Journal of Automated Reasoning **32:** 187–226, 2004.

Abductive Theorem Proving for Analyzing Student Explanations to Guide Feedback in Intelligent Tutoring Systems

MAXIM MAKATCHEV, PAMELA W. JORDAN, and KURT VANLEHN
Learning Research and Development Center, University of Pittsburgh, Pittsburgh, PA 15260, U.S.A.
e-mail: {maxim,pjordan,vanlehn}@pitt.edu

Abstract. The Why2-Atlas tutoring system presents students with qualitative physics questions and encourages them to explain their answers through natural language. Although there are inexpensive techniques for analyzing explanations, we claim that better understanding is necessary for use within tutoring systems. In this paper we motivate and describe how the system creates and uses a deeper proof-based representation of student essays in order to provide students with substantive feedback on their explanations. We describe in detail the abductive reasoner, Tacitus-lite+, that we use within the tutoring system. We also discuss evaluation results for an early version of the Why2-Atlas system and a subsequent evaluation of the theorem-proving module. We conclude with the discussion of work in progress and additional future work for deriving more benefits from a proof-based approach for tutoring applications.

Key words: intelligent tutoring systems, abductive reasoning, qualitative physics.

1. Introduction

Whereas most natural language explanations are produced and adapted to benefit or inform a hearer, a self-explanation is produced for the benefit of the speaker. If there is a hearer, he often already knows all about the topic, as is the case in a tutoring context. Self-explanation is a cognitively valuable pedagogical activity because it leads students to construct knowledge (Chi et al., 1994), and it can expose deep misconceptions (Slotta et al., 1995). But it is difficult to encourage self-explanation without giving the students substantive feedback on what they generate (Aleven and Koedinger, 2000; Chi et al., 2001). To give substantive feedback, the system has to be able to understand student explanations to some degree.

To study the problem of how to encourage students to productively self-explain, we built the Why2-Atlas intelligent tutoring system and selected qualitative physics as its domain of instruction. Qualitative physics is a worthy pedagogical goal because it is well known that college physics students are often unable to construct acceptable answers for even simple qualitative physics questions. Students with top grades in their physics classes get low scores on standardized measures of qualitative understanding, such as the Force Concepts Inventory (Hestenes et al., 1992).

Question: Suppose you are running in a straight line at constant speed. You throw a pumpkin straight up. Where will it land? Explain.

Explanation: Once the pumpkin leaves my hand, the horizontal force that I am exerting on it no longer exists, only a vertical force (caused by my throwing it). As it reaches it's maximum height, gravity (exerted vertically downward) will cause the pumpkin to fall. Since no horizontal force acted on the pumpkin from the time it left my hand, it will fall at the same place where it left my hands.

Figure 1. The statement of the problem and a verbatim explanation from a student who received no follow-up discussions on any problems.

Qualitative physics problems emphasize physics concepts more so than problem-solving tactics and applications of equations. Because little progress has been made at successfully improving qualitative understanding by altering physics classroom instruction and because training involving qualitative physics problems does help repair some persistent physics misconceptions (Hake, 1998), continued pursuit of qualitative understanding by tutoring systems is worthwhile.

The Why2-Atlas system covers seven qualitative problems on introductory mechanics that are adapted from (Hewitt, 1998). When the system presents one of these questions, it asks that the student type an answer and explanation, and it informs the student it will analyze and discuss the final response with the student. After the discussion, the system asks that the student revise the explanation, and the cycle of explanation, revision, and follow-up discussion continues until all the flaws in the student's response have been addressed. One such problem is shown in Figure 1. The student response shown is from our corpus of students' problem-solving sessions and illustrates how challenging these explanations are for a system to analyze for flaws that are worth discussing with a student. The dialogue shown in Figure 2 is a portion of a follow-up dialogue between a student and a human tutor for a flaw similar to that in Figure 1, while the dialogue shown in Figure 3 is a dialogue that the Why2-Atlas system held with a different student who is also exhibiting a similar flaw during an early evaluation of the system.

One difficulty in analyzing students' explanations for qualitative physics problems is that they require more than a single-sentence response and take the form of short essays. Two experienced physics instructors provided us with detailed explanations for 10 qualitative problems as well as minimal, but complete, essays that they would accept from students in an introductory physics course. The physics instructors arrived at their explanations and essays independently and then met and agreed on the minimum essays required. These reconciled, minimal essays contained 6.9 sentences on average. So the system must be capable of analyzing the relationships between sentences.

Statistical text classification approaches, such as latent semantic analysis (Landauer et al., 1998) and naive Bayes (McCallum and Nigam, 1998), have shown promise for assessing multisentential student explanations (Graesser et al., 2000; Rosé et al., 2002) and are particularly desirable because, for training, they require

T: After it is thrown what is the horizontal velocity of the pumpkin?

S: zero

T: If the velocity changes from whatever it was to zero, then there should be a force on it?

S: no

T: Does a change in velocity mean acceleration?

S: yes

T: What is required to produce an acceleration?

S: force

T: Right. So, if the pumpkins horizontal velocity was same as that of the runner and then it becomes zero when released then has there been a deceleration in it?

S: yes

T: Right. So there must be force which causes it?

S: yes

T: Ok.What is that force which reduces the horizontal velocity of the pumpkin to zero?

S: gravity

T: Let us see. What is the direction of the gravity?

S: vertical

T: Thats right. Can a vertical force cause a horizontal acceleration?

S: no

T: Right. So gravity cannot change the horizontal velocity of the pumpkin. can you see any horizontal force on the pumpkin?

S: no

T: Right. Then can its horizontal velocity component change after it is released?

S: no

Figure 2. A verbatim follow-up discussion between a human tutor and another student exhibiting the impetus misconception.

nothing more than natural language text from a particular domain and the identification of short texts that represent prescriptively good and bad descriptions against which a student text can be compared. For instance, a bad description that should match Figure 1 is the often-observed impetus misconception:

If there is no force on a moving object, it slows down.

These types of classification approaches create a statistically derived form of semantics for a text relative to the training data but do so by treating language as an unordered bag of words in which the organization of the words imparts no meaning. Because these techniques fail to capture this additional semantics, they are insensitive to a number of language phenomena that help distinguish between good and bad explanations. First, they would clearly be insensitive to argument ordering variations making the difference between “x accelerates faster than y” and “y accelerates faster than x” undetectable. Negation is another phenomenon to which these techniques are insensitive. For example, a good explanation followed

Explanation: The pumpkin moves at a constant velocity in the horizonatal direction which is zero. The only force moving the pumpkin is gravitational force. . .

T: I see a topic we should explore. Let's consider what forces act on objects after they have been pushed or thrown. Suppose a person is floating in deep space. The person, who is wearing a space suit of course, pushes a wrench away. While the person is pushing the wrench, what forces are acting on it?

S: Gravity.

T: Uhhh. . . Let's assume that the stars and planets are all so far away that gravitational forces are negligible. So the only force acting on the wrench is the pushing force exerted by the person on the wrench. After the person has pushed the wrench, it floats away from her. Now what forces are acting on the wrench?

S: The force given.

T: Let's look at this more closely. Are there any other forces on the wrench after it leaves the person? If so, what are they?

S: No.

T: Fine. If there was one force acting on the wrench after the person has pushed it away, how would that force affect the wrench's velocity?

S: I don't know.

T: Let's consider this then. If you exert even a small force on a stationary billiard ball, it accelerates and begins to move. If a stationary rocket ship turns on its thrusters, it accelerates and begins to move a little even if the thruster is weak. So if there is even a small force on the wrench, shouldn't it also accelerate?

S: Yes.

T: Okay. So if there is a force on the wrench due to the person after the wrench has left the person, then the wrench should accelerate. Is it really reasonable for the wrench to speed up all by itself after leaving the person? Have you ever seen a billiard ball speed up after being hit? Or a hockey puck speed up after being shot?

S: No.

T: Fine. Because the wrench really doesn't speed up after leaving the person, there is no force acting on it. The force of the person's push ends as soon as the person stops touching the wrench. In general, when do pushing forces, pulling forces and other forces due to the contact of two objects end?

S: When contact is over.

T: Well. . . When one objects exerts a contact force on another object, the force stops when the contact stops. Thus, when the wrench leaves the person's hand, the contact ends, so the pushing force ends. In general, after a person or some other agent has started an object moving but is no longer touching the object, does it still exert a force on the object?

S: Yes.

T: Uhhh. . . No, it does not. Except for field forces, like gravitational or magnetic force, forces only exist between two objects when they are touching each other. Your essay seemed to imply otherwise to me.

Figure 3. A verbatim student explanation and discussion with the Why2-Atlas system during an early evaluation that was completed by 26 students.

by "But I don't think that will happen" would be classified as good because "not" is too general to have a high information value.

A third, even more complex, language phenomenon to which these classification techniques are insensitive is anaphoric expressions (e.g., determining the referent of a pronoun such as "that" in the previous example, or "it"). Although other statistical techniques for pronominal anaphora resolution are highly effective (Strube et al., 2002; Ng and Cardie, 2002), they cannot be directly combined with superficial, statistically derived semantics approaches. As an example of the problem, consider the last clause of the essay shown above in Figure 1:

it will fall at the same place where it left my hands.

This clause would tend to be misclassified as the correct answer "The pumpkin will land in my hands." The reason is that the words "fall," "my," and "hands" have a high information value relative to the expected answer, while the temporal and nominal anaphora involved in "will fall" and "at the same place" do not. Hence, these anaphoric expressions will be overlooked although they change the meaning significantly in this case.

Fourth, the inferences captured by statistical semantics approaches are too weak. In Figure 1, the student has the extreme belief that the pumpkin has **no** horizontal velocity. This would probably not be recognized as a case of "slowing down" by this type of statistical analysis. Even more difficult is that **no** horizontal velocity is not explicit; there is a multistep chain of inference involved that statistical approaches are not equipped to handle. The chain of inference can be informally expressed as "pumpkin's final horizontal position = pumpkin's initial horizontal position" → "pumpkin's horizontal displacement is zero" → "pumpkin's horizontal velocity is zero."

Furthermore, these statistical techniques are often too insensitive to recognize that student statements are true but vague in cases where a few content words are missing. In these cases the tutor should acknowledge the correct statement and elicit more precision rather than continuing as if the statement were wrong or accepting it without requiring more precision from the student. For example, if a student makes a correct statement about an axial component of the velocity of an object but does not report it in terms of the horizontal and vertical components of the velocity, the tutor should ask which was intended.

Although additional preprocessing of the language and postprocessing of the classifications can be done to alleviate some of the problems involved (Rosé et al., 2002), there is no clear workaround for the problem of weak inferencing. To both capture what are subtle differences to statistical semantics classification and address the problem of weak inferencing, we need the precision possible so far only with approaches that try for a deeper understanding of the student's reasoning.

The Geometry Explanation Tutor is an operational prototype that does a deeper semantics classification (Aleven et al., 2001b, 2001a) of student utterances. It parses a student explanation into a propositional representation using a syntactic grammar and lexical semantics and then uses LOOM, a terminological knowledge represen-

tation tool, to classify these relative to prescriptive categories that typically express one proposition. This approach looks promising (Aleven et al., 2002), but the system's goal is to elicit a justification for a single step in a geometry proof; generally, such a justification can be expressed with a single sentence that succinctly translates into a small number of propositions. It isn't clear that this approach will work well for the longer, more complex explanations that the Why2-Atlas system elicits, since it will largely overlook the intersentential, or discourse-level, meaning of the text.

Our approach to the problem of recognizing inferential relationships between sentences is to create a proof based on the student's natural language essay and then check the proof. Why2-Atlas parses student utterances into propositional representations. It uses a syntactic grammar and lexical semantics to create a representation for each sentence (Rosé et al., 2002) and then resolves temporal and nominal anaphora (Jordan and VanLehn, 2002). But instead of classifying the resulting propositions relative to a terminological representation of the domain knowledge, the Why2-Atlas system constructs proofs by using abductive reasoning. *Abduction* is a process of reasoning from an observation to possible explanations for that observation. In this application the observations are the propositions that represent the student's essay, and the proof is the abductive reasoning steps that explain the propositions.

A proof-based approach gives more insight into the line of reasoning the student may be following across multiple sentences because proofs of the propositions should share subproofs. For example, consider the last sentence and part of the first sentence of the essay in Figure 1. The sentences have the informal proof shown in Figure 4, where the first column is a reference number for the proof step; the second column is a gloss of a proposition that is in the student's explanation, or is inferred, or is given; and the third column is the rule or justification for the proposition. The proof for the second sentence is steps 3–7, and the proof for the first sentence is steps 1–4, so that the first sentence is a subproof that supports the second. Moreover, subtle misconceptions such as impetus (as in step 5) are revealed when they must be used to prove a student-supplied proposition.

The proof-based approach also opens the possibility of implementing interactive proof generation through a dialogue with the student. This interaction can serve the dual purpose of revealing the conjectured argumentation behind the student's statement and disambiguating the student's intended meaning when there are multiple proofs. For example, if there are two equally good proofs of a student statement, where one involves a misconception about the relationship between force and velocity and the other involves a misunderstanding of when a particular force is negligible, then we can use the structure of the proof to identify a possible series of disambiguation questions. This example will be further developed in Sections 2 and 6.

Although we could use deductive inference as an approach for building and checking proofs of student explanations, abductive inference is a better choice

Reference Number	Proposition	Justification
1	before the release, the man is holding the pumpkin	given
2	the man exerts a nonzero horizontal force on the pumpkin	*if body1 & body2 in contact then body1 exerts a nonzero force on body2
3	after the release, nothing is is touching the pumpkin	given
4	after the release, the horizontal force is zero	if no contact then contact force is zero
5	the pumpkin's horizontal velocity is zero	*if zero force then zero velocity (impetus)
6	the pumpkin's horizontal displacement is zero	if zero velocity then zero displacement
7	the pumpkin's initial & final positions are equal	if zero displacement then initial and final positions are equal

Figure 4. An informal proof of the excerpt "Once the pumpkin leaves my hand, the horizontal force that I am exerting on it no longer exists. . . . Since no horizontal force acted on the pumpkin from the time it left my hand, it will fall at the same place where it left my hands" (from the essay in Figure 1). Buggy justifications are preceded by an asterisk.

because we are performing a diagnostic task, and we must robustly and efficiently deal with the ambiguity and vagueness introduced by natural language,* students' incomplete proofs, and an incomplete knowledge base.

Although the reasoning system we use within Why2-Atlas has some similarities with other qualitative physics reasoning systems (Weld and de Kleer, 1990) in its ontology and rules, their tasks are different. Most existing systems do the student's task: given a physical system, the reasoner can predict or explain the system's behavior deductively. In our case, the student essay is viewed as a fragmentary, incomplete, and possibly incorrect proof. Our task is to complete that proof insofar as possible.

In this paper we motivate and describe an abductive reasoning system that creates proof-based representations of student essays for tutorial applications. First we give an overview of the Why2-Atlas tutoring system architecture to clarify the context in which the abductive reasoner operates. As we describe the tutoring system, we explain the pedagogical considerations that motivate how we use a proof-based representation of a student's essay and provide an example of how a proof is built and used. We then motivate our choice of weighted abduction for

* Abductive inference has a long history in plan recognition, text understanding, and discourse processing (Appelt and Pollack, 1992; Charniak, 1986; Hobbs et al., 1993; McRoy and Hirst, 1995; Lascarides and Asher, 1991; Rayner and Alshawi, 1992).

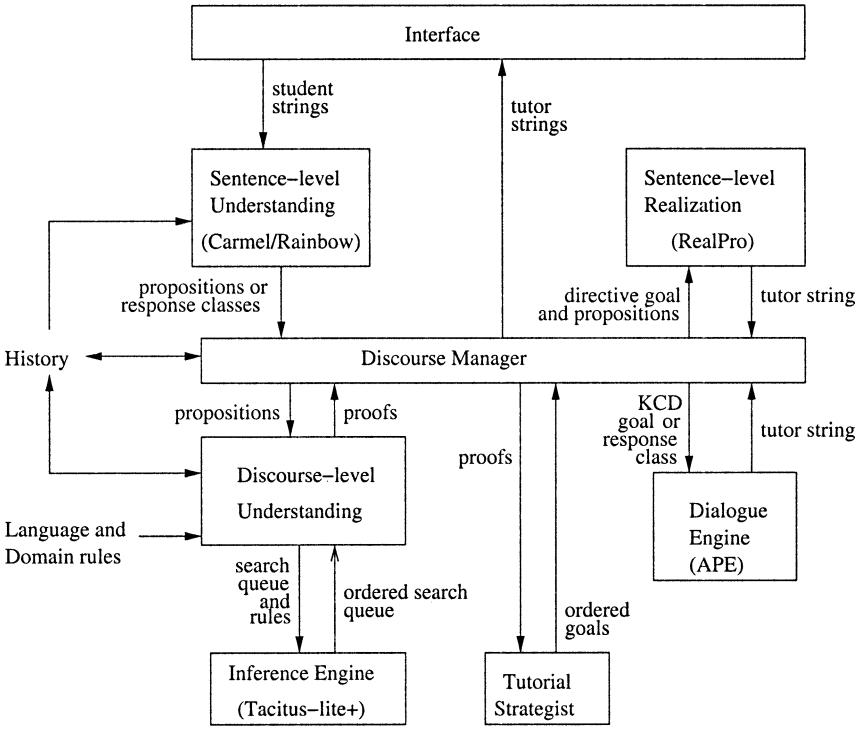

Figure 5. Why2-Atlas tutoring system architecture.

building proofs and explain in detail our abductive inference engine, Tacitus-lite+.* Next we present evaluation results for an early version of the Why2-Atlas system and the results of a subsequent evaluation of the abductive reasoner using a test suite of 45 student essays. Finally, we describe current work in progress and some of our future plans for deriving additional benefits from a proof-based approach for tutoring applications.

2. Building and Using Abductive Proofs

Our discussion in this paper focuses on building proofs using an abductive reasoner where the input is a propositional representation of the student's essay. In this section, we describe the architecture of the entire system as background, so that it is clearer how the input for the proof building is provided and what is done as a result of analyzing the proof. Except for the abductive inference engine module, none of the other system modules described in this section will be addressed in this paper.

* We are using an extended version of SRI's Tacitus-lite weighted abductive inference engine (Hobbs et al., 1993) as our main tool for building abductive proofs.

2.1. THE WHY2-ATLAS TUTORING SYSTEM

The architecture for the current version of the Why2-Atlas qualitative physics tutoring system is shown in Figure 5. The user interface for the system is a screen area in which the physics question is displayed along with an essay entry window and a dialogue window. As the student enters an answer and explanation for a qualitative physics question, the sentence-level understanding module builds sets of propositions to represent sentences as the student enters them. The user interface and the sentence-level understanding components are described in detail in (VanLehn et al., 2002; Rosé et al., 2002).

The sets of propositions are passed by the discourse manager to the discourse-level understanding module. Each set of propositions represents one interpretation of a sentence. The discourse-level understanding module resolves anaphoric expressions and other language dependencies within the sentence representation as described in (Jordan and VanLehn, 2002). It then uses domain reasoning rules and the Tacitus-lite+ abductive inference engine to create a set of proofs.

The proofs that are produced represent the student's knowledge and beliefs about physics with respect to the problem to which the student is responding. One difficulty that must be addressed is uncertainty about the beliefs and knowledge that should be attributed to a student. This uncertainty arises because some of the knowledge and beliefs about the student are inferred based on observed student actions or utterances (Zukerman and Albrecht, 2001). Thus, as with decision-theoretic approaches (Murray and VanLehn, 2000; Keeney and Raiffa, 1976), the system needs to reason about the utility of separately attributing each of these mutually exclusive representations of varying plausibility to the student. Tacitus-lite+ tries to estimate this by associating costs with the proofs it creates by weighted abduction. In weighted abduction, weights are assigned to propositions in the bodies of the Horn clauses in order to compute the cost of assuming a proposition without proof. Assuming a proposition is further referred to as *abducing*, and such a proposition is called an *assumption*. Weighted abduction is explained in more detail in Section 4.

Even with a mechanism for ascertaining the plausibility of alternative proofs, there can still be multiple proofs that are considered equally good representations. Hence, once proofs have been built, the discourse-level understanding module updates the history with the results from Tacitus-lite+ and selects the *best* proofs to send to the tutorial strategist. The tutorial strategist poses relevant communicative goals for itself by analyzing proofs. Acquiring and reasoning about student beliefs and knowledge are central issues addressed by work in student modeling. A student model is a type of user model, and in general a user model provides information the system can use in adapting to the needs of its user (Wahlster and Kobsa, 1989). The Why2-Atlas system uses the proofs derived from the student's essay to identify effective communicative strategies and goals that will (1) effectively help students realize and correct their errors and misconceptions and (2)

enable students to realize what reasoning is necessary when generating a complete explanation.

Currently there are four categories of communicative goals. Two of these, disambiguating terminology and clarifying the essay, are addressed through directives to modify the essay. The other two, remediating misconceptions and eliciting more complete explanations, are addressed through dialogue. Misconceptions are detected when the proof includes a rule that is incorrect or inapplicable. Incompleteness is detected under two conditions. First, there may be multiple proofs that are significantly different and equally plausible. This condition indicates that the student did not say enough in an explanation for the system to decide which proof best represents what the student's reasoning may be. Each possible line of reasoning could point to different underlying problems with the student's physics knowledge. The second condition occurs when the student fails to explicitly state a *mandatory point*, which is a proposition that domain instructors require of any acceptably complete essay. Once the tutorial strategist has identified communicative goals, it ranks them according to curriculum constraints and sends them to the discourse manager. The discourse manager selects the highest-priority goal after taking dialogue coherency into account and sends the goal to either the dialogue engine or the sentence-level realization module.

In an educational context it is generally more effective if students discover their own errors and misconceptions rather than always simply being told of the error and its correction. Therefore, the dialogue engine initiates and carries out a dialogue plan that will either help the student recognize and repair a misconception or elicit a more complete explanation from the student. The main mechanism for addressing these goals are what we call a knowledge construction dialogue (KCD) specification. A KCD specification is a hand-authored push-down network. Nodes in the KCD network are either the system's assertions and questions to students or pushes and pops to other networks. The links exiting a node correspond to anticipated responses to the question. Each assertion and question are a canned string, ready for presentation to a student. The dialogue engine is described in detail in (Rosé et al., 2001).

If the tutorial strategist's analysis of the proofs that represented the student's essay reveals a misconception or error, then the dialogue engine will engage the student in a knowledge construction dialogue (KCD) that works through an analogous, but simplified, problem and summarizes at the end with a generalization of the reasoning that the student is expected to transfer to the current problem. If incompleteness is revealed by the analysis of the proof, then the system will engage the student in a KCD that leads the student to express the missing detail by reminding the student of an appropriate rule of physics, and a fact that is relevant to the premise or conclusion of the rule, and then asking the results of applying the rule.

Working through an analogous problem is currently the only technique implemented in the system for leading a student to recognize an error or misconception.

Another possibility is to step through the reasoning associated with the current problem and ask the student to fill in any missing details. Having some of these details wrong may have led the student to draw a wrong conclusion, and with the corrected details the student may be able to easily see the error. Other techniques for dialogue strategies to correct misconceptions, errors, and incompleteness may be derivable from argumentation strategies used in argument generation as described in (Zukerman et al., 2000) (e.g., reductio ad absurdum, premise to goal, and reasoning by cases).

The other communicative goals, disambiguating terminology and clarifying the essay, are addressed by the discourse manager as directives for the student to modify the essay. It passes propositions and a goal to the sentence-level realization module, which uses templates to build the deep syntactic structures required by the RealPro realizer (Lavoie and Rambow, 1997) for generating a string that communicates the goal.

While a dialogue is in progress, the discourse-level understanding and tutorial strategist modules are currently bypassed until the essay is revised. Once the student revises the essay, it is reanalyzed, and the cycle repeats until no additional communicative goals arise from the system's analysis of the essay.

2.2. EXAMPLES OF BUILDING AND USING PROOFS

The Tacitus-lite+ abductive reasoner currently has 105 qualitative physics rules available to use in building proofs, where propositional representations of a student's sentences are input as observations that are to be explained. These rules cover seven problems as well as parts of many other problems. Figures 6 and 7 are examples of two simplified alternative abductive proofs for sentence (1).

The pumpkin slows down. (1)

For these examples, we take as given the fact that the air resistance is 0 and that the runner is not applying a horizontal force to the pumpkin after he throws it. Since students often overlook relevant givens, proofs that ignore these givens can be considered as well whenever the given is represented by a rule and a buggy counterpart is also included (as described in Section 3.4).

Each level of downward arrows from the gloss of a proposition in the two alternative proofs shown in Figures 6 and 7 represent a domain rule that can be used to prove that proposition. To simplify the example, we assume that the weights in all the rules are evenly divided between the propositions in the body of each rule. The number in parentheses at the end of each proposition represents the cost of abducing the proposition.

In both proofs shown in Figures 6 and 7, one way to prove that the velocity of the pumpkin is decreasing is to infer, through the rule *Imprecision*, that the horizontal component of the velocity vector was meant to be decreasing. The system will also build alternative proofs in which it tries to prove that the student means the vertical

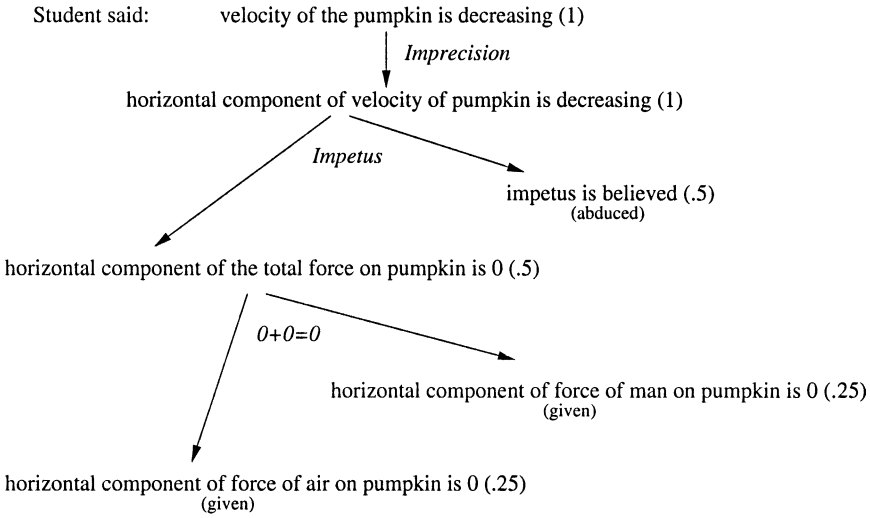

Figure 6. Example of one possible simplified abductive proof for "The pumpkin slows down." Rule names are in italics; arrows are in the direction of abductive inference. Total cost of the proof is .5.

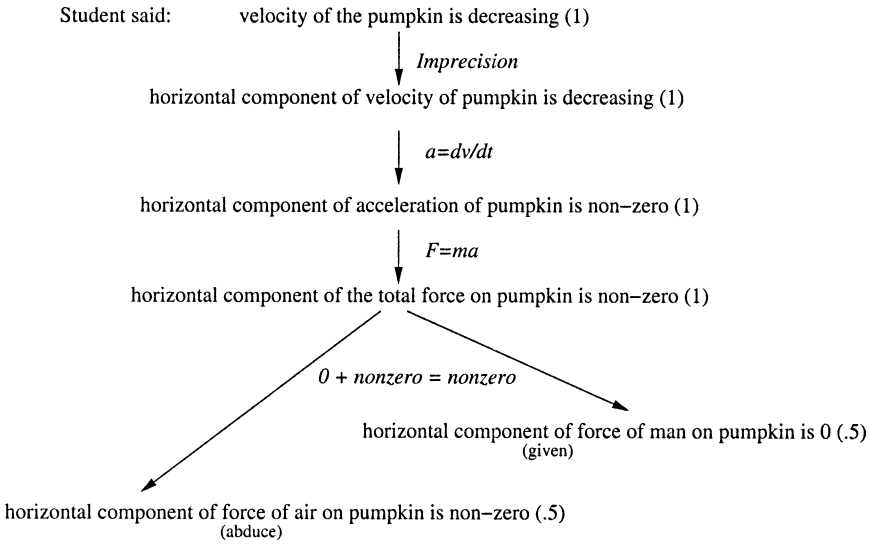

Figure 7. Example of an alternative simplified abductive proof for "The pumpkin slows down." Rule names are in italics; arrows are in the direction of abductive inference. Total cost of the proof is .5.

velocity instead (especially because, during certain time intervals, this is true), but for this example we will ignore these other proofs.

Next we consider two ways of proving that the horizontal component is decreasing. First, we consider the case of the proof in Figure 6. In this case Tacitus-lite+ has selected a buggy physics rule that is one manifestation of the impetus miscon-

ception; the student thinks that a force is necessary to maintain a constant velocity. In this proof it is abduced that the student has this bug at a cost of .5, and no further attempts are made to prove it. Alternatively, the system could try to gather more evidence that this is true by asking the student diagnostic questions.

Next, Tacitus-lite+ proves that the total force on the pumpkin is zero by proving that the possible addend forces are zero. Since it is a given that air resistance is negligible, this proposition unifies with this given fact for zero cost. Likewise, since we said that it was also given that the man is applying a horizontal force of 0 to the pumpkin after he throws it, this proposition unifies with the given fact for zero cost as well. Since the proof contains just one assumption, that the student has the impetus bug, the total cost of the proof is .5.

Looking again at the alternative proof in Figure 7, we see that it attempts to prove the horizontal component of the velocity is decreasing by first trying to prove that the horizontal component of the acceleration is nonzero in the direction opposite the velocity. To prove this, we must prove that the total horizontal force on the pumpkin is nonzero in the same direction as the acceleration. One approach is to prove that at least one of the addend forces is nonzero. The system can ignore either of the two givens at this point in order to try to prove that there is at least one nonzero force on the pumpkin. In this case it tries to prove that wind resistance is not negligible; but since it cannot prove this, the result must be abduced at a cost of .5. So the total cost of this alternative proof is .5 as well. In this example, the system now has two plausible proofs with no means of choosing between them without more information from the student.

If the student had instead supplied the sentence

The pumpkin slows down because there is no horizontal force on it. (2)

which provides some justification, the proof in Figure 6 would be preferred. This is because the proposition representing this justification conflicts with the inferred proposition in Figure 7 that the total horizontal force is nonzero. This new proposition will not add an additional cost to the proof in Figure 6 because it unifies with a proposition that has already been proven.

In the case of the proof in Figure 6 the tutorial strategist identifies a dialogue goal to address the impetus misconception, since an impetus bug assumption is part of the proof. In the case of the proof in Figure 7 it identifies a goal to address the wrong assumption that air resistance is nonnegligible.

In addition to identifying errors and misconceptions, the system can also give some direct, constructive feedback on an essay relative to the proof in response to certain kinds of vagueness. For example, with the proofs in Figures 6 and 7, when the system attempts to prove that the student means either the horizontal or vertical velocity, it triggers a clarification question asking the student to clarify whether the horizontal or vertical velocity is meant.

Now that the context for building and using proofs is established, we will focus solely on the details of how the abductive reasoner creates the proofs given propositional representations of a student's essay as input.

3. Qualitative Reasoning in Mechanics

The development of qualitative mechanics has been driven largely by the needs of automated reasoning about physical systems. The applications range from monitoring and engineering design to education. In education, qualitative physics has been used in modeling and design environments (Forbus et al., 2001), an example of which is Articulate Software (Forbus, 1997).

The subset of physics that Why2-Atlas addresses motivates the ontology for the propositional representations on which the theorem prover operates. The goal to model and check the correctness of the student's reasoning motivates the types of rules included.

3.1. THE STRUCTURE OF QUALITATIVE PROBLEMS

The qualitative problems that we have chosen are from a first-year college course in mechanics and have several differences from the problems addressed in the analysis of dynamical systems, which is the common domain for previous automated qualitative reasoning systems.

The problems in our domain, unlike design problems, usually have all the bodies explicitly defined in the problem statement. In several situations the descriptions of bodies ("you throw a pumpkin") allow for more than one idealization of a composite body (`man` and `pumpkin` versus `man`, `hand` and `pumpkin`).

The initial conditions are relatively unambiguously specified in the problem's description. This and other features of the problems we select ensures that the *envisionment*, as defined in (de Kleer, 1990), normally includes only one acceptable sequence of events. However, similarly to the case of composite bodies, various partitionings of the timeline into intervals are possible.

After the bodies and time intervals are identified, the next step in the solution of the problem is to choose the sequence of time intervals (usually one or two) as *main time intervals* at the beginning of which sufficient knowledge about physical quantities (*initial conditions*) can be obtained from the givens, so that a sequence of inferences will result in an appropriate conclusion about the sought physical quantity (which is usually related to the time instant at the end of the sequence of main time intervals).

In the case of the pumpkin problem (Fig. 1), the possible sets of main intervals include, for example, the following two:

1. (a) Man is pushing the pumpkin up; (b) the pumpkin is flying.
2. The pumpkin is flying.

The initial conditions corresponding to the first partitioning would include a zero initial vertical velocity and a nonzero upward force from the man. From these conditions we can infer (among other things) that

- from Newton's first law, the pumpkin's horizontal velocity is not changing during the pushing stage,
- from continuity of the velocity at the instant of the release, the horizontal velocity is still the same at the beginning of the second time interval (the flight) as it was before.

Human tutors, however, normally also accept a solution based on the second choice of the main interval. In this case the student could obtain the initial condition of the pumpkin having the same horizontal velocity as the runner at the beginning of the flight directly by interpreting the given "You throw a pumpkin straight up." Note that different choices of the main intervals correspond to different idealizations of the problem.

After the idealization stage is complete, the student has to apply qualitative inference rules to produce a solution. Thus, for the second choice of the main time interval above, the following is an acceptable solution:

- Only the force due to gravity acts on the pumpkin during the flight so it has no horizontal acceleration (an application of Newton's first law).
- Zero horizontal acceleration implies that the horizontal velocity of the pumpkin is constant during the flight.
- The constant horizontal velocity during the flight is equal to its value at the beginning of the flight, namely, to the velocity of the man (from the initial conditions).
- Therefore, the pumpkin and man have the same horizontal velocity during the flight, so the pumpkin will always be above the runner until it falls back into his hands (two bodies with the same initial position and same velocity over a time interval have the same positions over this time interval).

This set of inferences (an essay) can be viewed as a qualitative proof of the answer to the problem ("it falls back into his hands"). Note that the student's actual natural language argument can be presented in reverse order (or even some other ordering) and can include irrelevant steps. In the case of a different presentation order, the natural language used signals the underlying ordering of the steps involved so that after the discourse-level understanding module resolves anaphoric expressions and other language dependencies, the underlying order of the argument is revealed. If the underlying ordering is incorrect, then it can be addressed by anticipating typical incorrect orderings with buggy rules. Irrelevant steps are handled in one of two ways: (1) typical irrelevant steps are anticipated with buggy rules, or (2) the step is assumed without proof at a high cost (see Section 4.2) and can be presented to the student.

3.2. EFFECT ON THE IMPLEMENTATION

The goal of our reasoning engine is twofold. First, we'd like to know the logical steps the student did not mention explicitly in his essay. Second, we want to reason about the correctness of these hidden steps as well as of the statements in the essay.

Note that unlike many of the systems for automated reasoning in qualitative physics, we do not solve the physics problem. Also, since the problems do not deal with complex envisionments, we do not have to reason about envisioning. Our target is mainly the idealization and the following stage.

Not having to represent envisionments consisting of multiple plausible scenarios of events allows us to largely avoid one of the major difficulties facing the developers of qualitative physics problem solvers – implementation of a vast amount of common-sense knowledge.

3.3. QUALITATIVE PHYSICS ONTOLOGY

The Why2-Atlas ontology is inspired by that used in previous qualitative physics reasoning work. In particular, for both ontology and rules, we borrowed extensively from (Ploetzner and VanLehn, 1997), making appropriate simplifications given the subset of physics the system is addressing. The ontology is further adapted to take advantage of the knowledge representation facilities of the Tacitus-lite+ abductive reasoning engine, such as ordered sorts, which are described in detail in Section 3.5. Until then, less formally, a *term* is defined as either a *variable* or a *constant* (there are no functions in Tacitus-lite+). Terms are assigned *sorts* from a partially ordered set of sort symbols, such that every term has a unique least sort.

The Why2-Atlas ontology comprises bodies, physical quantities, states, times, and relations, each of which we describe below in more detail.

3.3.1. *Bodies*

The physics problems in our scope deal only with solid bodies, with the possible exception of air, which occurs only in the context of air resistance. We distinguish bodies with respect to their contact properties: bodies that generally require contact to exert a force are of sort `Regular-body`; the other bodies, such as planets, are said to be of sort `Special-body`. For the sake of simplicity, all the forces in our ontology have a corresponding pair of bodies; therefore, we treat air, in the context of air resistance, as a special body.

Regular bodies usually have the semantics of point masses. The few exceptions are handled with ad hoc axioms (see, for example, contact states in Section 3.3.3).

3.3.2. *Physical Quantities*

The constants of the sort `Quantity1b` that represent vector quantities attributed to a single body are `position`, `displacement`, `velocity`, `acceleration`, and `total-force`. The sort `Quantity2b` for vector quantities involving two bodies has

Table I. Slots of a vector quantity of sort Quantity1b.

Description	The Generic Sort of the Filler
Quantity	Quantity1b
Identifier	Id
Body (or two bodies in the case of force)	Body
Axial component or not	Comp
Qualitative derivative of the magnitude	D-mag
Quantitative derivative of the magnitude	D-mag-num
Zero or nonzero magnitude	Mag-zero
Quantitative magnitude	Mag-num
Sign for axial component	Dir
Quantitative direction	Dir-num
Qualitative derivative of the direction	D-dir
Beginning of time interval	Time
End of time interval	Time

- Quantity1b = {position, displacement, velocity, ... }
- Id
- Comp
  - Axial = {horizontal, vertical}
  - No-comp = {no-comp} (i. e. full vector)
- D-mag
  - Constant = {constant}
  - Nonconstant = {increase, decrease, varying}
- D-mag-num
- Mag-zero = {zero, nonzero}
- Mag-num
- Dir = {pos, neg}
- Dir-num
- D-dir = {constant, nonconstant}
- Time = *Problem-specific constants*

Figure 8. Fragment of the sort hierarchy.

a single member in our ontology, the constant force. The constants of the sort Scalar are duration, mass, and distance.

Every vector quantity has slots and respective restrictions on the sort of a slot filler as shown in Table I. The hierarchy of sorts from Table I (except for sort Body, which was described before) is shown in Figure 8. The names of sorts begin with an uppercase character; the names of constants begin with a lowercase character. Note that sorts Id, D-mag-num, Mag-num, and Dir-num do not have subsorts or constants. Variables of these sorts are used only for cross-referencing between atoms (see Section 3.5).

3.3.3. *States*

Individual bodies can be in the following *states*: `vacuum` or `freefall`. Being in one of these states implies respective restrictions on the forces applied on the body.

A special state between two bodies is `contact`. The contact between two bodies can be `attached` – the bodies can exert mutual forces and the positions of two bodies are equal; `detached` – the bodies do not exert mutual forces (except for possibly the forces due to gravity); or `moving-contact` – the bodies can exert mutual force (no conclusion on the respective positions). The last type of contact is introduced to account for the fact that we often want to treat bodies as point masses capable of pushing or pulling each other for certain time intervals (a nonimpact type of contact), for example the man pushing the pumpkin up.

3.3.4. *Time*

The current representation of time most closely resembles the method of *temporal arguments* (Haugh, 1987), with a limited number of arguments per predicate, but our predicates take a mix of temporal and nontemporal arguments, similar to (Bacchus et al., 1989).

We use time instants as basic primitives. A time interval is a pair (t_i, t_j) of instants. This definition of time intervals is sufficient for implementing the semantics of *open time intervals* in the context of the mechanics domain.

3.3.5. *Relations*

The multiplace relations are represented in Table II. The respective hierarchy of sorts is shown in Figure 9. The relation `non-equal` can be used for any pair of terms. There is no explicit relation for equating arbitrary terms. Instead, substitution is used to ensure that equal terms have the same names. The relation `before` relates time instants in the obvious way. The relation `rel-position` provides the means to represent the relative position of two bodies with respect to each other, independently of the choice of a coordinate system – a common way to informally compare positions in natural language. The relation `compare` provides the means to represent the ratio and difference of the magnitudes of two quantities and, for quantities changing with time, the magnitudes of the derivatives of two quantities. The ralation `compare-dir` represents the relative directions of two vector quantities.

3.4. RULES

A distinctive feature of the task of modeling the student's reasoning is that it becomes necessary to account for erroneous facts and rules. False facts corresponding to a wrong idealization are called *buggy givens*. Other false facts are typically conclusions that students make by applying false domain rules and are modeled by buggy domain rules and buggy metaknowledge rules.

Table II. Relations.

Relation	1st and 2nd Arguments	3rd Argument	4th Argument
non-equal	any terms		
before	Time		
rel-position	Body	Rel-location	
compare	Mag-num or D-mag-num of any scalar or vector quantity	Ratio	Difference
compare-dir	Dir-num of any vector quantity	Rel-dir	

- Rel-location = {at, nonequal, to-left-of, below, etc.}
- Ratio
  - Greater-than-one = {two, etc.}
  - One = {one}
- Difference
  - Nonzero
  - Zero = {zero}
- Rel-dir
  - Parallel
    - Collinear = {codirected, opposite}
    - Non-collinear = {non-collinear}
  - Non-parallel = {orthogonal, non-orthogonal}

Figure 9. Sort hierarchy for arguments of relations.

3.4.1. *Idealization*

The canonical idealization of the problem is formalized as *givens*, or facts, for the abductive prover. The facts that may be misunderstood by the student because of a possibly wrong idealization are represented as pairs of *correct and buggy givens*. In the context of a student's reasoning about the problem, buggy givens are wrong assumptions the student made during idealization.

For example, with the pumpkin problem, the facts

→ the force of air resistance on the pumpkin is zero

and

→ the force of air resistance on the pumpkin is nonzero

are a pair of correct and buggy givens, respectively.

The facts that we consider to be common knowledge that are shared by the student (i.e., we do not account for possible misunderstandings of those facts) are represented as *givens*. Thus, assuming that no misunderstanding about the trajectory of the man is possible, we can define as a given

→ the vertical position of the man is constant at all times.

This pairing of correct and buggy givens is subject to integrity constraints – only one member of the pair can be in any given proof (see Section 4.5). The intention is to reduce the search space during proof generation; it represents a risky assumption that we have made: *Students rarely believe both of the givens in an inconsistent pairing within the same proof.*

3.4.2. *Metaknowledge*

To account for any wrong rules that students may be applying when they come up with wrong conclusions, we pair the good version of such a rule with its buggy counterpart. This pairing of correct and false rules is subject to integrity constraints that are analogous to those for pairings of correct and false givens. Again, although this approach reduces the search space during proof generation, it assumes that students do not believe both of the rules in an inconsistent pairing.

3.4.3. *Qualitative Newtonian Mechanics*

Currently we are focusing on the problems of kinetics of a rigid body in a plane of noncircular motion. Thus, the domain axioms cover qualitative kinematics and qualitative versions of Newton's laws and their derivatives. Since some problems are essentially two dimensional, we have also implemented a basic algebra for vector components.

Currently, there are 24 idealization rules (excluding problem-specific givens that are assumed to be shared knowledge), 24 metaknowledge rules, and 57 rules of qualitative Newtonian mechanics.

3.5. KNOWLEDGE REPRESENTATION

The domain propositions described above are represented in the theorem prover by using order-sorted first-order logic (FOL) (see, for example, (Schmidt-Schauß, 1989; Walther, 1987)).

As we mentioned earlier, a *term* for us is a *variable* or a *constant*. Tacitus-lite+ does not provide any built-in support for functions since function-free clauses are the natural output from the Sentence-Level Understanding module (see Section 2). Every term has a sort specification that maps it to a member of a partially ordered set of sorts.

Tacitus-lite+ allows for the use of predicate variables, which can also be assigned sorts. This is encapsulated in the framework of FOL by grouping a predicate name together with its arguments as arguments of a metapredicate M_i, where i is the total number of resulting arguments. Since there is exactly one metapredicate symbol for each arity, metapredicate symbols can be safely omitted. Every such representation of an atom is augmented with a corresponding sort specification for the argument terms. For example, "Horizontal velocity of the pumpkin is decreas-

ing" is represented as shown below (where constants are in the lower-case script, sorts begin with an upper-case letter and variables begin with "?"):

```
((velocity v1 pumpkin horizontal decrease
            ?d-mag-num ?mag-zero ?mag-num ?dir ?dir-num ?d-dir ?t1 ?t2)
 (Quantity1b Id Regular-body Axial Nonconstant
               D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time))
```

Each atom is indexed with a unique *identifier*, a constant of sort `Id`, which is used for cross-referencing. For example, "Force of gravity acting on the pumpkin is constant and nonzero" has the following representation in which the *identifiers* `f1` and `ph1` appear as arguments in the `due-to` predicate:

```
((force f1 ?body1 pumpkin ?comp constant
              ?d-mag-num nonzero ?mag-num ?dir ?dir-num ?d-dir ?t1 ?t2)
 (Quantity2b Id Body Regular-body Comp Constant
               D-mag-num Mag-zero Mag-num Dir Dir-num D-dir Time Time))
((due-to d1 f1 ph1) (Due-to Id Id Id)
((phenomenon ph1 gravity) (Phenomenon Id Field-interaction))
```

There is no explicit negation. Instead, a negative student statement is represented as a conjunction of atoms with appropriate arguments whenever possible. Thus, the fact that "there is no force" is represented as the force being zero. This simplification is intentional and is done to avoid the problem of finding the scope of negation in natural language text. The version of the system currently under development extends the knowledge representation to cover disjunctions, conditional statements, and certain types of negations (see Section 6).

Rules in Tacitus-lite+ are in the form of extended *Horn clauses*; namely, the head of a rule can be a conjunction of atoms. For example, a rule that states "if the velocity of a body is zero over a time interval then its initial position is equal to its final position" is represented as follows:

```
((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))
```

4. Weighted Abduction and Tacitus-lite+

4.1. ABDUCTION

Abduction is a process of reasoning from an observation to possible explanations for that observation. In the case of the Why2-Atlas system the observations are what the student said, and the possible explanations for why the student said this are the qualitative physics rules (both good and bad) and orderings of those rules that support what the student said. In order to arrive at the explanation, some assumptions have to be made along the way, since all the inferences that underlie an explanation will not be expressed.

Formally, an *abductive framework* can be defined as a triple $\langle T, A, I \rangle$, where T is a theory, A is the set of abducible literals, and I is a set of integrity constraints (Kakas et al., 1998; Paul, 1993). Then an *abductive task* for a given sentence G (observation), is to find a set $\Delta \subseteq A$ such that

$$T \cup \Delta \models G, \tag{3}$$

$$T \cup \Delta \text{ satisfies } I. \tag{4}$$

In the case of an *abductive logic programming framework*, and in the context of Tacitus-lite, T is the set of *givens* and *rules* of the logic program. Any literal can be abduced in our implementation, and the semantics of satisfying the integrity constraints I follows the *consistency view* as described in Section 4.5.

Naturally, more than one solution may exist for the abductive task. Often it is required that the solution Δ be minimal, namely, that no proper subset Δ' of Δ have the property $T \cup \Delta' \models G$. For the purpose of modeling the student's reasoning, however, other factors that influence the choice of solution may be more relevant, as elaborated on in the next section.

4.2. CRITERIA FOR SELECTING AN ABDUCTIVE EXPLANATION

Various approaches are possible to define a preferred explanation among all admissible ones. (Leake, 1995) distinguishes between plausibility criteria and goal-based criteria. The following categories of plausibility criteria are identified: structural minimality, proof-based criteria, probabilistic/cost-based criteria, and criteria based on analogy with the previous explanations. The goal-based criteria are the factors that depend on the intended use of the explanation: as Leake notices, "A good explanation in a humorous context may be one that is farfetched or obviously false" (Leake, 1995).

For our task of building a model of the student's reasoning, a combination of a number of these criteria is used. Informally, we formulate our preference as "the less deep, the fewer incorrect rules, and the smaller total cost of assumptions." More formally we would like to maximize a certain function of measures of utility and plausibility.

The utility measure is a goal-based criterion that estimates 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. It gives preference to the shallow proofs, which reflects our assumption of cognitive economy: if a short proof and a long proof both explain the student's utterance, and all rules and assumptions in both proofs are equally likely, then the short proof is the more likely interpretation. Of course, comparison of the depths of proofs is complicated by the fact that the rules in the theorem prover are not all of equal importance in the context of the solution. Thus, some steps of the formal proof can be safely omitted in an actual solution provided by an expert. In the context of using the proof as a student model, this preference makes the model optimistic about the student's skills. In the context of using the proof for guiding tutoring feedback, a shallow proof has greater utility because according to our assumption, it is the type of the proof the tutor would prefer to talk about. Another factor that contributes to the utility is the preference for explanations that use good physics as opposed to "buggy" physics.

Since an explicit estimation of utility requires the generation of multiple proofs and is therefore computationally expensive, we deploy a number of proof search heuristics in an attempt to optimize the combination of the two measures. Although currently the parameters of these heuristics are fixed for the duration of the tutoring session, our implementation allows for varying the parameters on the fly. This may be useful for dynamic adjustment of the student model, for example when there is an indication that the model should be more pessimistic about the student's skills (more on the heuristics in Section 4.6).

While the depth preference is neutral to the content of the explanation and the correctness preference gives only binary output for each rule, the cost-based criteria make it possible to take into account the relative plausibility of individual hypotheses. Thus, cost-based abduction, as with the approach defined in (Charniak and Shimony, 1990) and applied to natural language understanding applications, assigns quantitative costs to the hypotheses and orders the explanations by the total cost of their hypotheses.* The cost of a hypothesis is fixed and therefore is not sensitive to such factors as (a) the relative plausibility of the goals (observations) to be explained, (b) the explanatory chain that generated this hypothesis, and (c) the relative plausibility of the antecedents of a particular rule.

This motivated us to choose another approach, weighted abduction (Hobbs et al., 1988), which attempts to avoid these limitations by defining the cost of a hypothesis

* It was also shown in (Charniak and Shimony, 1990) and in (Poole, 1993) that belief revision in Bayesian networks can be accurately modeled by cost-based abduction. That is, when costs are chosen appropriately for the conjuncts of the rules, and the proof graph produced by applying them to explain an utterance inherits those costs as conditional probabilities, then the resulting network is a Bayesian network and thus can produce mathematically sound posterior marginal probabilities (Conati et al., 2002).

as a function of the explanatory chain that led to it and the cost of the goal at the head of the chain. The drawback of weighted abduction in comparison to cost-based abduction, however, is the lack of a precisely defined semantics of weights. We do not attempt to provide a formal definition of such semantics in this paper; instead, we use ad hoc heuristics that are suitable for our particular application.

4.3. WEIGHTED ABDUCTION

Following the weighted abductive inference algorithm described in (Stickel, 1988), Tacitus-lite is a collection of rules where each rule is expressed as a Horn clause. Further, each conjunct p_i has a weight w_i associated with it:

$$p_1^{w_1} \wedge \cdots \wedge p_n^{w_n} \rightarrow r. \tag{5}$$

The weight is used to calculate the cost of abducing p_i instead of proving it, where $cost(p_i) = cost(r) \cdot w_i$. 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: (1) abduces the atom at the cost associated with it, (2) unifies it with an atom that either is a fact or has already been proven or abduced (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), or (3) attempts to prove it with a rule. Tacitus-lite calls the second action *factoring*.

All possible proofs could be generated. However, Tacitus-lite allows the applications builder to set depth bounds on the number of rules applied in proving an observation and on the global number of proofs generated during search. Tacitus-lite maintains a queue of proofs where the initial proof reflects abducing all the observations and each of the three above actions adds a new proof to the queue. The proof generation can be stopped at any point, and the proofs with the lowest cost can be selected as the most plausible proofs for the observations.

Tacitus-lite uses a best-first search guided by heuristics that select which proof to expand, which observation or goal in that proof to act upon, which action to apply, and which rule to use when that is the selected action. As we mentioned, most of the heuristics in Why2-Atlas are specific to the domain and application.

SRI's release of Tacitus-lite was subsequently extended as part of the research project described in (Thomason et al., 1996) and was named Tacitus-lite+ at that time. We are using two main extensions from that work: (1) proofs falling below a user defined cost threshold halt the search and (2) a simple system of variable sorts reduces the number of rules written and the size of the search space (Hobbs et al., 1988, p. 102).

Unlike the earlier applications of Tacitus-lite+, which used it solely for reasoning about language, Why2-Atlas also uses it for shallow qualitative physics reasoning. To support qualitative physics reasoning, we've made a number of general

inference engine extensions, such as improved consistency checking and allowing the rule author to express both good and bad rules in the same rule set.

While computing the minimal explanation with respect to many kinds of prioritization is known to be NP-hard (Bylander et al., 1991; Charniak and Shimony, 1994; Eiter and Gottlob, 1993; Selman and Levesque, 1990), polynomial algorithms have been found for some useful classes of abductive problems (Eshghi, 1993), including cost-based abduction (Santos and Santos, 1996). To the best of our knowledge no such promising complexity results exist for the problems specific to weighted abduction. We are still searching for the best heuristics to use with our domain and application.

4.4. ORDER-SORTED ABDUCTIVE INFERENCE

Let S be a set of *sort symbols* with a partial order $\preccurlyeq$. A *sorted term* is a pair (t, τ), denoted as $t : \tau$, where t is a term (a constant or a variable in our case) and τ is a sort symbol. 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.

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).

Order-sorted deduction has received extensive treatment on its own and as an extension of unsorted logics (Walther, 1987; Cohn, 1989; Kaneiwa and Tojo, 2001; Frisch, 1991). 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 following property:

$$\exists x \ \tau_i(x) \wedge \tau_j(x) \rightarrow (\tau_i \preccurlyeq \tau_j) \vee (\tau_j \preccurlyeq \tau_i), \tag{6}$$

where $\tau_i \preccurlyeq \tau_j$ is equivalent to $\forall x \tau_i(x) \rightarrow \tau_j(x)$. Without loss of generality for the rest of this section, we will use binary predicates and constant-free atoms. The latter can be easily achieved in order-sorted logic by transforming an ordered atom $p(a) : (\tau)$ that includes a constant a into $p(x_a) : (\tau_a)$ and creating a new variable x_a and a new sort symbol $\tau_a \in S$ such that $\tau_a \preccurlyeq \tau$. We will also assume that the rules of the order-sorted logic program T are *nongeneralizing*, that is, for any rule of the form

$$p(x, y) : (\tau_1, \tau_2) \leftarrow q(x, z) : (\tau_3, \tau_4), \tag{7}$$

it holds that $\tau_1 \preccurlyeq \tau_3$. If for rule (7) this condition doesn't hold (and therefore, according to (6), $\tau_3 \preccurlyeq \tau_1$ must hold), it can be transformed into the nongeneralizing form by substituting the sorts for the terms in the head of the rule by the most specific sorts (for the respective terms) found in the body of the rule. For example, rule (7), where $\tau_3 \preccurlyeq \tau_1$, can be transformed into the nongeneralizing rule

$$p(x, y) : (\tau_3, \tau_2) \leftarrow q(x, z) : (\tau_3, \tau_4). \tag{8}$$

It is easy to see from the set-theoretic semantics of sorts that this transformation is model preserving.

Given the constraint (6) on the sort hierarchy, modus ponens can be extended to sorted deduction as follows:

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

Similar to (Kakas et al., 1998) our abductive reasoning procedure interleaves consistency check and backchaining stages. Briefly, the procedure can be described as follows:

1. Unify the goal with the head of the rule.
2. If unification succeeds, apply the unifier to the body of the rule and generate the candidate new goals from the atoms in the body of the rule.
3. Check whether the candidate new goals satisfy the consistency constraints. If the constraints are violated, mark the pair (goal, rule) as not applicable.
4. If the consistency constraints are satisfied, (a) add the candidate new goals to the list of goals; (b) remove the goal from the list of goals; and (c) add the goal and the rest of the atoms from the conjunction in the head of the rule (having applied the unifier to them as well) to the list of provens (which is used in the consistency check).

4.5. CONSISTENCY CONSTRAINTS

Our definition of what it means for a knowledge base $T \cup \Delta$ to satisfy an integrity constraint $\phi \in I$ is most closely related to the *consistency view*; see, for example, (Kakas et al., 1998):

$$T \cup \Delta \quad \text{satisfies} \quad \phi \quad \text{iff} \quad T \cup \Delta \not\models \neg\phi.$$

The particular integrity constraint we wish to enforce is

$$\neg[p \wedge p^*] \tag{9}$$

for every atom p, where p^* means an *opposite* of p, following the approach described in (Kakas et al., 1998; Eshghi and Kowalski, 1989). The definition of an opposite of an atom is domain specific, and for a given atom an opposite is not necessarily unique. For example, in the domain of qualitative mechanics, one of the opposites of "velocity of pumpkin is constant" is "velocity of pumpkin is nonconstant"; another is "velocity of pumpkin is increasing." More formally, every predicate p has a distinguished subset of argument places, called *functional arguments*, with the following property: There is a mapping (specific to p) from groundings of functional arguments to groundings of the rest of the arguments

(although this mapping may be unknown). For example, if in binary predicate p the first argument is functional (and the second is not), then for a given atom $p(x, y) : (\tau_1, \tau_2)$, the set P^* of opposites would be as follows:

$$P^* = \{p(x, y') : (\tau_1, \tau_2') \mid \tau_2' \text{ is incomparable to } \tau_2\}.$$

In terms of provability, the abductive explanation Δ is said to satisfy constraint (9) if and only if for every atom p,

$$T \cup \Delta \nvdash p \wedge p^*, \quad \forall p^* \in P^*. \tag{10}$$

For the sake of computational efficiency we do not implement the completeness part of the semantics of negation as failure (NAF), which requires that one of the following must hold: $T \cup \Delta \vdash p$ or $T \cup \Delta \vdash p^*$. Neither do we do a full implementation of constraint (10) because in this case each step of a proof must be checked by testing whether each opposite of the atom is provable with no new steps or with steps that cost less than the proof of the original atom. As suggested in (Appelt and Pollack, 1992) in the case of weighted abduction one should settle for incomplete consistency checking and focus on detecting the inconsistencies that are most likely to arise in the application domain.

Instead of implementing (10), we prevent abductive inference on rules that would immediately give rise to a new goal $p^* \in P^*$ when the proof generated so far has atom p. Namely, we guarantee that the following holds for every atom p:

$$p \wedge p^* \notin T \cup \Delta \cup \text{Proof}, \quad \forall p^* \in P^*. \tag{11}$$

For example, the atoms corresponding to the pair of statements

velocity of pumpkin is increasing

and

velocity of pumpkin is nonconstant

are consistent (since constant `increasing` is of sort `Nonconstant`), while the atoms corresponding to the statements

velocity of pumpkin is increasing

and

velocity of pumpkin is constant

are inconsistent (since constant `increasing` is of sort `Nonconstant` which is incomparable with sort `Constant`).

As an example of the above, consider a fragment of a proof tree starting from the subgoal "horizontal velocity of pumpkin is constant" as shown in Figure 10. First, assume that the fact "total vertical force on pumpkin is a nonzero constant,"

```
''horizontal velocity of pumpkin is constant''
```
Rule 24: "The magnitude of a vector is constant →
the magnitude of every component of the vector is constant"
```
''velocity of pumpkin is constant''
```
Rule 13-int: "Acceleration of a body is zero →
velocity of the body is constant"
```
''acceleration of pumpkin is 0''
```
Rule 6: "Total force on a body is zero →
acceleration of the body is zero"
```
''total force on pumpkin is 0''
```
Rule 23iff: "The magnitude of every component of a vector is zero →
the magnitude of the vector is zero"
```
''total horizontal force on pumpkin is 0''
''total vertical force on pumpkin is 0''
```

Figure 10. Example of an inconsistent proof. One of the newly generated goals "Total horizontal force on pumpkin is 0" is inconsistent with the previously proven fact "Total vertical force on pumpkin is a nonzero constant."

which refers to the time the pumpkin is in free-fall, has been proven in another branch of the proof tree. In this case, the application of rule 23iff should not be allowed in the same proof because it results in the need to prove the contradictory statement "total vertical force on pumpkin is 0."

Another kind of inconsistency is related to metaknowledge reasoning, namely, the rules that have buggy counterparts. For example, if a correct rule (in the sense of a rule schema, e.g., p, q, and q^* have unbound variables)

$$p \to q$$

has a buggy counterpart

$$p \to q^*,$$

then both of them cannot be a part of theory T, which includes fact $\exists x \ p(x)$, provided we want to keep T consistent. The obvious workaround is to implement such pairs of rules as

$$bug^* \wedge p \to q$$

and

$$bug \wedge p \to q^*,$$

where *bug* and bug^* are mutually exclusive abducibles, because of constraint (11), that do not appear anywhere else.

In the actual implementation we handle this constraint on the metalevel by simply disallowing the appearance of pairs of these rules within the same proof.

We restate here that while the consistency constraints we describe are natural in theorem proving, from the viewpoint of student modeling they represent a risky assumption: that the student does not simultaneously hold inconsistent beliefs. There is, however, some justification of the assumption that the student doesn't hold directly contradicting beliefs, implemented as constraint (11), coming from cognitive science: it has been shown that even young children are unlikely to make mistakes in tasks involving taxonomic categories (Chi and Ceci, 1987).

4.6. SEARCH ISSUES

Our goal is to guarantee that the resultant proof will satisfy one of the following criteria:

- Strong criterion: there is no cheaper proof within the given threshold on depth of the proof. This is likely to require close to an exhaustive search.
- Medium criterion: there is no cheaper proof that is of the same depth or shorter, and the proof search has met the threshold on depth or the threshold on number of proofs. This criterion is likely to require exhaustion of the latter threshold by iterative deepening for a significantly large depth threshold. Therefore a deeper, cheaper proof within the depth threshold would not be found.
- Weak criterion: there is no cheaper proof that is of the same depth or shorter, and one of the thresholds (depth, number of proofs, satisfactory proof cost) is met by the proof search.

The cost threshold allows us to avoid iterative deepening and implement heuristics to help find a low-cost proof before we exhaust depth or number of proofs thresholds. Thus, our current search strives to satisfy the weak criterion. Heuristics are used to select the best potential proof for expansion, the best goals in the proof to address, and the best possible rule to apply to prove the goals. A description of the heuristics follows.

The best goals are those that have high assumption costs and have not been expanded for the most number of steps of the proof. Rules that would not satisfy the condition of consistency are eliminated from the list of potential candidates. Then the remaining pool of consistent rules is divided into classes of rules of the same cost. The costs of rules cause the search to try to apply first the most specific correct rules, then the more general correct rules, and, only when these two fail, the buggy rules. This approach reflects our subjective estimate of the probability of a successful proof for each choice of rule.

The cheapest rule is chosen, with possible conflicts being resolved nondeterministically. If no applicable rules are found in the given class, the next cheapest class is searched. If no class has an applicable rule, the goals are abduced, and their cost is added to the cost of the proof accordingly.

5. Evaluating the Tutoring System and the Theorem Prover

The Why2-Atlas system participated in an evaluation study in the spring of 2002 in order to acquire baseline measurements of student learning gains. The experimental setup focused on the question of whether similar content delivered through dialogue or a static text had different effects on student learning. Each condition selected the material it presented from a limited, well-defined set of prescribed physics topics for each training problem. The static text condition presented all of the prescribed topics, while the dialogue condition chose a subset of the topics that it deemed necessary given a student's essay responses and previous dialogue.

The population tested was undergraduate students who had recently completed an introductory physics course. Although the ideal population is physics students who are currently taking physics and who are just learning the content covered by the experiment, it is difficult to recruit enough students from such a highly constrained population to offset low experiment completion rates. But given that previous studies indicated that even physics students who have done well in their courses perform poorly on qualitative physics problems (Hake, 1998), we expected that students who had completed an introductory course to be appropriate as well for the experiment.

Why2-Atlas was one of several dialogue conditions for the experiment. Another dialogue condition of interest here was human tutoring dialogue. Although these human tutors and their students communicated through typing, the human tutoring condition was hypothesized to be better than the static text condition. We expected that the Why2-Atlas system would be at least as good as the static text condition given that the system and knowledge sources were known to be incomplete.

Although the students in every condition showed significant learning gains, a surprising result was that the gains for all of the conditions were statistically similar. Because the human tutoring condition and the static text condition were similar, and counter to previous experiments comparing human tutoring to simpler instruction, subsequent experiments focused on comparisons involving human tutors and the static text.

In addition, as we had expected, our system's accuracy for identifying misconceptions was poor. None of the misconceptions that were identified were justified according to human judgments of a sampling of the essays from students who completed the experiment for the Why2-Atlas system. We also confirmed that the propositional inputs to the theorem prover only partially represented the content of student essays, so the system evaluation provided no informative performance measurement of the theorem prover. Because the theorem prover was getting sparse representations of student's essays and because the system evaluation results are inconclusive, we cannot yet test for a correlation between student learning and the system's accuracy at selecting appropriate dialogue topics. We expect that, with system improvements and improvements in the experimental design, an upcoming repeat of the system evaluation will be more informative.

In the interim, we created a test suite using essays collected during the baseline evaluation and subsequent experiments. In addition to the essays from the baseline evaluation, we have since collected essays from students who have never taken physics but who receive a short instructive text prior to testing and training. To create the test suite, we randomly selected 45 essays, while balancing problems, subjects, subject backgrounds, and essay versions. The 45 essays cover seven problems and were written by 21 students (11 with physics backgrounds and 10 without); 32 of these essays are written by students who had not previously had a physics course, and the remainder by students who had completed a physics course. Since each student may have revised an essay for a problem multiple times, we randomly selected one problem essay per student. Once the essays for the test suite were selected, we hand-corrected the logged inputs for the various system modules for each student essay in the test suite and had human judges annotate each essay with the physics principles it covered and the misconceptions it exhibited.

Two main test suite evaluations are of interest for the abductive theorem prover relative to processing bounds and efficiency: (1) the accuracy of the misconceptions revealed by the proofs and (2) the accuracy of the whole proofs as student models. We have an evaluation of the accuracy of misconceptions relative to the test suite but have just begun a preliminary evaluation of the accuracy of the whole proofs generated.

5.1. ACCURACY OF THE MISCONCEPTIONS REVEALED

To assess the accuracy of the misconceptions that are identified as a result of the proofs produced by the theorem prover, we can compare the misconceptions selected with those that should have been identified according to the human judgments for the essays in the test suite.

Our goal here is to minimize the number of misconceptions missed by the system that a human judge identified as relevant. In the 45 essays of the test suite, three essays have two misconceptions each, eight essays have one misconception each, and the rest of the essays don't have any misconceptions from the list of 54 misconceptions that could arise for the training problems according to our physics experts.

To evaluate the accuracy of the theorem prover at revealing misconceptions, we compare the theorem prover's results for an essay relative to the misconceptions possible for the problem (system identified (SI) versus system did not identify (SDI)) to those of the human judgments annotated in the test suite relative to the misconceptions possible (human identified (HI) versus human did not identify (HDI)) to determine the number of

- true positives (TP), where TP = SI ∩ HI
- true negatives (TN), where TN = SDI ∩ HDI
- false positives (FP), where FP = SI ∩ HDI
- false negatives (FN), where FN = SDI ∩ HI

If for problem 1, experienced physics instructors indicate that 5 misconceptions are relevant, A–E, and for an essay instance on problem 1 the theorem prover output reveals misconceptions A and B while a human judge identified as present B and C instead, then for that essay instance, TP = 1, TN = 2, FP = 1, FN = 1. So TP + FN is the number of misconceptions that the human identified as present, and FP + TN is the number of relevant misconceptions that were not present according to the human judge.

We build a confusion matrix with the cells TP, FP, TN, FN by summing the values across each essay in the test suite. From this confusion matrix we can compute the following measures, which are frequently used in classification tasks for information retrieval and machine learning:

- recall (R) = TP/(TP + FN)
- precision (P) = TP/(TP + FP)
- positive false alarm rate = FP/(FP + TN)
- negative false alarm rate = FN/(FP + TN)

In addition, we also recorded the theorem prover's results at various proof cost thresholds to see how the performance changes as we move closer toward building a complete proof. For each threshold of interest, we create a separate confusion matrix. However, it is possible that other thresholds (for example, the threshold on number of possible proofs generated) are exceeded before a proof satisfying the cost threshold is found. When this case arises, we add the results of the best proof so far to the target threshold confusion matrix regardless of what the actual cost is.

As shown in Figure 11, the recall increases from 0 at a proof cost of 1 (where everything is assumed without proof) to 62% at a proof cost threshold of 0.2. As the recall increases, the precision degrades but then levels off. We expect that the precision will also improve rather than degrade once the planned improvements to the theorem prover are implemented (see Section 6). These results mean that the theorem prover can help to reveal up to 62% of the misconceptions that a human would recognize, but at the cost of identifying some misconceptions that are not justified by the essays. We consider recall to be the more important measure for misconceptions because it is important to find and address the misconceptions that are expected to be obvious to a human tutor.

In order to get a sense of how difficult the task of finding misconceptions is, it is useful to also examine the false alarm rates, as shown in Figure 12. The negative false alarm rate is inversely related to recall in that as recall increases, the negative false alarm rate declines and indicates how many misconceptions are overlooked. Our goal is for this number to fall as close to 0 as possible because we hypothesize that overlooking misconceptions is detrimental in tutorial applications. We have observed that students can have a complete explanation and still conclude the wrong answer from that explanation.

The positive false alarm rate is inversely related to precision and indicates how many misconceptions the system incorrectly attributed to essays. While we'd prefer that this number fall to zero as well, it is not so bad to cover more misconceptions

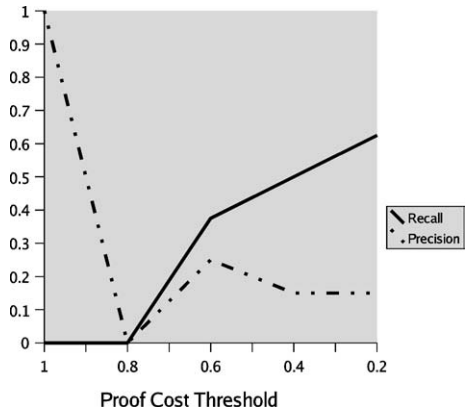

Figure 11. Recall and precision measures as proof cost threshold decreases.

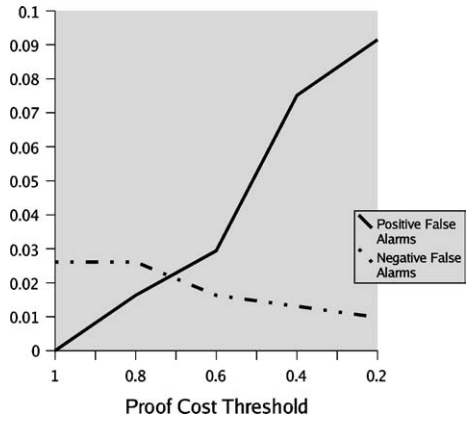

Figure 12. Negative and positive false alarm rates as proof cost threshold decreases.

than are needed. One might take the approach, as with the static text condition, of covering all the misconceptions that are expected to be possible for a problem, but some hypothesized downsides of this approach are inadvertently strengthening the reasoning associated with a misconception and a loss of interest and cooperation on the part of the student; the student's effort to explain during subsequent problems may drop off if the student perceives the system is not usually giving appropriately focused feedback on their essay.

As expected, the effort to find more complete proofs and improvements in recall and negative false alarm rates require an increase in processing time, as shown by Figure 13.

While the theorem prover's negative false alarm rate is considered good, we expect that additional testing and fine-tuning of the rules, inference procedures, and proof search heuristics will further improve the results.

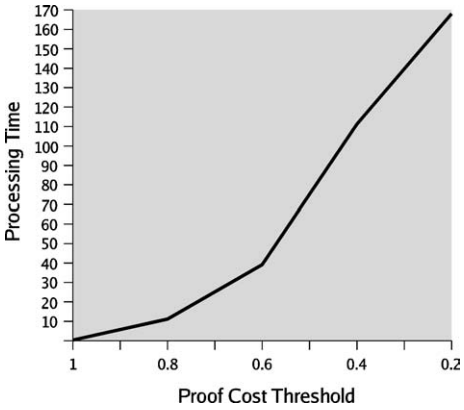

Figure 13. Processing time in seconds as proof cost threshold decreases.

5.2. ACCURACY OF THE WHOLE PROOFS GENERATED

Comparing the misconceptions revealed with those that a human judge identifies is only a coarse measure of the accuracy of the proof generated. To determine the fitness of the theorem prover for assessing completeness of an explanation, we must also consider the accuracy of the whole proofs generated. Assessing proof accuracy is more difficult because the proofs must be hand verified. In addition, it is difficult to create a reliable gold standard against which to evaluate the accuracy of proofs for essays and the reasons for any inaccuracy. The reason is that, in general, language in context gives rise to many inferences (Austin, 1962; Searle, 1975). In this case, we will judge whether the proof is at least a plausible model for the student essay. Such an accuracy evaluation is still in progress.

6. Future Work

A number of improvements to Tacitus-lite+ and to Why2-Atlas are in progress, and we also plan to address a number of improvements in the future. The improvements we have identified for Tacitus-lite+ are as follows:

- To integrate into the tutoring system a refinement where the factoring operation (as defined in Section 4.3) distinguishes between factoring with more specific and with more general atoms (and charges the proof an appropriate cost).
- To respond to explicit conflicts in students' essays in the near future and to work on interactive proof generation (i.e., asking questions of the student when ambiguities arise, rather than dealing with them after the proof generation is complete).
- To explore a stronger consistency criterion than (11) in order to improve confidence in the consistency of an abductive explanation. On the other hand, a consistent proof may not be appropriate as a model for all types of students.

This trade-off can be accounted for by allowing a certain degree of flexibility in the consistency criterion. The relationship between the consistency of the explanation and its measures of utility and plausibility is not straightforward and is currently being investigated.

– To address weaknesses in the current reasoning system that stem from a lack of explicit negation, quantifiers and disjunction in its knowledge representation. For example, it could be beneficial for the sort hierarchy to distinguish between lexical negation (*decreasing*) and classical negation ($\neg$*increasing*), as proposed in (Kaneiwa and Tojo, 2001).

Naturally, addition of any of these features will require more sophisticated reasoning procedures.

In the area of improvements to Why2-Atlas, we are currently extending the system to cover a larger subset of physics and as a result more physics problems. Our next most immediate goal is to improve the feedback to students relative to the proofs produced by Tacitus-lite+. We need to address two situations. The first is when the system generates multiple lowest-cost proofs, as in the earlier example associated with Figures 6 and 7. The second is when it produces just one lowest-cost proof in which the student has presented unambiguous correct information in the essay that leads the system to infer a correct and complete explanation or the student has provided information that is clearly incorrect.

Although work on natural language interpretation selection (Zukerman and George, 2002) and natural language generation work on argumentation and explanation (e.g., Zukerman et al., 2000; Horacek, 1997; Zukerman and McConachy, 1993) will give us a starting point for deriving feedback to the student relative to a proof, there are a number of new challenges to address. First, generation work on argumentation and explanation has focused on generating text as opposed to dialogue and deals mainly with correct representations. Second, in the case of choosing between alternative representations, the work in (Zukerman and George, 2002) doesn't address the problem of choosing between representations that have the same number of inferences, as is the case for the alternative proofs in Figures 6 and 7.

In the situation where there are multiple lowest-cost proofs, the system needs to locate major differences between the alternative proofs and ask clarification questions that will help the system determine which proof is the best representation of the student's beliefs. For instance, in the earlier example with Figures 6 and 7, there are a number of possibilities for doing this. First, the system can ask, "Why do you think the pumpkin slows down?" If the student responds with "I don't know," then the system can ask a more leading question such as "What is the total horizontal force on the pumpkin?" since a proposition about total force is present in both proofs but has different argument values.

The other situation is where the system finds one plausible proof either on its own or because it asked questions that allowed it to narrow down the possibilities to one proof as above. We are working on the case where the system has to infer

T: How do you know that the pumpkin will land on the man?

U: I don't know

T: What is the pumpkin's displacement relative to the man's?

U: it is equal to the man's

T: Right. The pumpkin's horizontal displacement will be equal to the man's displacement at all times. So what leads you to know this, given you already know that the man is carrying the pumpkin and is running at a constant speed?

U: I don't have any idea

T: What can you say about the velocity of the pumpkin before the runner throws it?

U: it is equal to the man's

T: Right. The horizontal velocity of the pumpkin is equal to the horizontal velocity of the runner. So what follows from knowing this ?

Figure 14. A prototype dialogue based on a simplified input proof in which the proposition "The pumpkin lands on the man" is marked as covered by the user's essay.

most of this one proof from just a few true student propositions (Jordan, 2004). Our approach is to use low-level constraints defined in terms of the structural relationship between a proof step of high assumption cost that is to be elicited and a proof step of low assumption cost. The structural relationships are ancestor, sibling, and transitive. Next we define higher-level elicitation strategies by specifying constraints on the distance between the steps in the three structural relationships, the orientation between the high-cost target step and the low-cost step, and the distance and orientation when choosing the next high-cost target step to elicit. In the case of an ancestor relationship, in one dialogue turn, the constraints select a step of low assumption cost that is at a distance N on a path from the step that is to be elicited and ask for an elaboration relative to the low-cost step. For example, if the student had said, "The pumpkin lands on me because the velocity of the pumpkin is constant" and N is 1, the system can ask either an open-ended question such as "What follows from knowing that the pumpkin's velocity is constant?" or a more focused question such as "What does the constant velocity tell us about the pumpkin's acceleration?"

We have implemented a proof of concept prototype using these three low-level constraints and additional higher-level dialogue strategy constraints. The prototype takes a simplified proof as input, where the proof is marked with which steps were covered in an essay and which are givens and therefore of low assumption cost. An excerpt of a dialogue produced by this prototype, where the initial essay is only "The pumpkin lands on the man," is shown in Figure 14.

We are just beginning to explore the case where the one plausible proof contains a bug because the student made some incorrect statements (as in the essay in Figure 1). Our approach treats these incorrect student statements as being correct and attempts to lead the student to a contradiction (Jordan, 2004), as with reductio ad absurdum (Zukerman et al., 2000). For example, if the target incorrect statement is "There was a horizontal force acting on the pumpkin before the throw," then a con-

tradiction is sought for the givens "The horizontal velocity of the man is constant before the throw" and "The man is carrying the pumpkin before the throw."

7. Conclusions

In this paper we have presented weighted abductive proofs as a method for developing a deeper understanding of students' explanations for qualitative physics problems than can be afforded by superficial sentence-level semantics. We viewed abductive proofs that are based on student essays as a way to model students' beliefs and knowledge of physics. We described how feedback that is adapted to a student's particular needs can be generated based on these student models. To show how we are able to acquire these student models, we presented a qualitative physics ontology with sorts and a collection of correct and buggy rules that were designed to cover a subset of Newtonian mechanics and the most common misconceptions. We also described how we adapted a weighted-abduction reasoning framework for the task of building proofs of student essays. A combination of heuristics was developed to assist in choosing the best proof and hence the best model of the student by having these heuristics approximate selection criteria that are based on measures of utility and plausibility of a candidate model.

Acknowledgments

This research was supported by MURI grant N00014-00-1-0600 from ONR Cognitive Science and by NSF grant 9720359. We thank the entire NLT team for their many contributions in creating and building the Why2-Atlas system. In particular, we thank Michael Ringenberg and Roy Wilson for their work with Tacitus-lite+, and Uma Pappuswamy and Michael Böttner for their work with the ontology and the domain rules.

References

Aleven, V. and Koedinger, K. R. (2000) The need for tutorial dialog to support self-explanation, in *Building Dialogue System for Tutorial Applications, Papers of the 2000 AAAI Fall Symposium.*

Aleven, V., Popescu, O. and Koedinger, K. (2002) Pilot-testing a tutorial dialogue system that supports self-explanation, in *Proceedings of Intelligent Tutoring Systems Conference*, LNCS 2363, Springer, pp. 344–354.

Aleven, V., Popescu, O. and Koedinger, K. R. (2001a) Toward tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor, in *Proceedings of 10th International Conference on Artificial Intelligence in Education (AI-ED 2001)*, IOS Press, Amsterdam, pp. 246–255.

Aleven, V., Popescu, O. and Koedinger, K. R. (2001b) A tutorial dialogue system with knowledge-based understanding and classification of student explanations, in *Working Notes of 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Appelt, D. and Pollack, M. (1992) Weighted abduction for plan ascription, *User Modeling and User-Adapted Interaction* **2**(1–2), 1–25.

Austin, J. L. (1962) *How to Do Things with Words*, Oxford University Press, Oxford.
Bacchus, F., Tennenberg, J. and Koomen, J. (1989) A non-reified temporal logic, in J. F. Allen, R. Fikes and E. Sandewall (eds.), *KR'89: Principles of Knowledge Representation and Reasoning*, San Mateo, California, Morgan Kaufmann, pp. 2–10.
Bylander, T., Allemang, D., Tanner, M. C. and Josephson, J. R. (1991) The computational complexity of abduction, *Artificial Intelligence* **49**(1–3), 25–60.
Charniak, E. (1986) A neat theory of marker passing, in *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI'86)*, pp. 584–588.
Charniak, E. and Shimony, S. E. (1990) Probabilistic semantics for cost-based abduction, in *Proceedings of AAAI-90*, pp. 106–111.
Charniak, E. and Shimony, S. E. (1994) Cost-based abduction and MAP explanation, *Artificial Intelligence* **66**, 345–374.
Chi, M. T. H. and Ceci, S. J. (1987) Content knowledge: Its role, representation and restructuring in memory development, *Advances in Child Development and Behavior* **20**, 91–142.
Chi, M. T. H., de Leeuw, N., Chiu, M.-H. and LaVancher, C. (1994) Eliciting self-explanations improves understanding, *Cognitive Science* **18**, 439–477.
Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T. and Hausmann, R. G. (2001) Learning from human tutoring, *Cognitive Science* **25**(4), 471–533.
Cohn, A. G. (1989) Taxonomic reasoning with many-sorted logics, *Artificial Intelligence* **3**, 89–128.
Conati, C., Gertner, A. and VanLehn, K. (2002) Using Bayesian networks to manage uncertainty in student modeling, *J. User Modeling and User-Adapted Interaction* **12**(4).
de Kleer, J. (1990) Multiple representations of knowledge in a mechanics problem-solver, in D. S. Weld and J. de Kleer (eds.), *Readings in Qualitative Reasoning about Physical Systems*, San Mateo, California, Morgan Kaufmann, pp. 40–45.
Eiter, T. and Gottlob, G. (1993) The complexity of logic-based abduction, in *Symposium on Theoretical Aspects of Computer Science*, pp. 70–79.
Eshghi, K. (1993) A tractable class of abduction problems, in *Proceedings 13th International Joint Conference on Artificial Intelligence*, Chambery, France, pp. 3–8.
Eshghi, K. and Kowalski, R. A. (1989) Abduction compared with negation by failure, in *Proceedings of the 6th International Conference on Logic Programming (ICLP '89)*, pp. 234–254.
Forbus, K., Carney, K., Harris, R. and Sherin, B. (2001) A qualitative modeling environment for middle-school students: A progress report, in *QR-01*.
Forbus, K. D. (1997) Using qualitative physics to create articulate educational software, *IEEE Expert*, pp. 32–41.
Frisch, A. M. (1991) The substitutional framework for sorted deduction: Fundamental results on hybrid reasoning, *Artificial Intelligence* **49**(1–3), 161–198.
Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N. and the TRG (2000) Using latent semantic analysis to evaluate the contributions of students in AutoTutor, *Interactive Learning Environments* **8**, 129–148.
Hake, R. R. (1998) Interactive-engagement versus traditional methods: A six-thousand student survey of mechanics test data for introductory physics students, *American Journal of Physics* **66**(4), 64–74.
Haugh, B. (1987) Non-standard semantics for the method of temporal arguments, in *Proc. of IJCAI'87*, pp. 449–454.
Hestenes, D., Wells, M. and Swackhamer, G. (1992) Force concept inventory, *The Physics Teacher* **30**, 141–158.
Hewitt, P. G. (1998) *Conceptual Physics*, 8th edn, Addison-Wesley.
Hobbs, J., Stickel, M., Appelt, D. and Martin, P. (1993) Interpretation as abduction, *Artificial Intelligence* **63**(1–2), 69–142.
Hobbs, J., Stickel, M., Martin, P. and Edwards, D. (1988) Interpretation as abduction, in *Proc. 26th Annual Meeting of the ACL, Association of Computational Linguistics*, pp. 95–103.

Horacek, H. (1997) A model for adapting explanations to users' likely inferences, *User Modeling and User-Adapted Interaction* **7**(1), 1–55.

Jordan, P. and VanLehn, K. (2002) Discourse processing for explanatory essays in tutorial applications, in *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*.

Jordan, P. W. (2004) Using student explanations as models for adapting tutorial dialogues, in *Proceedings of 17th International FLAIRS Conference*.

Kakas, A., Kowalski, R. A. and Toni, F. (1998) The role of abduction in logic programming, in D. M. Gabbay, C. J. Hogger and J. A. Robinson (eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 5, Oxford University Press, pp. 235–324.

Kaneiwa, K. and Tojo, S. (2001) An order-sorted resolution with implicitly negative sorts, in *Proceedings of the 2001 International Conference on Logic Programming (ICLP'01)*, LNCS 2237, Springer, pp. 300–314.

Keeney, R. and Raiffa, H. (1976) *Decisions with Multiple Objectives*, Wiley.

Landauer, T. K., Foltz, P. W. and Laham, D. (1998) An introduction to latent semantic analysis, *Discourse Processes* **25**, 259–284.

Lascarides, A. and Asher, N. (1991) Discourse relations and defeasible knowledge, in *29th Annual Meeting of the Association for Computational Linguistics*, pp. 55–62.

Lavoie, B. and Rambow, O. (1997) A fast and portable realizer for text generation systems, in *Proceedings of the Fifth Conference on Applied Natural Language Processing Chapter of the Association for Computational Linguistics*, Washington, DC, pp. 265–268.

Leake, D. (1995) Abduction, experience, and goals: A model of everyday abductive explanation, *J. Experimental and Theoretical Artificial Intelligence* **7**, 407–428.

McCallum, A. and Nigam, K. (1998) A comparison of event models for naive Bayes text classification, in *Proceedings of AAAI/ICML-98 Workshop on Learning for Text Categorization*, AAAI Press.

McRoy, S. and Hirst, G. (1995) The repair of speech act misunderstandings by abductive inference, *Computational Linguistics* **21**(4), 435–478.

Murray, R. C. and VanLehn, K. (2000) DT tutor: A dynamic decision-theoretic approach for optimal selection of tutorial actions, in *Proceedings of Intelligent Tutoring Systems Conference*, LNCS 1839, Springer, pp. 153–162.

Ng, V. and Cardie, C. (2002) Improving machine learning approaches to coreference resolution, in *Proceedings of Association for Computational Linguistics 2002*.

Paul, G. (1993) Approaches to abductive reasoning – An overview, *Artificial Intelligence Review* **7**(2), 109–152.

Ploetzner, R. and VanLehn, K. (1997) The acquisition of qualitative physics knowledge during textbook-based physics training, *Cognition and Instruction* **15**(2), 169–205.

Poole, D. (1993) Probabilistic Horn abduction and Bayesian networks, *Artificial Intelligence* **64**(1), 81–129.

Rayner, M. and Alshawi, H. (1992) Deriving database queries from logical forms by abductive definition expansion, in *Proceedings of the Third Conference of Applied Natural Language Processing*, Trento, Italy, pp. 1–8.

Rosé, C., Bhembe, D., Roque, A., Siler, S., Srivastava, R. and VanLehn, K. (2002) A hybrid understanding approach for robust selection of tutoring goals, in *Proceedings of Intelligent Tutoring Systems Conference*, LNCS 2363, Springer, pp. 552–561.

Rosé, C., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K. and Weinstein, A. (2001) Interactive conceptual tutoring in Atlas-Andes, in *Proceedings of AI in Education 2001 Conference*.

Rosé, C., Roque, A., Bhembe, D. and VanLehn, K. (2002) An efficient incremental architecture for robust interpretation, in *Proceedings of Human Language Technology Conference*, San Diego, CA.

Santos, Jr., E. and Santos, E. S. (1996) Polynomial solvability of cost-based abduction, *Artificial Intelligence* **86**(1), 157–170.

Schmidt-Schauß, M. (1989) *Computational Aspects of an Order-Sorted Logic with Term Declarations*, Springer.

Searle, J. R. (1975) Indirect speech acts, in P. Cole and J. Morgan (eds.), *Syntax and Semantics 3: Speech Acts*, Academic Press. Reprinted in S. Davis (ed.), *Pragmatics. A Reader*, Oxford University Press, 1991.

Selman, B. and Levesque, H. J. (1990) Abductive and default reasoning: A computational core, in *Proceedings of AAAI-90*, Boston, MA, pp. 343–348.

Slotta, J. D., Chi, M. T. and Joram, E. (1995) Assessing students' misclassifications of physics concepts: An ontological basis for conceptual change, *Cognition and Instruction* **13**(3), 373–400.

Stickel, M. (1988) 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.

Strube, M., Rapp, S. and Müller, C. (2002) The influence of minimum edit distance on reference resolution, in *Proceedings of Empirical Methods in Natural Language Processing Conference*.

Thomason, R. H., Hobbs, J. and Moore, J. D. (1996) Communicative goals, in K. Jokinen, M. Maybury, M. Zock and I. Zukerman (eds.), *Proceedings of the ECAI 96 Workshop Gaps and Bridges: New Directions in Planning and Natural Language Generation*.

VanLehn, K., Jordan, P., Rosé, C., Bhembe, D., Böttner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S. and Srivastava, R. (2002) The architecture of Why2-Atlas: A coach for qualitative physics essay writing, in *Proceedings of Intelligent Tutoring Systems Conference*, LNCS 2363, Springer, pp. 158–167.

Wahlster, W. and Kobsa, A. (1989) User models in dialogue systems, in A. Kobsa and W. Wahlster (eds.), *User Models in Dialogue Systems*, Springer Verlag, Berlin, pp. 4–34.

Walther, C. (1987) *A Many-Sorted Calculus Based on Resolution and Paramodulation*, Morgan Kaufmann, Los Altos, California.

Weld, D. S. and de Kleer, J. (eds.) (1990) *Readings in Qualitative Reasoning about Physical Systems*, Morgan Kaufmann, San Mateo, California.

Zukerman, I. and Albrecht, D. W. (2001) Predictive statistical models for user modeling, *User Modeling and User-Adpated Interaction* **11**(1–2), 5–18.

Zukerman, I. and George, S. (2002) A minimum message length approach for argument interpretation, in *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*.

Zukerman, I. and McConachy, R. (1993) Generating concise discourse that addresses a user's inferences, in *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, Inc., pp. 1202–1207.

Zukerman, I., McConachy, R. and Korb, K. B. (2000) Using argumentation strategies in automated argument generation, in *Proceedings of the 1st International Natural Language Generation Conference*, pp. 55–62.