Towards Understanding Geometry Explanations

Octav Popescu, Kenneth R. Koedinger

Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
octav@cmu.edu
koedinger@cs.cmu.edu

Abstract

Previous studies using our PACT Geometry Tutor have shown that students learn mathematics more deeply in the same amount of time if they are asked not only to make correct problem solving steps but also to explain those steps (Aleven, Koedinger, and Cross 1999). In the current tutor, students enter the name of a geometry definition or theorem to explain a step, but our goal is for them to state these explanations in their own words. We are building a natural language understanding system to analyze such free-form explanations. We have identified a number of challenges posed by real student explanations and present an evaluation of our system's performance on a corpus of such explanations. Understanding in this tutoring domain requires more than figuring out the intentions of the student, but also to what extent the student has stated those intentions with sufficient mathematical precision. We argue that this kind of analysis cannot be performed with statistical methods alone, but requires a knowledge-based understanding system.

What's the Goal?

Cognitive tutors have been shown to be very effective in raising students' test scores in high-school algebra and other domains (Koedinger at al. 1997). However, a limitation of many current systems is that they teach "at the problem-solving level", meaning that they provide assistance in the context of problem solving, but engage students only indirectly in thinking about the reasons behind the solution steps. That is, they provide context-sensitive hints but they do not ask students to explain or justify answers, for instance, "why did you do this step?" or "what rule can you apply next and why?" or "what does this rule mean?" Students' understanding may improve if the tutor encourages them to think more consciously about such questions, and even more so, if the tutor can engage students in a dialog to help them improve their explanations. Understanding students' input expressed in natural language is key step in implementing a tutorial dialog around student explanations. Although some current intelligent tutoring systems like Autotutor (Wiemer-Hastings, Wiemer-Hastings, and Graesser 1999) or Circsim (Glass 1997) do have some natural language processing capabilities, they do not deeply process student explanations. We are building a system to do so.

We are building this understanding system in the context of the PACT Geometry Tutor, a computer-based "cognitive tutor" used two days a week as part of a complete high school geometry course. The PACT Geometry Tutor assists students in learning by doing as they work problems on the computer. As a kind of Cognitive Tutor (Anderson et al. 1995), this system is based on an underlying cognitive model, implemented as an ACT-R production system (Anderson and Lebiere, 1998), of both novice and ideal student knowledge. This model is used to monitor student performance and to provide instruction just when students need it and in a context that demands it. The tutor maintains a detailed assessment of each student's skills, which is used to select appropriate problems and determine pacing. In the fall of 2000, the PACT Geometry Tutor will be in regular use (two days per week) in about ten schools around the US. The tutor curriculum consists of six extensive units: Area, Pythagorean Theorem, Angles, Similar Triangles, Quadrilaterals, and Circles.

In the Angles unit (Aleven et al. 1999), students analyze geometric figures with given properties (e.g., parallel lines or isosceles triangles) and solve problems by inferring measures of angles and segments. Each step in a problem involves both entering a measure value (e.g., 50 degrees) and the name of a "reason", a geometric definition or theorem (e.g., Triangle Sum) that explains why the entered value must be what it is. Students enter a reason name either by typing or by selecting it from an on-line glossary.

As a first step toward tutorial dialog around student explanations, Aleven and Koedinger (2000) performed a classroom experiment with a version of the tutor that required students to express reasons in full and in their own words. This tutor was not capable of analyzing these free form explanations. Students had substantial difficulty providing free form explanations without tutor support. In the face of this difficulty and without an immediate check on the correctness of their explanations, students frequently (more than half the time) gave up or entered off-task statements. Thus, explanation analysis appears needed.

This paper reports on our progress toward a natural language understanding system that can analyze student explanations. In particular, the system must distinguish among a variety of explanation attempts including those that: are correct and complete, express the right idea but incompletely or imprecisely, are partially correct but include incorrect information, are stated correctly but are inappropriate for the current step, and are totally wrong. Because correct mathematical reasons are expressed in general and unambiguous terms, the system cannot employ knowledge of the context of the student's explanation in analyzing its correctness. Although context knowledge is available, it is pedagogically important that students state reasons unambiguously as such precision is a key instructional goal. Thus our system must detect correct reasons without making use of context information.

In many natural language understanding domains, it is sufficient to understand the intention of the speaker or writer, for instance, in processing information requests. However, in tutoring domains it is important to not only assess whether students appear to have the right idea, but also whether they can articulate this idea in full. In other words, does the student understand the crucial components of the idea that delineate when and how it applies and when it does not? Unlike information retrieval, we argue that natural language understanding of this kind cannot be performed with statistical methods alone, but requires a knowledge-based understanding system.

In the following section we describe a corpus of student explanations we have collected that has guided our initial development. We describe the challenges posed by real student explanations and overview the strategies we have implemented to address these challenges. Next, we discuss why knowledge-based understanding is particularly appropriate in this domain. We then describe how the system works in more detail. Finally, we present a preliminary evaluation of the system's performance relative to a human grader and discuss limitations and next steps.

Features of the Corpus

As a basis for the development of the system, we collected a corpus of students' explanations from written pre- and post-tests. Those tests were given to students in classes that also use the PACT Geometry Tutor as part of the course. Students were given a "Reason Sheet" which provided names and a sentence description of 19 geometry definitions or theorems. Students were asked to provide an explanation for each step in each problem. The different test forms had about 21 steps on them, so we collected about 21 explanations per student. The data comes from 100 students who took either or both the pre-test and the post-test. Our corpus included a total of 2333 explanations. Table 1 shows these explanations categorized into 6 broad categories. These categories are:

- **Blank**: No explanation was provided.
- **Procedural Replay**: The explanation only restated the operations performed in computing the value.
- **Reason**: The explanation consisted of a more or less complete sentence stating the actual theorem.
- **Reference**: The explanation named one of the theorems in the reason sheet.
- **Procedural Replay & Reason**: The explanation included a combination of a procedural replay and a reason.
- **Procedural Replay & Reference**: The explanation included a combination of a procedural replay and a reference.

The **%Correct** column in Table 1 shows what percentage of the explanations in the respective category were judged correct by a human grader. This judgement includes both an evaluation of how good the sentence was as an expression of the intended reason, and of how appropriate the intended reason was as an explanation for the particular value it was associated with. Table 2 shows examples of student explanations.

Table 1. Categories of student explanations, their frequency, and percent correct.

	Count	Frequency	%Correct
Blank	255	10.9%	0%
Procedural Replay	117	5.0%	0%
Reason	676	30%	23%
Reference	1196	51%	56%
Proc. Replay & Reason	60	2.6%	70%
Proc. Replay & Ref.	29	1.2%	79%
Grand Total	2333		45%

Table 2. Examples of student explanations of steps where "`Linear pair`" is the correct reason.

	Correct	Incorrect
Procedural Replay	`[none]`	`I took 107 and subtracted it from 180`
Procedural Replay & Reason	`180-118 because linears add up to 180`	`180-118=62 for straight line`
Procedural Replay & Reference	`linear pair - 180-107`	`[none]`
Reason	`the sum of the measures of a linear pair of angles is 180 degrees.`	`linear pairs = 180`
Reference	`because it's a linear pair.`	`isosceles triangle`

Our NLU development has so far focused on understanding student attempts at "Reasons", thus we have been investigating the 676 instances of Reason attempts in the corpus.

Challenges and How They Were Addressed

The corpus presents numerous problems for building a semantic representation that models the logical meaning of sentences. Key challenges are described below. We also provide an overview of the solutions we are developing. In the section following, we provide more detail on how our natural language understanding system works.

Number agreement:

Example: `The angles is congruent.`
Students often mismatch number markings of nouns, pronouns, and verbs. While relaxing the grammar to allow for such inconsistencies is not a problem, the semantic representation is another issue. The representation of a singular noun should be as a single item of the appropriate class, while that of a plural noun should be a set of such items. Since the system does not know which one is wrong, it cannot consistently rely on number information in either the noun or the verb for this choice. The current solution gives priority to information in the noun but at the same time uses a "lazy" strategy in building the sets corresponding to plural nouns, instantiating them only when needed based on semantic constrains of related concepts.

Plurals:

Example: `A pair of vertical angles formed by intersecting lines are congruent.`
Even if the subject is singular, the verb is in plural form. However the sentence has a valid meaning, because the subject represents a set of objects, while the predicate expresses a relation among these objects that are the elements of the set. Then, in case of subjects that represent sets, the system chooses, based on the semantic features of the properties involved ("`congruent`"), whether to assert the property on the set itself or on its elements.

Distributive versus collective reading:

Example: `The angles in a triangle are 180 degrees.`
Is this property asserted about each of the angles in the set (e.g., if 180 were 60) or about the sum of their measures? That is, is this a distributive property or a collective property? One solution is to both build the distributive reading whenever semantic constraints allow the property to be asserted on the elements of referenced set and build the collective reading whenever the set of elements forms a structure that has the asserted property well defined for it. Further developments could involve making a choice based on plausible ranges of values. Ambiguities that remain are ideal candidates for tutorial dialog: "is every angle in a triangle equal to 180 degrees?".

Metonymy:

Example: `The sum of a linear pair of angles is 180.`
Metonymy is the phenomenon of imprecise reference whereby a technically incorrect concept is used as a shortcut for the correct concept (e.g., "New York called" to indicate the guy from New York called). In the example above, it is technically incorrect to add angles, but "angles" is a shortcut for "the measures of the angles". The complete phrase is: "`sum of the measures of the angles in a linear pair`". The difficulty is not so much building a semantic structure where angles are added instead of measures, but rather in recognizing that this structure is logically equivalent to adding measures. The problem is composed here with the previous problem of collective vs. distributive reading, because, in order to get to angles, the system has to first realize the sum is not a relation over pairs of angles, but over the elements of this pair. The solution we have so far is to make the system recognize the failure of a direct semantic connection between sum and angles, and then try to build the right semantic representation by searching for the conceptual link between sum and angles in the semantic vicinity of the concepts involved. While the example is frequent enough that it could be dealt with as a special case, there are numerous other instances of metonymy in student explanation attempts.

Anaphora resolution:

Example: `If the lengths of two sides of a triangle are equal, then the measures of the angles opposite them will also be equal.`
In this example the pronoun "`them`" in the main clause is used as a reference to the set denoted by "`two sides`" in the conditional clause. Without solving this reference the system won't be able to build the right semantic representation for the sentence. There are however several different candidate references for binding "`them`" besides the right one: "`the lengths`", "`a triangle`", "`the measures`", and "`the angles`". Currently the system uses a combination of syntactic and semantic constraints to choose among possible candidates. However, if two or more candidates still pass those tests, it will just choose the one that is structurally closest to the pronoun. In this case, the system will eliminate "`the lengths`" and "`the measures`", based on a semantic constraint that geometry objects cannot oppose measures. And it will eliminate "`the angles`" based on the syntactic constraint that a pronoun has to be free in its local domain. While in theory "`a triangle`" could be ruled out based on number information, the unreliability of number markings mentioned above makes this more difficult.

Prepositional phrase attachment ambiguity:

Example: `The sum of the measures of the three interior angles in a triangle is equal to 180 degrees.`

While there might be syntactical preferences, there is no syntactic constraint to prevent the prepositional phrase "`in a triangle`" from being attached to either "`the sum`", "`the measures`", or "`the three interior angles`". However semantic constraints on the nature of concepts that can be related by the relation "`in`" can ensure the correct choice.

Why Is Knowledge-Based NLU Appropriate Here?

Statistical-based methods like Latent Semantic Analysis (e.g., Wiemer-Hastings, Wiemer-Hastings, and Graesser 1999), trigrams, Hidden Markov Models, etc. have been used for NLU in various applications with various degrees of success. We believe that a knowledge-based approach has better chances of success in the context of the current project for several reasons relating to characteristics of the task at hand:

- The domain of mathematical discourse has a limited number of precisely defined concepts and relations among them. This makes one of the main problems with knowledge-based approaches, the task of modeling the domain of discourse in a knowledge base, much easier than in generic vague domains of other applications.
- Students' sentences tend to be rather short, making many statistical approaches less likely to work well because of lack of enough information.
- Statistical approaches tend to work fine on the task of classifying texts as belonging to one of several given classes. Small changes in the text usually won't change the classification. In analyzing students' explanations, subtle language distinctions might lead to significant differences in meaning, making statistical approaches less likely to succeed in classifying them accordingly.
- Methods that don't take word order into consideration, like Latent Semantic Analysis and many text classification algorithms (e.g., Brüninghaus and Ashley 1997), will fail to classify properly a rule and its converse, which are the same bag of words. For instance, consider the rule "`If two angles form a line, then they sum to 180`" and its converse "`If two angles sum to 180, then they form a line`". The same words are used in both, but the first is true while the second is false. As a second example, consider the rule "`If two sides of a triangle are congruent, then the angles opposite them are congruent`" and its converse "`If two angles of a triangle are congruent, then the sides opposite them are congruent`". In this case, both rules are true, but in a given situation only one will be a correct reason.
- Being able to carry out an intelligent dialog with students about what is wrong and what is right in the explanations they give necessitates having a precise representation of the meaning of those explanations, so that their elements can be analyzed in detail. Statistical approaches tend not to be able to provide such representations.

How Does the System Work?

In the current implementation we approach the understanding of geometry theorems through having them classified by a description logic system. Thus we aim at building a unique semantic representation for all sentences with the same meaning content. This representation should carry all logical conditions so that it can be recognized by a logic system as being a valid representation for a given theorem.

In order to do that, the system uses a parser that takes a sentence and a grammar and builds a feature structure representing the syntactic structure of the sentence. At the same time it issues directives to the logic-based knowledge representation system to piece together a semantic representation of the sentence. This representation is then evaluated by the classifier against definitions of geometry theorems. The next paragraphs describe the key system components and their function.

Parser

The parser used currently is LCFlex, a left-corner active-chart parser developed at University of Pittsburgh (Rosé and Lavie 1999). The parser uses a context free formalism augmented with feature structures. The formalism originates in a previous GLR* system at Carnegie Mellon University (Lavie 1995) (Tomita 1988). The parser takes words of the input sentence one by one, and after a lexicon lookup, it identifies those rules of the grammar that could model the current word sequence. It keeps track of the active rules at any point in time by storing them in a chart that shows all possible partial parses up to that point.

Grammar

The grammar consists of context free rules accompanied by equations. The equations specify restrictions on the feature structures corresponding to the non-terminals of the rule. The current grammar used in the system follows loosely the Lexical Functional Grammar theory (Bresnan 1997).

An example of a simplified rule that builds a clause from a sequence of a noun phrase and a verb phrase is:

```
(<Cl> <== (<NP> <VP>)
((x0 = x2)
 ((x0 subject) = x1)
  (*test* (connect-instances
           (x0 sem content)
           (x0 subject sem-role)
           (x0 subject sem content)))))
```

Grammar rules also include calls that direct the logic system to build the semantic representation. The rule above shows such a call (`connect-instances`) that will connect the semantic representations of the elements through a binary relation specified by the subject's semantic role for the current predicate.

Knowledge Representation

The system uses Loom (MacGregor 1991) (Brill 1993) both to express knowledge about the domain of discourse and to build semantic representations of natural language sentences. Knowledge about the domain of discourse is represented as Loom definitions and productions. Those serve the purpose of axioms in a logic language.

The semantic representation of natural language sentences is expressed in terms of Loom assertions over Loom instances. Loom instances are used to represent discourse referents and play the role of variables in first order logic. Assertions correspond to predicates.

Loom provides deductive support for declarative knowledge expressed as definitions through a description classifier (MacGregor 1991). The classifier utilizes forward-chaining, semantic unification and object-oriented truth maintenance technologies to dynamically infer conceptual affiliation of instances based on given definitions.

As an illustration, a simplified version of the concept definitions necessary for building a semantic representation of the statement:

`The measure of a right angle is 90 degrees.`

is:

```
(defconcept Measure
  :is-primitive Abstraction)

(defconcept Measure-Unit
  :is (:one-of 'degree 'rad
               'meter 'centimeter))

(defconcept Geometry-Measure
  :is (:and Measure
            (:the value Number)
            (:the unit Measure-Unit))

(defconcept Geometry-Object
  :is-primitive
  (:and Abstraction
      (:all Measure Geometry-Measure)))

(defproperty Right
  :is-primitive Quality
  :domain Geometry-Object)

(defconcept Angle-Measure
  :is (:and Geometry-Measure
           (:the unit
               (:one-of 'degree 'rad))))

(defconcept Angle
  :is-primitive
   (:and Geometry-Object
         (:the measure Angle-Measure)))

(defconcept Being&Having
  :is-primitive Configuration)

(defconcept Ascription
  :is (:and Being&Having
            (:exactly 1 attribuend)
            (:the attribute Measure))

(defproduction ascription
  :when (:detects (Ascription ?x))
  :do ((combine-instances
         (get-value ?x 'attribute)
         (get-value ?x 'attribuend))))
```

Based on these definitions, the parser creates the following Loom instances to stand for the discourse referents introduced by the subject and the verb phrase:

```
(tell (:about measure-1
       (:create Measure)
       (belongs-to angle-2)
       (measure-of angle-2)))

(tell (:about angle-2
       (:create Angle)
       right
       (measure measure-1)))

(tell (:about measure-4
       (:create Measure)
       (unit ‘degree)
       (value 90)))

(tell (:about being&having-3
       (:create Being&Having)
       (attribute measure-4)))
```

Then, when applying the grammar rule for clauses given above, and considering information about verb arguments found in the lexical entry for the verb "`be`", the parser issues the command:

```
(connect-instances ‘being&having-3
         ‘attribuend ‘measure-1)
```

The "`connect-instances`" function checks whether it makes semantic sense for the attribuend "`the measure of an angle`" to have the attribute "`90 degrees`". In this case it does. (For a sentence like "`the measure of an angle is blue`", Loom would find a semantic inconsistency between the arguments and will fail the analysis). Loom then modifies the instance for "`be`" above to:

```
(tell (:about being&having-3
       (:create Being&Having)
       (attribute measure-4)
       (attribuend measure-1))
```

Then the Loom classifier recognizes this instance as having all necessary elements to belong to concept "`Ascription`", and fires the production rule associated with it. This production rule then issues the command:

```
(combine-instances 'measure-4
                   'measure-1)
```

The effect is to actually combine the two instances into a single one, and thus create the structure that models the assertion in the sentence:

```
(tell (:about measure-1
       (:create Measure)
       (belongs-to angle-2)
       (measure-of angle-2)))
       (unit 'degree)
       (value 90)))
(tell (:about angle-2
       (:create Angle)
       right
       (measure measure-1)))
(tell (:about being&having-3
       (:create Being&Having)
       (attribute measure-1)
       (attribuend measure-1))
```

If given a concept definition for what constitutes a valid statement for the right angle theorem, as below, Loom will then classify the instance "`being&having-3`" above as belonging to this concept.

```
(defconcept Right-Angle-Reason
  :is (:and Ascription
         (:the attribuend
            (:and Measure
               (:the measure-of
                  (:and Angle Right)))
         (:the attribute
               (:the value 90))))
```

Progress So Far and Limitations

As a preliminary evaluation of the current status of the system, we have tested it on the 676 statements in the corpus on the task of simply accepting or rejecting the students' statements. Of these, the system classifies successfully 402 statements in the sense that its classification matches the classification of a human grader, or the reason was incorrect and rejected. Of these, 143 are a correct reason and 259 are incorrect. The incorrect reasons were classified so either because the student stated an incorrect reason or the student's attempt at the correct reason was incorrect or incomplete. The system misclassifies only 7 reasons that were considered correct by the human grader. 267 other statements are not classified. Of these, most (264) are incorrect, and very few (3) are correct. Table 3 summarizes the results of the comparison of the system with the human grader.

Table 3. Comparison of the system with a human grader

	Grader's Judgment	
System's Judgment	**Reason Correct**	**Reason Incorrect**
Reason Correct	143 (21%)	0 (0%)
Reason Incorrect	7 (1%)	259 (38%)
No Class	3 (0.5%)	264 (39%)

However the threshold for correctness in this test was set very high. Many of the incorrect statements are not necessarily totally incorrect, but rather incomplete or imprecisely stated. A strict teacher might say that these are indeed incorrect, but this same teacher would be able to recognize that these sentences are close to expressing a valid reason. While teachers may not totally accept such reasons, they would not totally reject them, but rather engage in a dialog to help the student refine them. Examples of these imprecise reasons are:

```
All angles in a triangle = 180.
The sum of the measures of the 3 int.
of a tri. are 180.
The sum of the triangle has to equal
180.
2 and 4 are same angles, linears add up
to 180.
Because a line has to equal 180.
```

Some of these sentences introduce new challenges for the process of building a semantic representation. The main problem is that in many of these cases the parse fails because of either syntactic constraints in the grammar (e.g., the adjective "linear" is used as a noun), or semantic constraints in the knowledge base (e.g., a line is given an angle measure). Solving such problems will require finding ways to relax those constraints when needed, but still enforce them in most cases.

What we want to achieve is to be able to distinguish solid-hits from near-hits from near-misses from total misses. And further to be able to change the correctness criteria by moving a partial reason either in or out of the list of acceptable reasons for a step.

We currently use the Loom classifier to tell us when the semantic representation of a natural language sentence constitutes a valid expression of a geometry theorem. In order to be able to also recognize when a given sentence is a near-hit under this approach, we have to build descriptive concepts of all the partial reasons we want to recognize. Currently we have already defined such a set of partial reasons. However it may not be reasonable to try to predict all different ways a student might miss a complete reason. A more general approach would be to develop a comparison algorithm. It would take the semantic representation of a student's sentence and a concept description of the target geometry theorem and would return differences between the two, in terms of missing and/or wrong assertions.

Conclusion and Next Steps

Previous versions of our PACT Geometry Tutor were able to recognize students' reasons only as names of or references to a set of predefined reasons. This paper describes a natural language system that is capable of understanding students' reasons expressed in free-form natural language. Recently we created a prototype version of the tutor that integrates this natural language understanding system and provides simple feedback on students' free-form explanations. We see this as a first step towards building tutors with full dialog capabilities. At this point our natural language understanding system demonstrated reasonable performance on an early test of accepting/rejecting reasons. However much remains to be done in order to be able to correctly recognize partially correct statements of the same reasons.

Among the immediate developments we plan to work on are:

- Continue to improve the coverage of the system's grammar and semantics.
- Collect a corpus of explanations generated in the context of using the tutor in normal class sessions.
- Create more partial concept categories.
- Develop tutorial dialog moves to respond to different classification states (solid-hit, near-hit, near-miss, total miss).
- Develop an algorithm to give us information about the differences between partial reasons and correct reasons.

References

Aleven, V. and Koedinger, K. R. 2000. Toward a tutorial dialog system that helps students to explain solution steps. In *Proceedings of the AAAI 2000 Fall Symposium on Building Dialogue Systems for Tutorial Applications*, Menlo Park, CA: AAAI Press.

Aleven, V.; Koedinger, K. R.; and Cross, K. 1999. Tutoring answer-explanation fosters learning with understanding. In Lajoie, S. P. and Vivet, M. eds. *Artificial Intelligence in Education, Open Learning Environments: New Computational Technologies to Support Learning, Exploration, and Collaboration, Proceedings of AIED-99*, 199-206. Amsterdam, Netherlands: IOS Press.

Anderson, J. R.; Corbett, A. T.; Koedinger, K. R.; and Pelletier, R. 1995. Cognitive tutors: Lessons learned. In *The Journal of the Learning Sciences*, 4:167-207.

Anderson, J. R. and Lebiere, C. 1998. *The Atomic Components of Thought.* Hillsdale, NJ: Erlbaum.

Bresnan, J. 1997. Lexical-functional syntax. ms, Stanford University.

Brill, D. 1993. Loom reference manual version 2.0. University of Southern California.

Brünninghaus, S. and Ashley, K. D. 1997 Finding factors learning to classify case opinions under abstract fact categories. In Brantig L. K. ed. *Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, Melbourne, Australia.

Glass, M. 1997. Some phenomena handled by the Circsim-tutor version 3 input understander. In Proceedings of the Tenth International Florida Artificial Intelligence Research Symposium, Daytona Beach, FL.

Lavie, A. 1995. A grammar based robust parser for spontaneous speech. Ph.D. diss., Carnegie Mellon University.

Koedinger, K. R. and Anderson, J. R. 1993. Reifying implicit planning in geometry. In S. Lajoie and S. Derry eds. *Computers as Cognitive Tools*, 15-45. Hillsdale, NJ: Erlbaum.

Koedinger, K. R.; Anderson, J. R.; Hadley, W. H.; and Mark, M. A. 1997. Intelligent tutoring goes to school in the big city. In *International Journal of Artificial Intelligence in Education*, 8:30-43.

MacGregor R. 1991. Using a description classifier to enhance deductive inference. In Proceedings of the Seventh IEEE Conference on AI Applications, 141-147. Miami, FL.

NCTM 1989. Curriculum and Evaluation Standards for School Mathematics. National Council of Teachers of Mathematics. Reston, VA: The Council.

Rosé, C. P. and Lavie, A. 1999. LCFlex: An efficient robust left-corner parser, User's manual, University of Pittsburgh.

Tomita, M. 1988. The generalized LR parser/compiler. User's guide, Carnegie Mellon University.

Wiemer-Hastings, P.; Wiemer-Hastings, K.; and Graesser, A. C. 1999. Improving an intelligent tutor's comprehension of students with latent semantic analysis. In Lajoie, S. P. and Vivet, M. eds. *Artificial Intelligence in Education, Open Learning Environments: New Computational Technologies to Support Learning, Exploration, and Collaboration, Proceedings of AIED-99*, 535-542. Amsterdam, Netherlands: IOS Press.