Modeling Students' Reasoning about Qualitative Physics: Heuristics for Abductive Proof Search

Maxim Makatchev, Pamela W. Jordan, and Kurt VanLehn

Learning Research and Development Center, University of Pittsburgh
{maxim,pjordan,vanlehn}@pitt.edu

Abstract. We describe a theorem prover that is used in the Why2-Atlas tutoring system for the purposes of evaluating the correctness of a student's essay and for guiding feedback to the student. The weighted abduction framework of the prover is augmented with various heuristics to assist in searching for a proof that maximizes measures of utility and plausibility. We focus on two new heuristics we added to the theorem prover: (a) a specificity-based cost for assuming an atom, and (b) a rule choice preference that is based on the similarity between the graph of cross-references between the propositions in a candidate rule and the graph of cross-references between the set of goals. The two heuristics are relevant to any abduction framework and knowledge representation that allow for a metric of specificity for a proposition and cross-referencing of propositions via shared variables.

1 Introduction

1.1 Why2-Atlas overview

The Why2-Atlas tutoring system is designed to encourage students to write their answers to qualitative physics problems along with detailed explanations to support their arguments [1]. For the purpose of eliciting more complete explanations the system attempts to provide students with substantive feedback that demonstrates understanding of a student's essay. A sample problem and a student's explanation for it is shown in Figure 1.

The sentence level understanding module in Why2-Atlas parses a student's essay into a first-order predicate representation [2]. The discourse-level understanding module then resolves temporal and nominal anaphora within the representation [3] and uses a theorem prover that attempts to generate a proof, treating propositions in the resolved representation as a set of goals, and the problem statement as a set of given facts. An informal example proof for a fragment of the essay in Figure 1 is shown in Figure 2. The proof is interpreted as a model of the reasoning the student used to arrive at the arguments in the essay, and provides a diagnosis when the arguments are faulty in a fashion similar to [4, 5]. For example, the proof in Figure 2 indicates that the student may have

Question: Suppose a man is in a free-falling elevator and is holding his keys motionless right in front of his face. He then lets go. What will be the position of the keys relative to the man's face as time passes? Explain.

Explanation: The keys are affected by gravity which pulls them to the elevator floor, because the keys then have a combined velocity of the freefall and the effect of gravity. If the elevator has enough speed the keys along with my head would be pressed against the ceiling of the elevator, because the acceleration of the elevator car along with me and the keys would overwhelm the gravitational pull.

Fig. 1. The statement of the problem and an example explanation.

Step #	Proposition	Justification
1	before the release, the keys have been in contact with the man, and the man has been in contact with the elevator	given
2	at the moment of release, velocity of the keys is equal to velocity of the elevator	bodies in contact over a time interval have same velocities
3	after the release, nothing is touching the keys	given
4	after the release, the keys are in freefall	if there is no any contact then the body is in freefall
5	after the release, the keys' acceleration is not equal to the elevator's acceleration	*elevator is not in freefall
6	after the release, the keys' velocity is not equal to the elevator's velocity	if initial velocity is the same and accelerations are different the final velocities are different
7	the keys touch the ceiling of the elevator	if the keys' velocity is smaller than the elevator's velocity, the keys touch the ceiling

Fig. 2. An informal proof of the excerpt "The keys would be pressed against the ceiling of the elevator" (From the essay in Figure 1). The buggy assumption is preceded by an asterisk.

wrongly assumed that the elevator is not in freefall. A highly plausible wrong assumption in the student's reasoning triggers an appropriate tutoring action [6].

The theorem prover, called Tacitus-lite+, is a derivative of SRI's Tacitus-lite that, among other extensions, incorporates sorts (sorts will be described in Section 2.3) [7, p. 102]. We further adapted Tacitus-lite+ to our application by (a) adding meta-level consistency checking, (b) enforcing a sound order-sorted inference procedure, and (c) expanding the proof search heuristics. In the rest of the paper we will refer to the prover as Tacitus-lite when talking about features present in the original SRI release, and as Tacitus-lite+ when talking about more recent extensions.

The goal of the proof search heuristics is to maximize (a) the measure of plausibility of the proof as a model of a student's reasoning and (b) the measure of utility of the proof for generating tutoring feedback. The measure of plausibility can be evaluated with respect to the misconceptions that were identified as present in the essay by the prover and by a human expert. A more precise plausibility measure may take into account plausibility of the proof as a whole. The measure of utility for the tutoring task can be interpreted in terms of relevance of the tutoring actions (triggered by the proof) to the student's essay, whether the proof was plausible or not.

A previous version of Tacitus-lite+ was evaluated as part of the Why2-Atlas evaluation studies, as well as on its own. The stand-alone evaluation uses manually constructed propositional representation of essays, to measure the performance of the theorem prover (in terms of the recognition of misconceptions in the essay) on 'gold' input [8]. The results of the latter evaluation were encouraging enough for us to continue development of the theorem proving approach for essay analysis.

1.2 Related work

In our earlier paper [9] we argued that statistical text classification approaches that treat text as an unordered bag of words (e.g. [10, 11]) do not provide a sufficiently deep understanding of the logical structure of the student's essay that is essential for our application. Structured models of conceptual knowledge, including those based on semantic networks and expert systems, are described in [12]. Another structured model, Bayesian belief networks, is a popular tool for learning and representing student models [13, 14]. By appropriately choosing the costs of propositions in rules, weighted abductive proofs can be interpreted as Bayesian belief networks [15, 4]. In general, the costs of propositions in abductive theorem proving do not have to adhere to probabilistic semantics, providing greater flexibility while also eliminating the need to create a proper probability space. On the other hand, the task of choosing a suitable cost semantics in weighted abduction remains a difficult problem and it is out of scope of this paper.

Theorem provers have been used in tutoring systems for various purposes, e.g. for building the solution space of a problem [16] and for question answering [17], to mention a few. Student modeling from the point of view of formal methods is reviewed in [18]. An interactive construction of a learner model that uses a theorem proving component is described in [19].

In this paper we focus on the recent additions to the set of proof search heuristics for Tacitus-lite+: a specificity-sensitive assumption cost and a rule choice preference that is based on the similarity between the graph of cross-references between the propositions in a candidate rule and the graph of cross-references between the set of goals. The paper is organized as follows: Section 2 introduces knowledge representation aspects of the prover; Section 3 defines the order-sorted abductive inference framework and describes the new proof search heuristics; finally, a summary is given in Section 4.

2 Knowledge representation for qualitative mechanics

In addition to the domain knowledge that is normally represented in qualitative physics frameworks (e. g. [20]), a natural language tutoring application requires a representation of possibly erroneous student beliefs that captures the differences between beliefs expressed formally and informally, as allowed by natural language. The process of building a formal representation of the problem can be described in terms of envisionment and idealization.

2.1 Envisionment and idealization

The internal (mental) representation of the problem plays a key role in problem solving among both novices and experts [21, 22]. The notion of an internal representation, described in [22] as "objects, operators, and constraints, as well as initial and final states," overlaps with the notion of *envisionment* [23], i.e. the sequence of events implied by the problem statement. While envisionment can be expressed as a sequence of events in common-sense terms, a further step towards representing the envisionment in formal physics terms (bodies, forces, motion) is referred to as *idealization* [8].

For example, consider the problem in Figure 1. A possible envisionment is: (1) the man is holding the keys (elevator is falling); (2) the man releases the keys; (3) the keys move up with respect to the elevator and hit the elevator ceiling.

The idealization would be:

Bodies: Keys, Man, Elevator, Earth.
Forces: Gravity, Man holding keys
Motion: Keys' downward velocity is smaller than the downward velocity of the elevator.

Because envisionment and idealization are important stages for constructing an internal representation, they fall under the scope of Why2-Atlas' tutoring. However, reasoning about the multitude of possible envisionments would require adding an extensive amount of common-sense knowledge to the system. To bypass this difficulty, we consider problems that would typically have few likely envisionments. Fortunately (for the knowledge engineers), there is a class of interesting qualitative physics problems that falls into this category. We therefore developed a knowledge representation that is capable of representing common correct and erroneous propositions at both the levels of envisionment and idealization.

2.2 Qualitative mechanics ontology

The ontology is designed to take advantage of the additional capability provided by an order-sorted language (described in Section 2.3). Namely, constants and variables, corresponding to physical quantities (e. g. force, velocity), physical bodies (man, earth) and agents (air) are associated with a sort symbol. The domains of the predicate symbols are restricted to certain sorts (so that each argument position has a corresponding sort symbol). These associations and constraints constitute an *order-sorted signature* [24].

The ontology consists of the following main concept classes: bodies, physical quantities, states, time, relations, as well as their respective slot-filler concepts. For details of the ontology we refer the reader to [8].

```
((acceleration a1 keys vertical ?d-mag
                 ?d-mag-num nonzero ?mag-num neg ?dir-num ?d-dir ?t1 ?t2)
(Quantity1b Id Regular-body Axial D-mag
                 D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time))
((due-to d1 a1 ph1)  (Due-to Id Id Id))
((phenomenon ph1 gravity)  (Phenomenon Id Field-interaction))
```

Fig. 3. Representation for "The keys have a downward acceleration due to gravity." The atoms are paired with their sorted signatures.

2.3 Order-sorted first-order predicate language

We adopted first-order predicate logic with sorts [24] as the representation language. Essentially, it is a first-order predicate language that is augmented with an *order-sorted signature* for its terms and predicate argument places. For the sake of computational efficiency and since function-free clauses are the natural output of the sentence-level understanding module (see Section 1), we do not implement functions, instead we use cross-referencing between atoms by means of shared variables. There is a single predicate symbol M_i for each i-place relation. For this reason predicate symbols are omitted in the actual representation. Each atom is indexed with a unique *identifier*, a constant of sort `Id`. The identifiers, as well as variable names, can be used for cross-referencing between atoms. For example, the proposition "The keys have a downward acceleration due to gravity" is represented as shown in Figure 3, where `a1`, `d1`, and `ph1` are atom identifiers. For this example we assume (a) a fixed coordinate system, with a vertical axis pointing up (thus `Dir` value is `neg`); (b) that the existence of an acceleration is equivalent to existence of a nonzero acceleration (thus `Mag-zero` value is `nonzero`).

2.4 Rules

As we mentioned in Section 2.1, it is important to have rules about both envisionment and idealization when modeling students' reasoning. The idealization of the canonical envisionment is represented as a set of givens for the theorem prover, namely rules of the form $\rightarrow a$. A student's reasoning may contain false facts, including an erroneous idealization and envisionment, and erroneous inferences. The former are represented via *buggy givens* and the latter are represented via *buggy rules*. Buggy rules normally have their respective correct counterparts in the rule base. Certain integrity constraints apply when a student model is generated, based on the assumption that the student is unlikely to use correct and buggy versions of a rule (or given) within the same argument.

An example of a correct rule, stating that "if the velocity of a body is zero over a time interval then its initial position is equal to its final position", is shown in Figure 4. Note that the rules are *extended Horn clauses*, namely the head of the rule is an atom or a conjunction of multiple atoms.

```
((velocity v1 ?body ?comp ?d-mag
                    ?d-mag-num zero ?mag-num ?dir ?dir-num ?d-dir ?t1 ?t2)
(Quantity1b Id Body Comp D-mag
                  D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time))
→
((position p1 ?body ?comp ?d-mag1
         ?d-mag-num1 ?mag-zero1 ?mag-num1 ?dir1 ?dir-num1 ?d-dir1 ?t1 ?t1)
(Quantity1b Id Body Comp D-mag
                  D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time))
((position p2 ?body ?comp ?d-mag1
         ?d-mag-num1 ?mag-zero1 ?mag-num1 ?dir1 ?dir-num1 ?d-dir1 ?t2 ?t2)
(Quantity1b Id Body Comp D-mag
                  D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time))
```

Fig. 4. Representation for the rule "If the velocity of a body is zero over a time interval then its initial position is equal to its final position."

3 Abductive reasoning

3.1 Order-sorted abductive logic programming

Similar to [25] we define the *abductive logic programming framework* as a triple $\langle T, A, I \rangle$, where T is the set of *givens* and *rules*, A is the set of abducible atoms (potential hypotheses) and I is a set of integrity constraints. Then an *abductive explanation* of a given set of sentences G (observations) consists of (a) subset Δ of abducibles A such that $T \cup \Delta \vdash G$ and $T \cup \Delta$ satisfies I together with (b) the corresponding *proof* of G. Since an abductive explanation is generally not unique, various criteria can be considered for choosing the most suitable explanation (see Section 3.2).

An *order-sorted abductive logic programming framework* $\langle T', A', I' \rangle$ is an abductive logic programming framework with all atoms augmented with the sorts of their argument terms (so that they are sorted atoms) [8]. Assume the following notation: a *sorted atom* is of the form $p(x_1, \ldots, x_n) : (\tau_1, \ldots, \tau_n)$, where the term x_i is of the sort τ_i. Then, in terms of unsorted predicate logic, formula $\exists x\ p(x) : (\tau)$ can be written as $\exists x\ p(x) \wedge \tau(x)$. For our domain we restrict the sort hierarchy to a tree structure that is naturally imposed by set semantics and that has the property $\exists x\ \tau_i(x) \wedge \tau_j(x) \rightarrow (\tau_i \preccurlyeq \tau_j) \vee (\tau_j \preccurlyeq \tau_i)$, where $\tau_i \preccurlyeq \tau_j$ is equivalent to $\forall x\ \tau_i(x) \rightarrow \tau_j(x)$.

Tacitus-lite+ does backward chaining using the order-sorted version of modus ponens:

$$\frac{\begin{array}{l} q(x', z') : (\tau_5, \tau_6) \\ p(x, y) : (\tau_1, \tau_2) \leftarrow q(x, z) : (\tau_3, \tau_4) \\ \tau_5 \preccurlyeq \tau_3,\ \tau_6 \preccurlyeq \tau_4 \end{array}}{p(x', y') : (\min(\tau_5, \tau_1), \tau_2)} \qquad (1)$$

3.2 Proof search heuristics

In building a model of the student's reasoning, our goal is to simultaneously increase a function of measures of utility and plausibility. The utility measure is an estimate of the utility of the choice of a particular proof for the tutoring application given a plausibility distribution on a set of alternative proofs. The plausibility measure indicates which explanation is the most likely.

For example, even if a proof does not exactly coincide with the reasoning the student used to arrive at a particular conclusion that she stated in her essay, the proof may be of a high utility value, provided it correctly indicates the presence of certain misconceptions in the student's reasoning. However, generally plausible explanations have a high utility value and we deploy a number of heuristics to increase the plausibility of the proof.

Weighted abduction One of the characteristic properties of abduction is that atoms can be assumed as hypotheses, without proof. Normally it is required that the set of assumptions is minimal, in the sense that no proper subset of it is sufficient to explain the observation (or, in other words, to prove the goals). While this preference allows us to compare two explanations when one is a subset of another, *weighted abduction* provides a method to grade explanations so we can compare two arbitrary explanations.

Tacitus-lite extends the weighted abductive inference algorithm described in [26] for the case where rules are expressed as Horn clauses to the case where rules are expressed as extended Horn clauses, namely the head of a rule is an atom or a conjunction of atoms. Each conjunct p_i from the body of the rule has a weight w_i associated with it:

$$p_1^{w_1} \wedge \cdots \wedge p_m^{w_m} \rightarrow r_1 \wedge \cdots \wedge r_n$$

The weight is used to calculate the cost of abducing p_i, instead of proving it, via the formula $cost(p_i) = cost(g) \cdot w_i$, where g is the goal atom that has been proved via the rule at a preceding step (by unifying, say, with atom r_j). The costs of the observations are supplied with the observations as input to the prover.

Given a subgoal or observation atom to be proven, Tacitus-lite takes one of three actions; (a) assumes the atom at the cost associated with it; (b) unifies it with an atom that is either a fact or has already been proven or is another goal (in the latter case the cost of the resultant atom is counted once in the total cost of the proof, as the minimum of the two costs); (c) attempts to prove it with a rule. Tacitus-lite calls the action (b) *factoring*.

To account for the fact that in the order-sorted abductive framework a rule can generate new goals of various specificity (depending on the goals that were unified with the head of the rule), we adjust the weight of the assumed atom according to the sorts of its terms: a more general statement is less costly to assume, but a more specific statement is more costly. For example, the rule from Figure 4 can be applied to prove the goal "(Axial, or total) position of `?body3` has magnitude `?mag-num3`":

```
 ((position p3 ?body3 ?comp3 ?d-mag3
        ?d-mag-num3 ?mag-zero3 ?mag-num3 ?dir3 ?dir-num3 ?d-dir3 ?t3 ?t3)
(Quantity1b Id Body Axial D-mag
                D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time)),
```

which generates the subgoal "(Axial or total) velocity of `?body3` is zero":

```
((velocity v2 ?body3 ?comp3 ?d-mag2
              ?d-mag-num2 zero ?mag-num2 ?dir2 ?dir-num2 ?d-dir2 ?t3 ?t4)
(Quantity1b Id Body Axial D-mag
                 D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time))
```

The same rule can be applied to prove the more specific goal "Horizontal position of `?body3` has magnitude `?mag-num3`":

```
((position p4 ?body3 horizontal ?d-mag3
        ?d-mag-num3 ?mag-zero3 ?mag-num3 ?dir3 ?dir-num3 ?d-dir3 ?t3 ?t3)
(Quantity1b Id Body Axial D-mag
                D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time)),
```

and will generate the more specific subgoal "Horizontal velocity of `?body3` is zero":

```
((velocity v3 ?body3 horizontal ?d-mag2
              ?d-mag-num2 zero ?mag-num2 ?dir2 ?dir-num2 ?d-dir2 ?t3 ?t4)
(Quantity1b Id Body Axial D-mag
                 D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time))
```

Since the variables are assumed to be existentially quantified, in accordance with the sort semantics (see Section 3.1), the latter, more specific subgoal implies the former subgoal. Also, according to the ordered version of modus ponens (1), more rules can be used to prove the more general atom, increasing the chances for the atom to be proven, rather than assumed. These considerations suggest that *it should be less costly to assume more general atoms than more specific atoms*. The cost adjustment for the assumptions is implemented by computing a metric of specificity for the sorted signature of each assumed atom.

Rule choice heuristics Although the rules in Tacitus-lite are applied to prove individual goal atoms, a meaningful proposition usually consists of a few atoms cross-referenced via shared variables (see Section 2.3). When a rule is used to prove a particular goal atom, (a) a unifier is applied to the atoms in the head and the body of the rule; (b) atoms from the head of the rule are added to the list of proven atoms; and (c) atoms from the body of the rule are added to the list of goals. Consequently, suppose there exists a unifier θ that unifies both (a) a goal atom g_1 with an atom r_1 from the head of the rule $R: p_1 \wedge p_2 \rightarrow r_1 \wedge r_2$ so that g_1 can be proved with R via modus ponens, and (b) a goal atom g_2 with an atom r_2 from the head of the rule R so that g_2 can be proved via R.

Then, proving goal g_1 via R (and applying θ to g_1 and r_1) adds the atom $r_2\theta$ to the list of provens thus allowing for its potential factoring with goal g_2. In effect, a single application of a rule in which its head atoms match multiple goal atoms can result in proving multiple goal atoms via a number of subsequent factoring steps. This property of the prover is consistent (a) with backchaining using modus ponens (1), and (b) with the intuitive notion of cognitive economy, namely that the shortest (by the total number of rule applications) proofs are usually considered good by domain experts.

Moreover, if an atom $p_1\theta$ in the body of R can be unified with a goal $g_3\theta$, then the application of rule R will probably not result in an increase of the total cost of the goals due to the new goal $p_1\theta$, since it is possible to factor it with $g_3\theta$ and set the cost of the resultant atom as the minimum of the costs of $p_1\theta$ and $g_3\theta$. In other words, applying a rule where multiple atoms in its head and body match multiple goal atoms is likely to result in a faster reduction of the goal list, and therefore a shorter final proof.

The new version of Tacitus-lite+ extends the previous rule choice heuristics described in [9] with rule choice based on the best match between the set of atoms in a candidate rule and the set of goal atoms. To account for the structure of cross-references between the atoms, a labeled graph is constructed offline for every rule, so that the atoms are vertices labeled with respective sorted signatures and the cross-references are edges labeled with pairs of respective argument positions. Similarly a labeled graph is built on-the-fly for the current set of goal atoms. The rule choice procedure involves comparison of the goal graph and graphs of candidate rules so that the rule that maximizes the graph matching metric is preferred.

The match metric between two labeled graphs is based on the size of the largest common subgraph (LCSG). We have implemented the decision-tree-based LCSG algorithm proposed in [27]. The advantage of this algorithm is that the time complexity of its online stage is independent of the size of the rule graph: if n is the number of vertices in the goal graph, then the time complexity of the LCSG is $O(2^n n^3)$.

Since the graph matching includes independent subroutines for matching vertices (atoms with sorted signatures) and matching edges (cross-referenced atom arguments), the precision of both match subroutines can be varied to balance the trade-off between search precision and efficiency of the overall matching procedure. Currently we are evaluating the performance of the theorem prover under various settings.

4 Conclusion

We described an application of theorem proving for analyzing student's essays in the context of an interactive tutoring system. While formal methods have been applied to student modeling, there are a number of challenges to overcome: representing varying levels of formality in student language, the limited scope of the rule base, and limited resources for generating explanations and consistency checking. In our earlier paper [9] we argued that a weighted abduction theorem proving framework augmented with appropriate proof search heuristics provides a necessary deep-level understanding of a student's reasoning. In this paper we describe the recent additions to our proof search heuristics that have the goal of improving the plausibility of the proofs as models of students' reasoning as well as the computational efficiency of the proof search.

5 Acknowledgments

This work was funded by NSF grant 9720359 and ONR grant N00014-00-1-0600. We thank the entire Natural Language Tutoring group, in particular Michael Ringenberg and Roy Wilson for their work on Tacitus-lite+, and Uma Pappuswamy, Michael Böttner, and Brian 'Moses' Hall for their work on knowledge representation and rules.

References

1. VanLehn, K., Jordan, P., Rosé, C., Bhembe, D., Böttner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S., Srivastava, R.: The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In: Proceedings of Intelligent Tutoring Systems Conference. Volume 2363 of LNCS., Springer (2002) 158–167
2. Rosé, C., Roque, A., Bhembe, D., VanLehn, K.: An efficient incremental architecture for robust interpretation. In: Proceedings of Human Language Technology Conference, San Diego, CA. (2002)
3. Jordan, P., VanLehn, K.: Discourse processing for explanatory essays in tutorial applications. In: Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue. (2002)
4. Poole, D.: Probabilistic Horn abduction and Bayesian networks. Artificial Intelligence **64** (1993) 81–129
5. Young, R.M., O'Shea, T.: Errors in children's subtraction. Cognitive Science **5** (1981) 153–177
6. Jordan, P., Makatchev, M., Pappuswamy, U.: Extended explanations as student models for guiding tutorial dialogue. In: Proceedings of AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue. (2003) 65–70
7. Hobbs, J., Stickel, M., Martin, P., Edwards, D.: Interpretation as abduction. In: Proc. 26th Annual Meeting of the ACL, Association of Computational Linguistics. (1988) 95–103
8. Makatchev, M., Jordan, P.W., VanLehn, K.: Abductive theorem proving for analyzing student explanations to guide feedback in intelligent tutoring systems. To appear in Journal of Automated Reasoning, Special issue on Automated Reasoning and Theorem Proving in Education (2004)
9. Jordan, P., Makatchev, M., VanLehn, K.: Abductive theorem proving for analyzing student explanations. In: Proceedings of International Conference on Artificial Intelligence in Education, Sydney, Australia, IOS Press (2003) 73–80
10. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Processes **25** (1998) 259–284

11. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: Proceeding of AAAI/ICML-98 Workshop on Learning for Text Categorization, AAAI Press (1998)
12. Jonassen, D.: Using cognitive tools to represent problems. Journal of Research on Technology in Education **35** (2003) 362–381
13. Conati, C., Gertner, A., VanLehn, K.: Using bayesian networks to manage uncertainty in student modeling. Journal of User Modeling and User-Adapted Interaction **12** (2002) 371–417
14. Zapata-Rivera, J.D., Greer, J.: Student model accuracy using inspectable bayesian student models. In: International Conference of Artificial Intelligence in Education, Sydney, Australia (2003) 65–72
15. Charniak, E., Shimony, S.E.: Probabilistic semantics for cost based abduction. In: Proceedings of AAAI-90. (1990) 106–111
16. Matsuda, N., VanLehn, K.: GRAMY: A geometry theorem prover capable of construction. Journal of Automated Reasoning **32** (2004) 3–33
17. Murray, W.R., Pease, A., Sams, M.: Applying formal methods and representations in a natural language tutor to teach tactical reasoning. In: Proceedings of International Conference on Artificial Intelligence in Education, Sydney, Australia, IOS Press (2003) 349–356
18. Self, J.: Formal approaches to student modelling. In McCalla, G.I., Greer, J., eds.: Student Modelling: the key to individualized knowledge-based instruction. Springer, Berlin (1994) 295–352
19. Dimitrova, V.: STyLE-OLM: Interactive open learner modelling. Artificial Intelligence in Education **13** (2003) 35–78
20. Forbus, K., Carney, K., Harris, R., Sherin, B.: A qualitative modeling environment for middle-school students: A progress report. In: QR-01. (2001)
21. Ploetzner, R., Fehse, E., Kneser, C., Spada, H.: Learning to relate qualitative and quantitative problem representations in a model-based setting for collaborative problem solving. The Journal of the Learning Sciences **8** (1999) 177–214
22. Reimann, P., Chi, M.T.H.: Expertise in complex problem solving. In Gilhooly, K.J., ed.: Human and machine problem solving. Plenum Press, New York (1989) 161–192
23. de Kleer, J.: Multiple representations of knowledge in a mechanics problem-solver. In Weld, D.S., de Kleer, J., eds.: Readings in Qualitative Reasoning about Physical Systems. Morgan Kaufmann, San Mateo, California (1990) 40–45
24. Walther, C.: A many-sorted calculus based on resolution and paramodulation. Morgan Kaufmann, Los Altos, California (1987)
25. Kakas, A., Kowalski, R.A., Toni, F.: The role of abduction in logic programming. In Gabbay, D.M., Hogger, C.J., Robinson, J.A., eds.: Handbook of logic in Artificial Intelligence and Logic Programming. Volume 5. Oxford University Press (1998) 235–324
26. Stickel, M.: A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. Technical Report 451, SRI International, 333 Ravenswood Ave., Menlo Park, California (1988)
27. Shearer, K., Bunke, H., Venkatesh, S.: Video indexing and similarity retrieval by largest common subgraph detection using decision trees. Pattern Recognition **34** (2001) 1075–1091