 of 21 first attempts at writing an expression. Zero values indicate failures; -1 indicates help requests; and positive values indicate the elapsed time to enter the correct answer measured in seconds. Successes and failures intermix apparently randomly, although more successes occur in the second half of the curve.

```
("Student AB" (IDENTIFY-EXPRESSION WORKSHEET-TUTOR-MODE) 11 21
 38.016 10.945 0.000 -1.000 0.000 0.000 10.337
-1.000 11.189 0.000 11.172 0.000 0.000 0.000
61.257 0.000 19.762 8.839 47.127 40.550 18.833)
```

1994-1995: Based on classroom and tutor protocols, we refined our model to categorize expressions based on their mathematical formula. Students were required to write an expression of the form $MX+B$, where M is the slope and B the y -intercept. Depending on the sign of the slope and intercept values, and whether the intercept was zero or non-zero, six different forms were identified: MX , $MX+B$, $MX-B$, $-MX$, $-MX+B$, and $-MX-B$. Forms involving negatives were rare and were reserved for later lessons involving graphing. Lesson 1 therefore focuses on the three most common variants: MX , $MX+B$, and $MX-B$.

Reanalysis of the student data suggests that rewriting production rules to capture differences between forms helps to more accurately identify students' strengths and weaknesses. The previous accuracy profile becomes far more comprehensible: "Student AB" demonstrates considerable success with MX forms, but succeeds only 50% of the time when writing expressions with positive non-zero intercepts ($MX+B$ forms), and 25% of the time when dealing with negative intercepts ($MX-B$ forms). The student succeeds more often in the second half of each curve, suggesting that learning may be occurring.

```
("Student AB" (FORM=MX IDENTIFY-EXPRESSION) 7 11
 10.945 0.000 10.337 -1.000 11.189 0.000
 11.172 0.000 47.127 40.550 18.833)
("Student AB" (FORM=MX+B IDENTIFY-EXPRESSION) 3 6
 38.016 0.000 0.000 61.257 0.000 8.839)
("Student AB" (FORM=MX-B IDENTIFY-EXPRESSION) 1 4
-1.000 0.000 0.000 19.762)
```

A single factor ANOVA statistically tested the relationship of form to student success (Figure 2). Twenty-four problems from Lesson 1 were examined, not including a demonstration problem. Thirteen problems were of form MX, seven were MX+B, and four were MX-B. The success rate for writing the expression for each problem was defined as the percentage of students who correctly wrote an expression for that problem on their first attempt at doing so. From 16 to 63 students attempted each problem, averaging 42.375 students per problem.

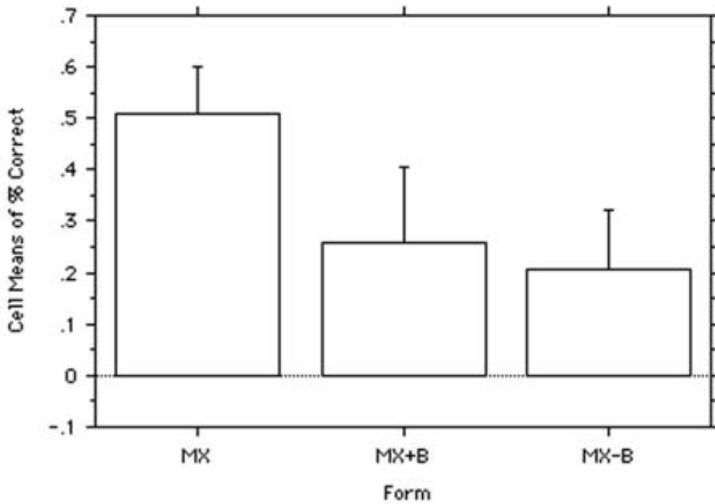


Fig. 2. Expression-Writing Success For Problems Categorized By Form

The mean success rate for MX problems was 50% ($sd = .149$); for MX+B, 25% ($sd = .159$); and for MX-B, 20% ($sd = .072$). Differences between forms were statistically significant ($F(2, 22) = 11.0, p < .0006$). Scheffe's S post-hoc comparisons indicated that MX problems were easier than MX+B ($p < .004$) and MX-B problems ($p < .005$). No significant difference was reported between MX+B and MX-B problems. Unfortunately, the data available for MX-B problems was limited, as only four MX-B problems were included in Lesson 1.

At the end of the 1994-1995 school year, the production model was refined to categorize expressions in these categories, and to use these production rules in knowledge-tracing when assigning new problems. Lessons and sections were reorganized to vary students' exposure to different forms. Buggy productions were added to identify cases where a student was omitting or mis-identifying the slope or intercept of the expression: e.g. "How many dollars do you start with when you calculate the cost of skating at Sky Rink?" Feedback was designed to prompt students to distinguish between these parts of the expression, for example, if the two numbers were reversed, "Which number should be the slope and which should be the intercept?" Incorrect entry of number signs was also

diagnosed by buggy productions: e.g. “Does the height of a tree increase or decrease as the time that the tree has been growing increases?”

1995-1996: Organizing expressions by form did not completely explain why students found some problems harder than others. When we compared problems in terms of expression-writing difficulty, there were still anomalies which were easier or harder than expected. We looked for features of difficult and easy problems and characteristic student errors indicating possible areas of difficulty.

Features of expressions which seemed to give students difficulty included large numbers, decimals, and fractions. These aspects of expression-writing were reified as: (1) the size of the numbers involved, operationally defined as small ($-100 < x < 100$) and large (greater than or equal to ± 100); and (2) the type of the number involved, with integer values categorized as simple numbers and non-integers as difficult numbers. An expression was classified as difficult if it included difficult numbers; and as large if it included large numbers. The production model and the tutor curriculum were modified to incorporate these categories into productions and knowledge-tracing.

It was also difficult for students to write expressions in which numbers had to be transformed. Textual descriptions of numbers such as “8 and one-quarter”, which could be written as either a decimal, “8.25”; or an expression, “8+1/4”, caused many errors. Students had difficulty with unit conversions even for such presumably familiar conversions such as entering “1/4 of an hour” as a value in minutes. Students failed sometimes because they did not realize that a conversion was needed, but more often because they did not know how to make the transformation. Asked directly how many minutes were in an hour, a student might answer, “50? 100?” Buggy productions were implemented to model-trace failures in conversion and provide feedback: e.g. “You need to calculate the rate of change in the expression for ‘the height of the river’ in feet, not in inches. Divide by 12 to get an amount in feet.” However, because problems involving textual conversions were hard to identify a priori, given the representation of problems in the tutor, conversions were not modeled explicitly for knowledge-tracing. Conversion is a desirable area for future research.

A regression analysis of all possible learning events in Lesson 1 for the 1996-1997 year suggests that this further categorization of expression-writing skill is useful in predicting student success and failure. A regression model was performed in which correctness on the first attempt at writing the expression was the dependent variable. The problem position (the n th problem assigned to a student, $n=1..20$), the algebraic form of the problem (MX, MX+B, or MX-B), and the difficulty of the numbers involved (integer or decimal) were the independent variables. The model yielded statistically significant effects of each of these variables ($R=.3168$). The regression equation is as follows:

$$\%Correct = 43.7 + 0.9 * POS - 28.0 * FORM2 - 34.8 * FORM3 - 16.0 * DEC \quad (1)$$

where POS is the problem position (1-20); FORM2 is 1 if the problem is an MX+B problem and 0 otherwise; FORM3 is 1 if the problem is an MX-B problem and 0 otherwise; and DEC is 1 if the problem has any decimal arguments and 0

if they are all integers. A possible learning event was defined as a student’s first attempt at writing an expression for a problem. At each possible learning event, a student could succeed or fail in writing the expression. The three effects can be described as follows (see Figure 3):

Expression form effect: ($p < .0001$) MX problems are easiest ($M = 44\%$), MX+B are harder by 28% ($M = 16\%$), MX-B are slightly harder by 35% ($M = 9\%$).

Number type effect: ($p < .0003$) Problems with integer arguments are easier ($M = 31\%$) and problems with decimal arguments are 16% harder ($M = 15\%$).

Learning effect: ($p < .03$) On average students are getting about 1% better on each expression writing experience. For example, as predicted by the model, students are about 44.6% correct on their first MX integer problem, but 61.7% correct by their twentieth MX integer problem. On the MX+B decimal problems, students start at 0.6% on the first problem and reach 17.7% by the twentieth problem.

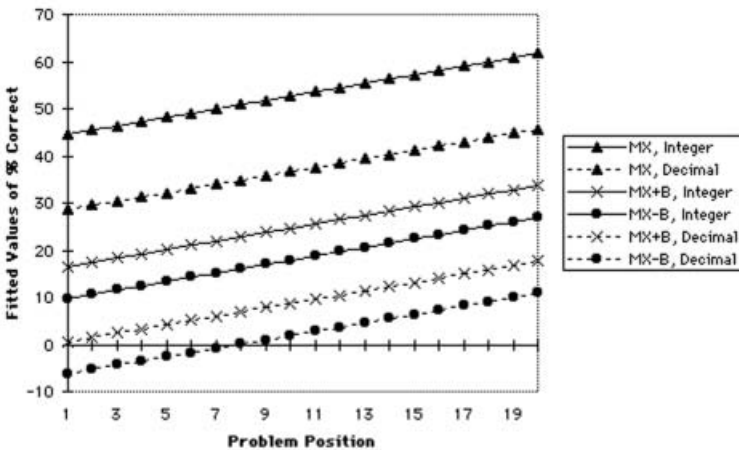


Fig. 3. Fitted Regression Analysis of Learning Events, by Form and Integer Difficulty

Interestingly, even though previous analysis did not show significant differences between MX+B and MX-B forms, defining the form in terms of MX, MX+B, and MX-B proved to be a stronger predictor of success in the regression than defining form in two categories, MX and MX+B (not shown). This suggests that sign is important in determining the difficulty of the form, as well as the presence or absence of a non-zero intercept. The number of problems with negative numbers in the curriculum should be increased to more effectively test this hypothesis. The second strongest indicator of student success on specific

problems was number type. Integers were easier to deal with than non-integer numbers. Again, this supported our hypothesis.

The other possible predictor of problem difficulty which we examined was number size. We did not find a significant effect of number size in the regression. Unfortunately, Lesson 1 included very few problems with large numbers, limiting our ability to examine this. Our original categorization may be either inadequate or inappropriate. One possibility is that “large” numbers like 2067959 are harder for students, while “large” numbers involving patterns or repeating digits, like 8000, are not much harder than “small” numbers like 80. Examination of the “large” problems in the 1996-1997 data set suggested this possibility, but better problem selection is needed to assess this properly.

The regression shows a small increase (approximately 1%) in the probability that students will get the n th+1 problem correct on their first attempt, compared to the n th problem. While this is evidence that students are learning as they work through the lesson, students are obviously still having a lot of trouble writing algebraic expressions for word problems after finishing this first lesson. One explanation for students’ difficulties is that these are disadvantaged urban students, with extremely poor background knowledge and skills. Incoming grade nine students at Langley scored near chance on an Iowa Algebra readiness test. Students are also exposed to many new skills at once in the first lesson: they have more opportunities to learn these skills as they work through the curriculum.

The regression analysis has potential implications for curriculum redesign. Assigning short sequences of problems with few remedial problems may mean that students don’t get enough exposure to master expression-writing. Presenting one type of problem at a time may lead to acquisition of superficial “rules” which do not generalize well [5], such as “pick the first number you see and multiply it by x ”. Mixing types of problems in longer sequences may better support learning.

Since students are substantially better at generating numerical answers for these problems (71%) than at writing abstract expressions (39%), students clearly are not using the expression to generate the result-unknown answer (cf. [5], [4]). Further investigation of students’ understanding of and ability to generate symbolic representations is highly desirable. At least when the quantities and relationships involved are fairly simple, students may understand the relationships in the problem situation, but not be able to generate symbolic representations of them. Deeper understanding of the relationships between skills, and the strategies that students use to solve problems, are also desirable, and may facilitate the development of tools to support the acquisition of more difficult skills by relating them to existing student knowledge.

4 Conclusions

Since 1992, we have been refining the PAT Algebra productions to more closely model students learning. Expression-writing skills have been a rich focus of development. Students must identify relevant operators, quantities, and units as they read the problem situation, and generate a symbolic representation

expressing the relationships involved in algebraic notation. In addition to determining whether a correct expression is entered, the tutor can now identify answers involving the presence, absence, and possible reversal of the slope and intercept values; the reversal of number signs, the use of incorrect operators, and failures to convert quantities involving units of measurement. In addition to its model-tracing capabilities, the tutor assigns remedial problems based on features of the expression such as the form of the equation and whether or not the numbers involved are integers or fractions and decimals. Data analysis indicates that these aspects of problem complexity are predictors of expression-writing difficulty and student success. The hypothesis that number size would relate to expression-writing difficulty was poorly tested and should be reexamined.

When form and number type are factored out in a regression analysis, evidence remains that students are learning to write expressions as they use the tutor, although the rate at which they learn is slow. It is important to place this result in context by remembering that expression-writing is one of many skills targeted by the PAT Tutor. Although the effect of Lesson 1 on students' acquisition of expression writing skills is modest, large overall effects of PAT's multiple lessons across a whole array of skills are heartening [6]; [7].

References

1. Anderson, J.R., Boyle, C.F., Corbett, A.T., Lewis, M.W.: Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, 42 (1990), 7-49. 526
2. Corbett, A.T., Anderson, J.R., Carver, V.H. and Brancolini, S.A.: Individual differences and predictive validity in student modeling. In A. Ram & K. Eiselt (eds.) *The Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum (1994) 526
3. Hadley, W.S.: *CMU PACT Center Mathematics Curriculum, 1997-1998*. Pittsburgh, PA: PACT Center, Carnegie Mellon University (1997) 524
4. Heffernan, N. T., Koedinger, K. R.: The composition effect in symbolizing: The role of symbol production vs. text comprehension, *Proceedings of the Nineteenth Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Erlbaum (1997) 532
5. Koedinger, K. R., Anderson, J. R.: Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. To appear in *Interactive Learning Environments*. 526, 527, 532, 532
6. Koedinger, K. R., Anderson, J.R., Hadley, W.H., Mark, M. A.: Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8 (1997), 30-43 524, 524, 533
7. Koedinger, K. R. Sueker, E.L.F.: PAT goes to college: Evaluating a cognitive tutor for developmental mathematics. In *Proceedings of the Second International Conference on the Learning Sciences*, Charlottesville, VA: Association for the Advancement of Computing in Education (1996) 524, 533
8. National Council of Teachers of Mathematics: *Curriculum and Evaluation Standards for School Mathematics*. Reston, VA: The Council (1989) 524