LEARNING IN MATHEMATICALLY-BASED DOMAINS: UNDERSTANDING AND GENERALIZING OBSTACLE CANCELLATIONS

by

Jude W. Shavlik and Gerald F. DeJong

Computer Sciences Technical Report #837

April 1989

Learning in Mathematically-Based Domains:
Understanding and Generalizing Obstacle Cancellations

Jude W. Shavlik

Computer Sciences Department
University of Wisconsin
Madison, WI 53706
shavlik@cs.wisc.edu

Gerald F. DeJong

Department of Computer Science
University of Illinois
Urbana, IL 61801

(Submitted to *Artificial Intelligence*)

ABSTRACT

Mathematical reasoning provides the basis for problem solving and learning in many complex domains. A model for applying *explanation-based learning* in mathematically-based domains is presented, and an implemented learning system is described. In explanation-based learning, a specific problem's solution is generalized into a form that can be later used to solve conceptually similar problems. The presented system's mathematical reasoning processes are guided by the manner in which variables are cancelled in specific problem solutions. Analyzing the cancellation of *obstacles* - variables that preclude the direct evaluation of the problem's unknown - leads to the generalization of the specific solution. Two important general issues in explanation-based learning are also addressed. Namely, generalizing the number of entities in a situation and acquiring efficiently-applicable concepts.

1. INTRODUCTION

> All the mathematical sciences are founded on relations between physical laws and laws of numbers, so that the aim of exact science is to reduce the problems of nature to the determination of quantities by operations with numbers.
>
> James Clerk Maxwell

Mathematically-based domains present several unique challenges and opportunities for machine learning. Many scientific and technical domains — physics, electronics, chemistry, and economics, for example — share the common formalism of mathematics, and much of the reasoning in these fields involves understanding the constraints inherent in mathematical descriptions. Mathematical models of real-world situations are constructed, and these mathematical abstractions are used to predict behavior in the domain being modelled. Hence, solving quantitative problems in these domains requires a competence in symbolic mathematical manipulation. Furthermore, since mathematics is the underlying formal language, many important domain concepts can be adequately captured only through a mathematical specification. Thus, concept learning in these domains is also rooted in mathematics.

The research reported here focusses on learning new concepts from examples in mathematically-oriented domains, using the *explanation-based learning* paradigm [7, 20]. A computer system, **Physics 101**, has been implemented that embodies the theories of learning and problem solving developed. The system is fully implemented and its performance is offered as an illustration of a number of theoretical points. It is intended to model the acquisition of concepts taught in a first-semester college physics course [30, 31, 32] — hence, the name **Physics 101**. However, the focus of the **Physics 101** implementation is not physics *per se*. Very little knowledge about physics is in the system. In fact, all of its initial physics knowledge is contained in a half-dozen or so formulae, listed later in this section, which capture Newton's laws and a few definitions. None of the system's algorithms utilize knowledge about physics, except that which is captured in these initial physics formulae. Rather, the focus is on mathematical reasoning, and the domain of physics is used as a testbed, since it is an elegant domain that stresses the use of complicated mathematics. The mathematical competence built into **Physics 101** is approximately at the level of someone who has completed one semester of calculus.

Physics 101 improves its own problem solving ability by analyzing its teacher's solutions to specific physics problems, and then generalizing the solution. The generalization process is driven by an *explanation* of why the particular solution worked for the particular problem. Knowledge about the domain allows the explanation to be developed, and then generalized. The example problems, as well as a broad class of conceptually similar problems, can be solved autonomously by **Physics 101** using the new schemata.

Mathematically-based domains are an area where the strengths of explanation-based learning are particularly appropriate, because explanation-based learning supports the construction of large concepts by analyzing how smaller concepts can be pieced together to solve a specific problem. Combining small concepts to form larger ones is the basis of progress in mathematical domains.

There are three significant results of this research. The first is the notion of *obstacles* and their *cancellation*. Analyzing the cancellation of these obstacles focusses the system's attention and guides learning. The second is a solution to the problem of *generalizing the structure of explanations* [6, 27, 33]. It is claimed that an explanation-based learning system must be able to reformulate its explanations if it is to learn concepts involving an arbitrary number of objects or actions [35]. Third, an approach to the *operationality/generality* problem [7, 12, 20, 29, 32] in explanation-based learning is presented. The notion of a *special case* is advanced to achieve operationality without sacrificing any generality of the concept.

2. THE LEARNING MODEL

Figure 1 illustrates the learning model reflected in **Physics 101**. After a physical situation is described and a problem is posed[1], the an attempt to solve the problem is made. Focus in this research is on the process of learning during a successful solution; particularly on learning from a teacher's example. When the learner cannot solve a problem, a solution from the instructor is requested. The solution provided must then be verified; additional details are requested when steps in the teacher's solution cannot be understood. The process by which an example is understood is divided into two phases. First, using current knowledge about mathematics and the task domain, the student verifies that the solution is valid. At the end of this phase the student knows that the instructor's solution solves the current problem, but does not have any detailed understanding of *why* the teacher chose these steps to solve the problem. During the second phase of understanding, the student determines a reason for the structure of each expression in the teacher's solution. Especially important is understanding new formulae encountered in the solution. After this phase the student has a firm understanding of how and why this solution solved the problem at hand. In the third and final phase the student is able to profitably generalize any new principles that were used in the solution process, thereby increasing knowledge of the task domain.

The remainder of this section provides additional introductory details about the **Physics 101** system and elaborates the research issues addressed.

2.1. Some Terminology

Before the concept of an obstacle can be motivated it is necessary to introduce some terminology. The following definitions are used in **Physics 101**:

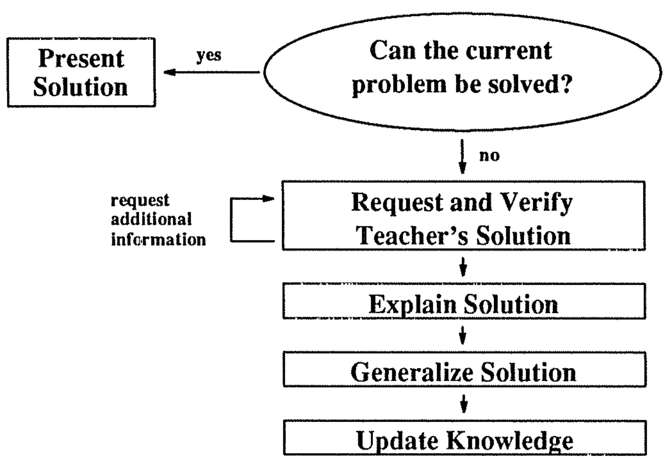

Figure 1. An Overview of the Learning Model

---

[1] Problems are expressed in the language of mathematics, not in ordinary English. The important problem of translating "word problems" into mathematical descriptions is not addressed. However, it has been addressed in the work of others [4, 24, 26].

(1) A *problem* is a set of variables, with specifications of some of the properties of these variables, together with a mathematical expression (called the *unknown*) for which the value of some property is desired. (A single, isolated variable is the simplest form of an unknown.) The properties of expressions can include such things as their value at specific points or their dependence on some other variable, such as time.

(2) An *unacceptable* variable is one whose value for the desired property of the unknown is also not known. For example, if it is desired to know the value of the variable X at time t, and the value of Y at time t is not known, then Y is an unacceptable variable in this problem.

(3) An *equation schema* is an inference rule which specifies a conjunction of antecedents (the preconditions) and a consequent (an equation). The equation can be used to rewrite terms in some expression. For example, the precondition of the equation $\frac{x}{x} = 1$ is that x be non-zero.

(4) *Problem-solving schemata* describe strategies used to solve problems. They guide the problem solver in its task of applying equation schemata in order to solve the problem.

(5) *Background knowledge* consists of equation schemata describing general mathematical laws (e.g., equations of algebra and calculus), equation schemata describing properties of the domain at hand (e.g., $force = mass \times acceleration$), problem-solving schemata, and miscellaneous inference rules. The domain-specific equations specify the variables known to the system (e.g. force, mass, and acceleration) that are not mentioned in the problem description.

(6) A *primary obstacle set* for a problem is a group of unacceptable variables which, if the values of their specified properties were known, would permit the system to solve the problem (that is, directly determine the value of the unknown's property). Section 3.4 provides a fuller definition of obstacles.

2.2. Obstacles and Their Cancellation

Reasoning about obstacles plays a major role in the **Physics 101** system. For an intuitive grasp of the technique, consider the following elementary physics problem:

> A massless point particle of charge 16 *esu* experiences an attraction to a charged sphere whose radius is 3 *cm*. The particle is 7 *cm* from the surface of the sphere. The sphere's charge is uniformly distributed about its surface with a density of 0.2 esu / cm^2. What is the force experienced by the particle?

Assume the background knowledge includes the formula for the electrostatic force between point charges, formulae for the volume and surface areas of a sphere, superposition of forces, etc. There are two ways to work this problem. The first is to grind through the equations, performing a surface integral over the sphere of the force contribution of each differential area. The second is to compute the total charge on the sphere and treat it as an equivalent point charge located at the sphere's center.

The second solution is much simpler, but it is not immediately apparent from the background knowledge that it is a valid solution. Somewhat sophisticated mathematical reasoning is needed to confirm this approach. Basically, it works for two reasons. First, due to the symmetry of the situation, the net force on the particle lies on the line connecting the particle and the center of the charged sphere. Forces in all other directions are cancelled. Second, the particle is attracted to the near portion of the sphere just as much as the far portion. This is a bit counter-intuitive. Even though the front of the sphere is closer to the particle, any differential solid angle cone from the particle through the sphere intercepts a slightly larger area on the far side than the near side. The larger size of the rear differential exactly balances the closeness advantage of the front differential.

If a student is shown the second, easier solution, and if he can convince himself that the simplifying transformation is correct, then he can learn an important new problem-solving technique. His performance will be improved for a class of conceptually similar, but possibly superficially very different, problems.

The notion of obstacle is related to that of *impasse* [3]. The difference is that obstacles are inherent in problem descriptions while impasses arise during the process of problem solving. Additional reasoning may lead to a way to resolve the impasse, and if so, problem solving can continue. Obstacles may lead to impasses if the problem solver does not know how to surmount them, but this is not necessary. Obstacles continue to exist in completed solutions and provide a focus with which the steps in a solution can be explained.

The notion of obstacles is central to determining the range of applicability of the new concept. The concept's generality is dictated by the generality of the inference rules participating in obstacle cancellation. Any problem which allows the same obstacle cancellation treatment will be covered by the new concept. The above trick works, for example, in a problem involving a gravitational force instead of electrostatic force - both are inverse square forces, but it would not work if the particle were *inside* the sphere as the obstacle cancellation structure is no longer supported.

2.3. Generalizing the Structure of Explanations

Understanding obstacle elimination provides the focus by which **Physics 101** explains solutions. Explanations that suffice for understanding a specific example are not always satisfactory for generalizing the example [35]. Often, the explanation *structure* must be generalized if a useful concept is to be produced.

Generalizing the structure of an explanation can involve generalizing the number of times a technique is employed, the number of objects involved, the order techniques are applied [22], or the actual techniques used. This article largely focusses on the topic of *generalizing number* or *generalizing to N* [34].

The need for generalizing number in EBL can be seen by considering the **LEAP** system [19]. This system is shown an example of using *NOR* gates to compute the boolean *AND* of two *OR*'s. It discovers that the technique generalizes to computing the boolean *AND* of any two inverted boolean functions. However, **LEAP** cannot generalize this technique to allow constructing the *AND* of an arbitrary number of inverted boolean functions using a multi-input *NOR* gate. This is the case even if **LEAP**'s initial background knowledge were to include the general version of DeMorgan's Law and the concept of multi-input *NOR* gates.

Ellman's system [8] also illustrates the need for generalizing number. From an example of a four-bit circular shift register, his system constructs a generalized design for an arbitrary four-bit permutation register. A design for an N-bit circular shift register cannot be produced. As Ellman points out, such generalization, though desirable, cannot be done using the standard EBL technique of changing constants to variables.

Many important concepts, in order to be properly learned, require generalization of number. For example, physical laws such as momentum and energy conservation apply to arbitrary numbers of objects, constructing towers of blocks requires an arbitrary number of repeated stacking actions, and setting a table involves a range of possible numbers of guests.[2] In addition, there is recent psychological evidence [1] that people can generalize number in an explanation-based fashion.

[2] These concepts are among those acquired by the systems described in [35].

Repetition of an action or object is not a sufficient condition for generalization to N to be appropriate. Compare two simple examples. Generalizing to N is necessary in one but inappropriate in the other. The examples are:

- observing a previously unknown method of moving an obstructed block, and
- seeing, for the first time, a toy wagon being built.

The initial states of the two problems appear in figure 2. Suppose a learning system observes someone achieving the desired states. In each case, consider what general concept should be acquired.

In the first example, the person wishes to move, using a robot manipulator, a block which has four other blocks stacked in a tower on top of it. The manipulator can pick up only one block at a time. The demonstrated solution is to move all four of the blocks in turn to some other location. After the underlying block has been cleared, it is moved. In the second example, the person wishes to construct a movable rectangular platform, one that is stable while supporting any load whose center of mass is over the platform. Given the platform and a bin containing two axles and four wheels, the solution is to first attach each of the axles to the platform. Next all four of the wheels are grabbed in turn and mounted on an axle protrusion.

This comparison illustrates an important problem in explanation-based learning. Number generalization should only occur when justified. Generalizing the block unstacking example should produce a plan for unstacking *any* number of obstructing blocks, not just four as observed. The wagon-building example, however, should not generalize the number "4." It makes no difference whether the system is given a bin of five, six, or 100 wheels, because only four wheels are needed to fulfill the functional requirements of a stable wagon.

Standard explanation-based learning algorithms (e.g., [20, 21]) and similar algorithms for chunking [13] cannot treat these cases differently. These algorithms, possibly after pruning the explanation to eliminate irrelevant parts, replace constants with constrained variables. They cannot significantly augment the explanation during generalization. Thus, the *building-a-wagon* type of concept will appear to be correct but the *unstacking-to-move* concept will be undergeneralized. The acquired schema will have generalized the identity of the blocks so that the target block need not be occluded by the same four blocks as in the example. Any four obstructing blocks can be unstacked. However, there must be exactly four blocks.[3] Unstacking five or more blocks is beyond the scope of the acquired concept.

Figure 2. Initial States for Two Sample Problems

[3] The SOAR system [13] would seem to acquire a number of concepts which together are slightly more general. As well as a new operator for moving four blocks, the system would acquire new operators for moving three blocks, two blocks, and one block, but not for five or more.

Note that EBL systems do not work correctly on the *building-a-wagon* kind of problems either — they just get lucky. They do nothing to augment explanation structures during generalization. It just happens that to acquire a schema to build a wagon, *not* generalizing the explanation structure is the appropriate thing to do.

One could simply define the scope of EBL systems to exclude the *unstacking-to-move* concept and those like it. This would be a mistake. First, the problem of augmenting the explanation during generalization, once seen, is ubiquitous. It is manifested in one form or another in many real-world domains. Second, if one simply defines the problem away, the resulting system could never guarantee that any of its concepts were as general as they should be. Even when such a system correctly constructed a concept like the *building-a-wagon* schema, it could not know that it had generalized properly. The system could not itself tell which concepts fall within its scope and which do not. The analysis of obstacle cancellation provides the motivation by much **Physics 101** properly generalizes number.

2.4. Operationality, Generality, and Special-Case Schemata

Acquiring the definition of a new general concept is only half of the problem faced by a machine learning system. To usefully contribute to improved problem solving, the system must be able to later apply the new concept effectively [12, 29, 32]. That is, the concept must be *operational* [23].

One can merge aspects of context with a general schema to yield a more specialized version. This more specialized schema is more operational. Less effort need be expended in satisfying its precondions; however, it applies in fewer contexts. For example, consider the general schema *eat-at-restaurant.* It applies to many restaurant types including fancy French restaurants, smorgasbord buffets, and pizzerias. Applying the schema in a particular context requires a substantial number of consistent decisions, and, therefore, a considerable amount of computation. Compare this to a more-narrowly applicable schema, *eat-at-fast-food-restaurant.* In it many decisions concerning cost, duration of meal, menu items, et cetera have already been made. Thus, the specialized schema is more operational, requiring less computation than the more general version.

In learning a new schema, it would seem that the system must weigh operationality against generality. However, this need not be the case. By acquiring several schemata, operationality can be preserved without sacrificing generality. In addition to the general schema, specializations of the concept, called *special-case schemata*, are formed and stored. Later in this article, a method for forming special cases, which guarantees improved operationality and yields schemata with potentially high utility, is presented.

2.5. Initial Knowledge of the System

Physics 101 possesses a large number of mathematical problem-solving techniques. For example, it can symbolically integrate expressions, cancel variables, perform arithmetic, and replace terms by utilizing known formulae. Figure 3 contains a portion of the system's hierarchy of calculation techniques. A calculation step may either rearrange the entities in the current expression, simplify the current expression, or replace terms in the expression by substituting formulae. Rearrangement involves such things as moving constants into and out of integrals and derivatives. Simplification involves algebraic cancellation, numerical calculation, and the solving of calculus. The formulae that may be used to replace variables existing in an expression are determined by the domain being investigated. Appendix A contains some of the general mathematical rewrite rules known to the system. (Further

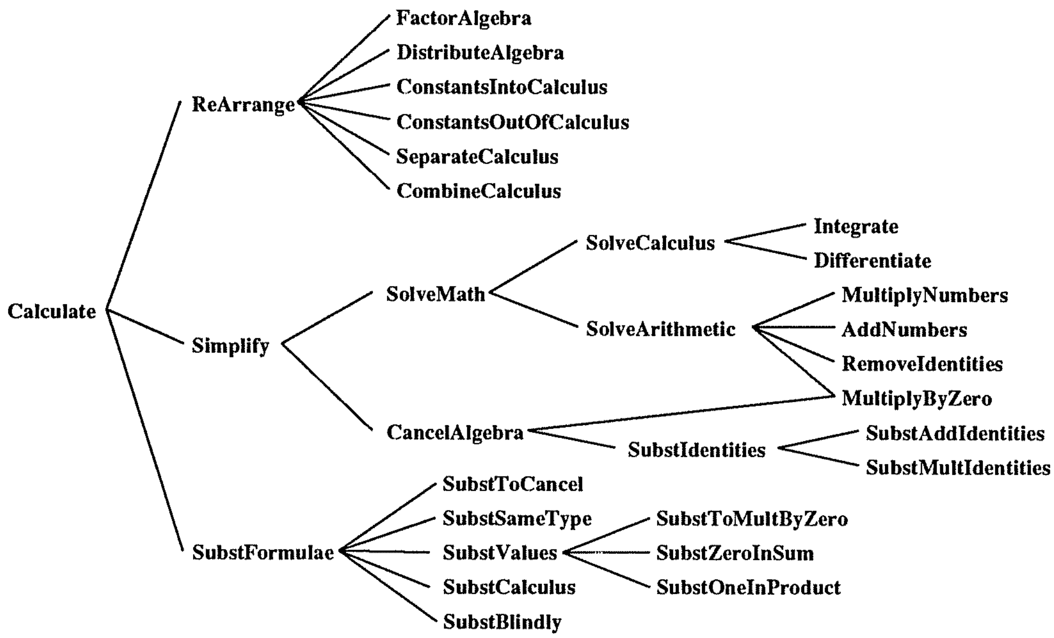

Figure 3. Some of the System's Mathematical Techniques

details on **Physics 101**'s problem solver appear in [35].)

Figure 4 contains the initial physics formulae provided to the system, along with the conditions on their applicability. The first two formulae in figure 4 define the physical concepts of velocity (V) and acceleration (A). An object's velocity is the time rate of change of its position (X) and its acceleration is the time rate of change of its velocity. Newton's second and third laws are also included. (Newton's first law need not be included because it is a special case of his second law.) The second law states that the net force (F) on an object equals its mass (M) times its acceleration (A). The net force is decomposed into two components: the external force (F_{ext}) and the internal force (F_{int}). External forces result from any external fields (e.g., gravity) that act upon objects. Object *?i* 's internal force is the sum of the forces the other objects in the world[4] exert on object *?i*. These *inter-object* forces are constrained by Newton's third law, which says that every action has an equal and opposite reaction.

Position, velocity, acceleration, and force are spatial *vectors*. Hence one of their indices (*?c*) indicates which vector component is being discussed (x, y, or z). All of these physical variables are functions of time. Mass, however, is a time-independent scalar. It is only indexed by the physical object whose mass is being specified.

When the number of objects in the world is specified, expanded versions of summations ($\sum$'s) and products ($\prod$'s) are used. For instance, if there are three objects in a world, the second from last equation in figure 4 becomes:

[4] The term *world* is used to refer to physical systems and situations. The term *system* is reserved for referring to computer programs.

$$V_{?i,?c}(t) = \frac{d}{dt} X_{?i,?c}(t)$$

Preconditions: Member(*?i,ObjectsInWorld*) $\wedge$ IsaComponent(*?c*)

$$A_{?i,?c}(t) = \frac{d}{dt} V_{?i,?c}(t)$$

Preconditions: Member(*?i,ObjectsInWorld*) $\wedge$ IsaComponent (*?c*)

$$F_{net,?i,?c}(?t) = M_{?i} * A_{?i,?c}(?t)$$

Preconditions: Member(*?i,ObjectsInWorld*) $\wedge$ IsaComponent(*?c*) $\wedge$ IsaTime(*?t*)

$$F_{net,?i,?c}(?t) = F_{ext,?i,?c}(?t) + F_{int,?i,?c}(?t)$$

Preconditions: Member(*?i,ObjectsInWorld*) $\wedge$ IsaComponent(*?c*) $\wedge$ IsaTime(*?t*)

$$F_{int,?i,?c}(?t) = \sum_{\substack{j \in ObjectsInWorld \\ j \neq ?i}} F_{j,?i,?c}(?t)$$

Preconditions: Member(*?i,ObjectsInWorld*) $\wedge$ IsaComponent(*?c*) $\wedge$ IsaTime (*?t*)

$$F_{?j,?i,?c}(?t) = -F_{?i,?j,?c}(?t)$$

Preconditions: Member(*?i,ObjectsInWorld*) $\wedge$ Member(*?j, ObjectsInWorld*) $\wedge$
$?i \neq ?j$ $\wedge$ IsaComponent(*?c*) $\wedge$ IsaTime(*?t*)

Figure 4. The Initial Physics Formulae of the System

$$F_{int,?i,?c}(?t) = F_{?j1,?i,?c}(?t) + F_{?j2,?i,?c}(?t)$$

Preconditions:
Permutation($\{?i\ ?j1\ ?j2\}$, *ObjectsInWorld*) $\wedge$ IsaComponent(*?c*) $\wedge$ IsaTime(*?t*)

Expanded equations are used because of their greater psychological-plausibility as a model of learning by a college freshman.

Problem-specific equations can also be provided when a new problem is posed. For example, it may be stated that the external force on object k is $M_k\, g\, X_{k,y}$. Generalization of solutions involving equations of this type is described in [35].

2.6. Other Approaches to Learning in Mathematical Domains

There is a substantial literature describing computer models of learning in mathematical domains. Before further describing the **Physics 101** system, these other approaches are compared to that taken in this research.

Some approaches have been inductive (e.g., [14, 16, 17]). They learn new concepts that are consistent with the examples presented to or generated by them, by analyzing the similarities and differences among the examples. Confidence in the accuracy of the acquired results grows as more examples are seen. With only a few examples to analyze, irrelevant characteristics may appear irrelevant (because, for example, they appear in all of the positive examples seen) or necessary characteristics may appear unnecessary (because, for instance, they do not appear in any negative examples). **Physics 101**, being explanation-based, produces intertwined descriptions of the role of each aspect of a problem. These explanation structures allow the system to produce, from a small number of examples, new concepts that are as certain as the underlying concepts from which the explanation is constructed. The explanation-based approach allows **Physics 101** to incorporate into new concepts descriptions of the effects of problem characteristics not seen in the example from which it learns. For example, although it experiences an example involving momentum conservation, it learns the more general equation describing how external forces affect momentum, as described later in this article.

Using analogy to learn new concepts in mathematically-based domains is another heavily investigated approach (e.g., [9], [10]). In analogical learning, new knowledge is acquired by mapping old knowledge from a well-understood domain to a novel situation. One way analogy differs from the approach taken in explanation-based systems is that in analogy attention is not focussed on combining pieces of knowledge into larger knowledge chunks (schemata) and number generalization is not addressed. Also, using analogy, no learning takes place the first time a problem is solved. Instead, an analogous problem must be encountered before the solution to the first problem contributes to the acquisition of knowledge.

There have also been a number of other explanation-based approaches to learning in mathematical domains. **LEX2** [18] is a system that learns heuristics under which an integration operator is useful. **LP** [36] analyzes worked mathematics problems and learns information that constrains search through its operator space. **LA** [25] learns new schemata for use in natural deduction proofs. Bennett's system [2] learns approximations that transform intractable problems into soluble ones. However, none of these systems generalize number.

3. BUILDING EXPLANATIONS IN PHYSICS 101

In explanation-based learning, the solution to a specific sample problem is generalized and the result saved, in the hopes of being applicable to future problems. The next section addresses the generalization of explanations. This section addresses the construction of explanations in mathematically-based domains. The focus is on understanding solutions, presented by a teacher, to problems the system could not solve on its own.[5] During this understanding process, gaps in the teacher-provided solution need filling. The teacher must provide enough guidance so that the system's limited problem solver can successfully solve the problem. If the provided solution can be sufficiently-well understood, a new schema may result, and the next time the system faces a problem related to the current one it will be able to solve it without the need for external assistance.

Understanding a teacher-provided solution involves two phases. First, the system attempts to verify that each of the instructor's solution steps mathematically follows. If successful, in the second phase the mathematical reasoning component of **Physics 101** builds an explanation of *why* the solution works. A sample collision problem illustrates these two phases.

3.1. A Sample Problem

In the one-dimensional problem shown in figure 5, there are three balls moving in free space, without the influence of any external forces. (Nothing is specified about the forces between the balls.

[5] Although the emphasis is on teacher-provided solutions, much of the discussion is this section also applies to the explanation of the system's own problem solving. It may take a substantial amount of work for **Physics 101** to solve a problem, even when it does not need external help, and this effort can be reduced in the future by creating a new schema from the analysis of its labors.

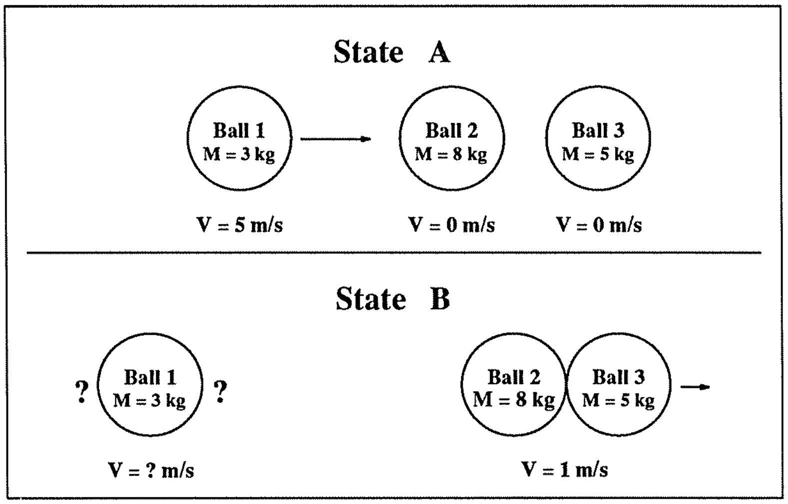

Figure 5. A Three-Body Collision Problem

Besides their mutual gravitational attraction, there could be a long-range electrical interaction and a very complicated interaction during the collision.) In the initial state (**state A**) the first ball is moving toward the other two, which are stationary. Some time later (**state B**) the second and third balls are recoiling from the resulting collision. The task is to determine the velocity of the first ball after the collision.

Physics 101 cannot solve this problem with only the physics schemata shown in figure 5. Initially, the formula $V = \frac{d}{dt}X$ is tried, which leads nowhere, then the solution steps presented below are attempted.

$$\begin{aligned} V_{1,x}(t) &= \int A_{1,x}(t)dt \\ &= \int \frac{F_{net,1,x}(t)}{M_1} dt \end{aligned}$$

The system's problem solver is incomplete, for reasons detailed in [35]. One source of incompleteness occurs because the problem solver never performs two unmotivated ("blind") substitutions consecutively. The problem solver possesses no motivated strategy to lead it past the second line in the above calculation, and asks for a solution from its teacher.

The teacher's solution to the collision problem is as follows:

$$M_1 V_{1,x}(A) + M_2 V_{2,x}(A) + M_3 V_{3,x}(A) = M_1 V_{1,x}(B) + M_2 V_{2,x}(B) + M_3 V_{3,x}(B)$$

$$(3\,kg)(-5\,\tfrac{m}{s}) = 8\,kg\ V_{1,x}(B) + (3\,kg)(1\,\tfrac{m}{s}) + (5\,kg)(1\,\tfrac{m}{s})$$

$$-15\,\tfrac{kg\,m}{s} = 8\,kg\ V_{1,x}(B) + 8\,\tfrac{kg\,m}{s}$$

$$V_{1,x}(B) = -2.88\,\tfrac{m}{s}$$

Without being explicitly stated, the principle of conservation of momentum is being invoked, as the momentum ($M \times V$) of the balls at two different times is equated. This equation is not a variation of any formula known to the system (figure 4). A physically-consistent mathematical derivation is needed if **Physics 101** is to accept the solution provided.

3.2. Verifying a Teacher's Solution

In order to accept a teacher's answer, the system must verify each of the steps in the solution. Besides being mathematically correct, the calculations must be consistent with its domain-specific knowledge. To be valid, each of the solution steps must be assigned to one of the following four classifications.

(1) Instantiation of a known formula: $force = mass \times acceleration$ is an example of this type.

(2) Definition of a new variable in order to shorten later expressions: $resistance = voltage \, / \, current$ would fall in this category.

(3) Rearrangement of a previously-used formula. These equations are mathematical variants of previous steps. The replacement of variables by their values also falls into this category.

(4) Statement of an unknown relationship among known variables. These steps require full justification, which the system must perform symbolically by reasoning about algebra and calculus. Steps in this category are candidates for generalization.

Physics 101 possesses several methods for verifying equations falling into category 4. Two are suggested when the two sides of an equation only differ as to the time at which they are evaluated (a condition satisfied by the initial equation in the teacher's solution). One method is to determine if the common form underlying each side of the equation is constant with respect to time. The second suggested method is to determine how the underlying form explicitly depends on time. If time can be made explicit, it is easy to see if it is valid to equate the expression at two different times. This method can handle more situations than the first, and is the one used by **Physics 101** to understand the teacher's solution.

The remainder of this subsection briefly describes how the system verifies the first equation in the teacher's solution. If the teacher provides the calculations justifying the equation (e.g., the steps in figure 6 below), **Physics 101** determines which rewrite rules are used and then proceeds as if it had performed the calculation itself. What is central to the explanation and generalization processes is the final calculation and not the process that produced it.

The underlying time-dependent expression could be periodic, and hence could be equated at times separated by some number of periods. Alternatively, the expression could be parabolic, hyperbolic, cubic, etc. Here there would be some other relationship between times where it is valid to equate the expression. Another possibility is that the expression is constant with respect to time. In this last case, it is valid to equate the expression at *any* two times.

Once the system selects a method for verifying a new equation, it must perform the mathematics necessary to determine any additional information required by the method. For example, it may need to determine the derivative of an expression or legally eliminate all the terms whose time-dependence is not known. In this phase, **Physics 101** relies on its mathematical knowledge.

The trace produced by the system while it is verifying the new equation appears in figure 6. The goal is to convert, via a series of equality-preserving transformations, the top expression into an equivalent expression whose time dependence is explicit. Once this is done, the system can determine if the first step in the teacher's solution is valid. (The top expression in figure 6 is called the *left-hand side* of the calculation, while the other expressions are termed *right-hand sides*.)

$$M_1 V_{1,x}(t) + M_2 V_{2,x}(t) + M_3 V_{3,x}(t)$$

1	SubstSameType	$= M_1 \int A_{1,x}(t)\,dt + M_2 \int A_{2,x}(t)\,dt + M_3 \int A_{3,x}(t)\,dt$
2	SubstToCancel	$= M_1 \int \frac{F_{net,1,x}(t)}{M_1}\,dt + M_2 \int \frac{F_{net,2,x}(t)}{M_2}\,dt + M_3 \int \frac{F_{net,3,x}(t)}{M_3}\,dt$
3	ConstsOutCalculus	$= \frac{M_1}{M_1} \int F_{net,1,x}(t)\,dt + \frac{M_2}{M_2} \int F_{net,2,x}(t)\,dt + \frac{M_3}{M_3} \int F_{net,3,x}(t)\,dt$
4	SubMultIdentities	$= 1 \int F_{net,1,x}(t)\,dt + 1 \int F_{net,2,x}(t)\,dt + 1 \int F_{net,3,x}(t)\,dt$
5	RemoveIdentities	$= \int F_{net,1,x}(t)\,dt + \int F_{net,2,x}(t)\,dt + \int F_{net,3,x}(t)\,dt$
6	SubstSameType	$= \int (F_{ext,1,x}(t) + F_{int,1,x}(t))\,dt + \int (F_{ext,2,x}(t) + F_{int,2,x}(t))\,dt + \int (F_{ext,3,x}(t) + F_{int,3,x}(t))\,dt$
7	SubstSameType	$= \int (F_{ext,1,x}(t) + F_{2,1,x}(t) + F_{3,1,x}(t))\,dt + \int (F_{ext,2,x}(t) + F_{1,2,x}(t) + F_{3,2,x}(t))\,dt + \int (F_{ext,3,x}(t) + F_{1,3,x}(t) + F_{2,3,x}(t))\,dt$
8	SubstToCancel	$= \int (F_{ext,1,x}(t) + F_{2,1,x}(t) + F_{3,1,x}(t))\,dt + \int (F_{ext,2,x}(t) - F_{2,1,x}(t) + F_{3,2,x}(t))\,dt + \int (F_{ext,3,x}(t) - F_{3,1,x}(t) - F_{3,2,x}(t))\,dt$
9	CombineCalculus	$= \int (F_{ext,1,x}(t) + F_{2,1,x}(t) + F_{3,1,x}(t) + F_{ext,2,x}(t) - F_{2,1,x}(t) + F_{3,2,x}(t) + F_{ext,3,x}(t) - F_{3,1,x}(t) - F_{3,1,x}(t))\,dt$
10	SubAddIdentities	$= \int (F_{ext,1,x}(t) + 0\,\frac{kg\,m}{s^2} + 0\,\frac{kg\,m}{s^2} + F_{ext,2,x}(t) + 0\,\frac{kg\,m}{s^2} + F_{ext,3,x}(t))\,dt$
11	AddNumbers	$= \int (F_{ext,1,x}(t) + F_{ext,2,x}(t) + F_{ext,3,x}(t) + 0\,\frac{kg\,m}{s^2})\,dt$
12	RemoveIdentities	$= \int (F_{ext,1,x}(t) + F_{ext,2,x}(t) + F_{ext,3,x}(t))\,dt$
13	SubstValues	$= \int (0\,\frac{kg\,m}{s^2} + 0\,\frac{kg\,m}{s^2} + 0\,\frac{kg\,m}{s^2})\,dt$
14	AddNumbers	$= \int 0\,\frac{kg\,m}{s^2}\,dt$
15	Integrate	$= constant_1$

Figure 6. Verifying the First Equation in the Teacher's Solution

Three qualitatively different strategies are used by **Physics 101** during various stages of problem solving. The first strategy is to apply operators that will lead to definite progress toward the goal. Attention is focussed by this strategy as long as some operator in this class is applicable. When clear progress cannot be achieved, the problem solver must decide how best to proceed. The second strategy is then invoked to select operators that preserve important characteristics of the current problem. These operators are likely to keep the problem solver from diverging sharply from the goal while possibly enabling the application of operators by the first strategy. When the problem solver can follow neither of the first two strategies, the third strategy of arbitrarily applying legal operators is used. To minimize blind search, the third strategy is never used more than once consecutively. This problem-solving model is described more fully in [35].

The annotations to the left of the expressions in figure 6 are produced by the system. These annotations indicate which of **Physics 101**'s problem-solving schemata (figure 4) is used to perform each calculation step. In the first step, the velocities are replaced by the integrals of acceleration.[6] In the next step, the formulae substitutions are chosen because the mass terms can be cancelled. Before this cancellation can take place, however, the cancelling terms must be brought together. The calculation continues in a like manner until all the unknown variables are eliminated. Then the known values are substituted and the ensuing arithmetic and calculus is solved. Since the initial expression is constant, it can be equated at any two times. The first equation in the teacher's solution is valid.

3.3. Explaining Solutions

At this point, the system has ascertained that its teacher's use of a new equation is indeed valid. Figure 6 can be viewed as an explanation structure. Underlying the calculation are a large number of equation schemata, along with their associated preconditions. These schemata justify the transformations from one line to the next. Between each pair of consecutive expressions are the equation schemata (and their preconditions) used to rewrite one expression into the next. This is illustrated in figure 7. One side of this figure shows the general structure existing between consecutive expressions, while the other presents a simple, concrete example.

However, although figure 6 constitutes an acceptable explanation of the solution to the specific example of figure 6, the underlying explanation does not directly suffice to produce the proper general concept. Applying a standard explanation-based generalization algorithm does produce a generalization of the specific solution (see section 4.2), and a number of attributes of the problem are generalized. For example, the result does not only apply to colliding balls - it applies to situations involving any type of physical object. Nor does the problem have to be in the x-direction, since none of the schemata used to

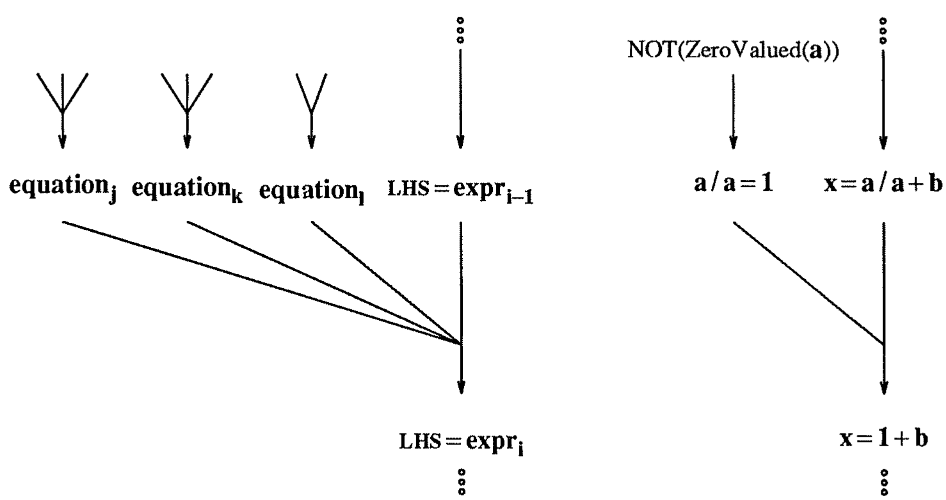

Figure 7. The Underlying Structure of a Calculation

[6] Initially, the system chooses to replace the velocities by the derivative of the positions. This leads nowhere and the system backtracks. No other backtracking occurs during the calculation of figure 6. The system is guided by the goal of cancelling variables, which greatly reduces the amount of unnecessary substitutions during problem solving.

tie the calculation together constrain the component of the variables. A third property generalized is that the external forces need not individually be zero; all that step 15 of figure 6 requires is that the external forces sum to zero. However, since the explanation structure is not generalized, one unfortunate aspect of the specific example remains. The new law only applies to physical situations involving *three* objects. Without that property, adding the $M\ V$ terms of three objects will not lead to the complete cancellation of internal forces of the objects. The preconditions of the new schema would insist on a three-object world without external forces, because only then will the sum of three momentum terms always be constant across time. Unfortunately this result is not broadly applicable. The system would need to learn separate rules when it encountered a four-object system, a five-object system, etc.

In order to produce the proper generalization to N, the system must determine a reason for including each variable in this equation. This will determine which variables are required in its general form.

In the explanation process, **Physics 101** determines how the value of the desired property of the current problem's *unknown* is obtained. As stated earlier, the problem's unknown is the expression about which the value of some property is being sought; in the sample problem, V_1 is the unknown and its value in state B is being sought. During this process, the system determines the role of each variable in the initial expression of the calculation.

During a calculation there are three ways to eliminate a variable:

(1) its value can be substituted,

(2) it can be symbolically replaced during a formula substitution, or

(3) it can be *cancelled.*

Recall that *obstacles* are expressions appearing in a calculation but whose values are not known. *Primary obstacles* are obstacles descended from the unknown. In the momentum problem the only primary obstacles not replaced in a formula substitution are $F_{2,1}$ and $F_{3,1}$. If the value of the desired property of each of the primary obstacles were known, the value of the unknown's desired property would be specified. The system ascertains how these obstacles are eliminated from the calculation. Cancelling obstacles is seen as the essence of the solution strategy, because when all the obstacles have been cancelled the value of the unknown's desired property can be easily calculated.

Physics 101 can identify the first five of the following six types of variable cancellations:

additive identity
: These are algebraic cancellations of the form $x - x = 0$. Line 10 in figure 6 contains two additive cancellations.

multiplicative identity
: These are algebraic cancellations of the form $x / x = 1$. Line 4 in figure 6 involves two multiplicative cancellations.

multiplication by zero
: These are cancellations that result from an expression (which may contain several variables) being multiplied by zero. None appear in figure 6.

integration (to a number)
: This type of cancellation occurs when variables disappear during symbolic integration. When integration produces *new* variables (other than the integration constant), this calculation is viewed as a substitution involving the original terms. No cancellations of this type appear in figure 6.

differentiation (to a number)
This is analogous to cancellation during integration.

assumed ignorable
A term can be additively ignored because it is assumed to be approximately zero or multiplicatively ignored because it is assumed to be approximately equal to one.

Figure 8 illustrates the concept of primary obstacles. The goal in this sample problem is to determine the value of V_1. Since this is not known, the problem is transformed to that of finding A_1 (for simplicity, the integral sign is ignored here). However, the value of A_1 is not known either. This leads to the substitution of $F_{net,1}$ divided by M_1. The mass is known, but the net force is not. The net force is then decomposed into two components - a known external force and an unknown internal force. Finally, the internal force is further decomposed into its constituents. These two inter-object forces are the obstacles to knowing the value of $V_{1,x}$. **Physics 101** needs to determine how the solution in figure 6 circumvents the need to know the value of these two variables.

Figure 9 contains the *cancellation graph* for the three-body collision problem. This data structure is built by the system during the understanding of the specific solution. The process by which this is done is described in the next subsection. The graph holds the information that explains how the specific example's obstacles are eliminated from the calculation. This information is used to guide the generalization process described in the next section.

To reason about a calculation, **Physics 101** must be able to distinguish different instances of the same variable. For example, the M_1 introduced in line 2 of figure 6 plays a different role than the M_1 appearing in the left-hand side of the equation. In the system, variables are marked to record in which solution step they first appear. This information is recorded in the above cancellation graph by the subscript preceding a variable. Variables originating in the left-hand side are prefixed with zeroes.

To understand the calculation, the system first determines that the primary obstacles $F_{2,1,x}$ and $F_{3,1,x}$ are eliminated by being *additively cancelled.* Although cancelled additively, these variables descended from a multiplicative expression ($A = \frac{F}{M}$). Hence, the system must determine how they are *additively isolated.* Multiplication by M_1 performed this task. So an explanation of the M_1 term in the

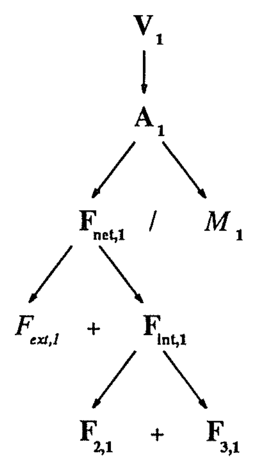

Figure 8. Decomposing the Unknown

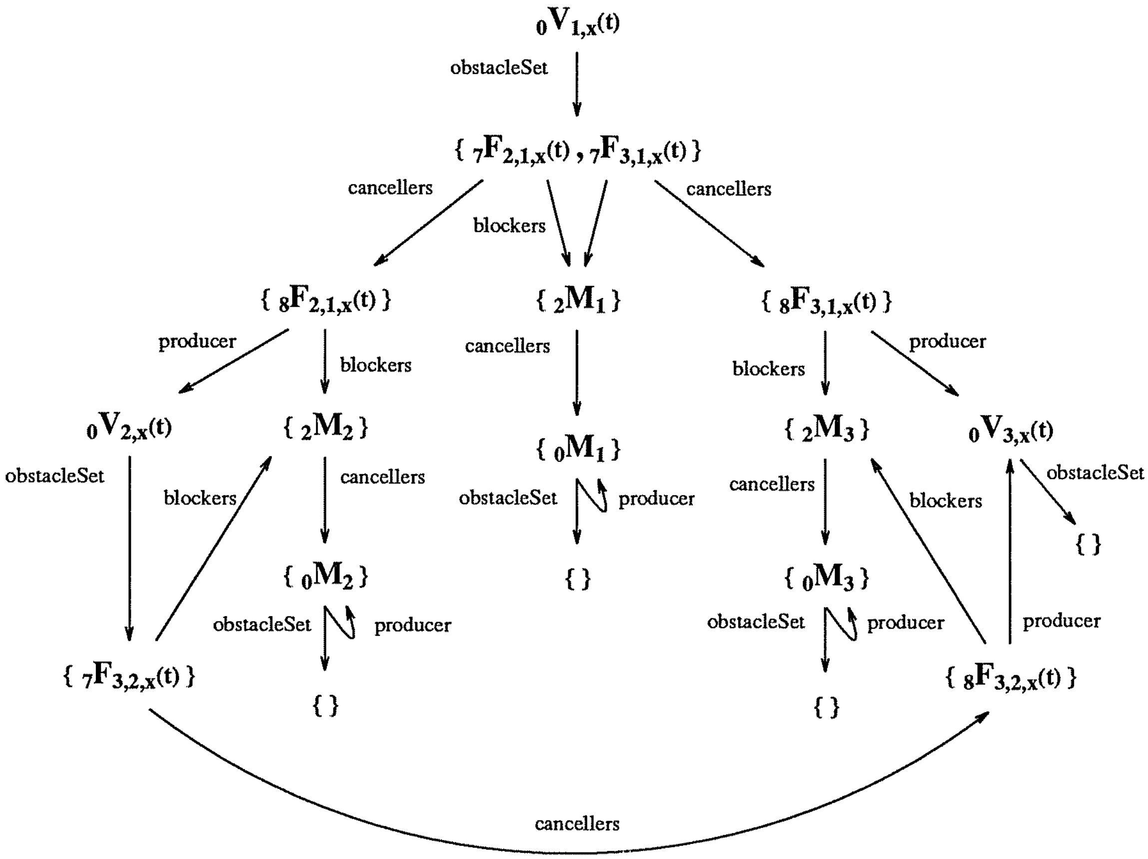

Figure 9. The Cancellation Graph

left-hand side expression of figure 6 is obtained.

The next thing to do is to determine how the terms that additively cancel $F_{2,1,x}$ and $F_{3,1,x}$ are introduced into the calculation. $F_{2,1,x}$ is cancelled by a force descended from $V_{2,x}$. This $F_{2,1,x}$, too, must first be additively isolated. **Physics 101** discovers that the left-hand side's M_2 performs this isolation. The system now has explanations for the M_2 and the $V_{2,x}$ terms in the left-hand side. Similar reasoning determines the role of M_3 and $V_{3,x}$.

Cancellation of the primary obstacles requires the presence of additional variables on the left-hand side of the equation. These extra terms may themselves contain obstacle variables. These are called *secondary obstacles*. The system must also determine how these obstacles are eliminated from the calculation. The elimination of the secondary obstacles may in turn require the presence of additional variables in the left-hand side expression, which may introduce additional secondary obstacles. This recursion must terminate, however, as the calculation is known to have eliminated all of the unacceptable terms.

Cancelling the inter-object forces involving ball 1 introduced one secondary obstacle — $F_{3,2,x}$. This secondary obstacle was additively cancelled by a force descended from $V_{3,x}$. Cancelling this secondary obstacle produced no new obstacles.

Once the system determines how all of the obstacles in the calculation are cancelled, generalization can occur. At this time, **Physics 101** can also report any variables in the left-hand side of a calculation that are irrelevant to the determination of the value of the unknown. Those variables not visited during the construction of the cancellation graph are not necessary, even though they are present in the teacher's solution.

3.4. Constructing the Cancellation Graph - Algorithmic Details

This section presents the cancellation graph algorithm and illustrates it with an algebraic example. Although obstacle cancellation guides the system's construction of calculations, this algorithm assumes that a *complete* calculation has been produced. The cancellation graph algorithm then analyzes the calculation and ascertains why it achieved its goal. Hence, the algorithm can also be applied to analyzing complete calculations produced by an external agent.

The algorithm is expressed here in a pseudo-code, designed for readability. (The actual implementation is written in Lisp.) In this algorithm, the notation *record.field* is used to represent a given field of a record, and back arrows (←) indicate value assignment. The construct

for each *element* **in** *set* **unless** *test* **do** *statement*

means that *element* is successively bound to each member of *set* and, unless *test* is true, *statement* is evaluated.

Figure 10 contains the algorithm for constructing cancellation graphs, while table 1 lists the fields referenced and describes their setting when the algorithm commences. (Fields ending with question marks are boolean-valued.) Further details on these fields are provided as the algorithm is discussed.

These graphs record the roles played by each of the variables in the calculation. A sample problem is used to illustrate the algorithm. It is contained in table 2, and its cancellation graph appears in figure 11. The features of this specific cancellation graph are explained as the algorithm is described. Recall that the first expression in a sequence of calculation steps is called the *left-hand side* of the calculation, while the subsequent expressions are termed *right-hand sides.*

In the sample calculation, Greek letters are used for variables that cannot appear in the final right-hand side of the calculation. For example, it may be that the final right-hand side can only contain variables whose value is known or variables that are time-independent. The goal is to rewrite the left-hand side, via a series of equality-preserving transformations, into an expression containing no Greek letters. The equation schemata used to produce each new expression are shown in the right column. A mixture of domain-specific and general algebraic rewrite rules are used. β is the variable about which some information is sought, i.e., it is the problem's unknown. In this example, subscripts are used to differentiate various instantiations of the *same* variable (e.g., b_1 and b_2). In other words, b_1 and b_2 both refer to the variable b.

This sample calculation illustrates the major components and attributes of cancellation graphs. The graph specifies the role of each variable in the left-hand side of the calculation. Combined together in the given manner, these variables support the answer to the question about the unknown. One important feature is that unnecessary variables (e.g., α in table 1) do not appear in the graph.

The first step in building a cancellation graph is to create the *primary obstacle set* for the problem's unknown. An obstacle set for a collection of variables consists of those variables possessing the following properties:

(1) they are in the given collection or descended (via rewriting with equation schemata) from a member of it,

```
procedure CheckForObstacles (variable)
    if variable.notChecked? then
        variable.notChecked? ← false
        variable.obstacleSet ← FindObstacleSet({variable})
        AnalyzeObstacles(variable.obstacleSet)

procedure FindObstacleSet (variables)
    answer ← { }                                        /* answer is a local variable. */
    for each variable in variables unless variable.eliminated?
        do  if variable.replaced?
                then   answer ← answer ∪ FindObstacleSet(variable.descendants)
            else if variable.unacceptable?
                then   answer ← answer ∪ {variable}
    return answer

procedure AnalyzeObstacles (obstacleSet)
    for each obstacle in obstacleSet do Eliminate(obstacle)
    for each obstacle in obstacleSet do CheckForSecondaryObstacles(obstacle)

procedure Eliminate (variable)
    variable.eliminated? ← true
    AnalyzeBlockers(variable.blockers)
    AnalyzePartners(variable.partners)
    AnalyzeCancellers(variable.cancellers)

procedure AnalyzeBlockers (blockers)
    for each blocker in blockers unless blocker.eliminated? do Eliminate(blocker)

procedure AnalyzePartners (partners)
    for each partner in partners unless partner.eliminated?
        do  partner.eliminated? ← true
            AnalyzeBlockers(partner.blockers)

procedure AnalyzeCancellers (cancellers)
    for each canceller in cancellers
        do  canceller.eliminated? ← true
            AnalyzeBlockers(canceller.blockers)

procedure CheckForSecondaryObstacles (variable)
    for each blocker in variable.blockers do CheckForSecondaryObstacles(blocker)
    for each partner in variable.partners
        do  CheckForObstacles(partner.producer)
            for each blocker in partner.blockers do CheckForSecondaryObstacles(blocker)
    for each canceller in variable.cancellers
        do  CheckForObstacles(canceller.producer)
            for each blocker in canceller.blockers do CheckForSecondaryObstacles(blocker)
```

```
CheckForObstacles(unknown)                              /* Construct the cancellation graph. */
```

Figure 10. Building the Cancellation Graph

Table 1. Fields Used in the Cancellation Graph Algorithm

Field	Initial Setting
notChecked?	initially *true*, reset in algorithm when checked for obstacles
obstacleSet	initially *{}*, set in algorithm
unacceptable?	*true* if this variable cannot appear in the final right-hand side of the calculation
eliminated?	initially *false*, set in algorithm
replaced?	*true* if replaced by a rewrite rule during the calculation
descendants	variables introduced when this variable rewritten
blockers	set of variables preventing cancellation of this variable
partners	set of variables in the cancelled expression in addition to this variable
cancellers	set of variables that cancelled this variable
producer	variable in the left-hand side of the calculation that produced this variable

(2) they cannot appear in the final right-hand side of a calculation,

(3) they are cancelled in the calculation, and

(4) they are not marked earlier in the algorithm as being eliminated from consideration.

The function *FindObstacleSet* assumes that the provided calculation is successful. This means that if an unacceptable variable is not replaced, it must be cancelled, since there can be no unacceptable variables in the final right-hand side of a calculation.

The next things to do are see how the obstacle is cancelled and then determine if there are any undesirable side-effects of doing this. If there are side-effects, how they are circumvented in the specific example needs to be determined.

An obstacle set is analyzed once it is constructed, and how each element of this set is eliminated from the calculation is determined. This involves three steps.

(1) Determining how those variables blocking the cancellation are eliminated. *Blockers* are variables preventing additive or multiplicative access (depending on the manner of cancellation) to an obstacle.

(2) Determining how the obstacle's cancellation partners are brought into position for the cancellation. *Partners* are variables that must be present in combination with the obstacle in order for the obstacle to be cancelled.

(3) Determining how the cancellers of the obstacle are brought into position. *Cancellers* are the variables that directly cancel an obstacle.

Once all the members of an obstacle set are eliminated, a check is made (in the procedure *CheckForSecondaryObstacles*) to see if this process lead to the existence of additional (secondary) obstacle sets. Cancelling an obstacle requires the presence of its partners and cancellers, as well as those variables that cancelled the blockers of the obstacle, the blockers of its partners, and the blockers of its cancellers. Hence, additional variables must be present in the left-hand side of the calculation. These *producer* variables are checked to see if their presence means additional obstacle sets are present in the

Table 2. Sample Calculation

	Calculation Steps	Rewrite Rules Used
	$\alpha((a - b_1)\frac{\beta}{c_1} + d)g$	β is the unknown
(1)	$= f((a - b_1)\frac{\beta}{c_1} + d)g$	$\alpha = f$
(2)	$= f((a - b_1)\frac{\delta_1 c_2}{c_1} + d)g$	$\beta = \delta c$
(3)	$= f((a - b_1)\delta_1 + d)g$	$?x / ?x = 1$ $?x\ 1 = ?x$
(4)	$= f((\varepsilon_1 + b_2 - b_1)\delta_1 + d)g$	$a = \varepsilon + b$
(5)	$= f(\varepsilon_1 \delta_1 + d)g$	$?x - ?x = 0$ $?x + 0 = ?x$
(6)	$= f(\varepsilon_1 \delta_1 + \gamma_1 - \varepsilon_2 \delta_2)g$	$d = \gamma - \varepsilon\delta$
(7)	$= f \gamma_1 g$	$?x + ?y = ?y + ?x$ $?x - ?x = 0$ $?x + 0 = ?x$
(8)	$= \frac{f \gamma_1 h}{\gamma_2}$	$g = \frac{h}{\gamma}$
(9)	$= fh$	$?x / ?x = 1$ $1\ ?x = ?x$

calculation. If so, these secondary obstacle sets are analyzed in the same manner as the primary obstacle set.

Notice that eliminating the secondary obstacles can also require additional variables in the the left-hand side of the calculation. These variables may also have obstacles sets (again called secondary sets) that need to be eliminated. Since the successful calculation contained a finite number of calculation steps (and, hence, a finite number of potential obstacle variables), the algorithm always terminates.

Also notice that all of the variables, other than α, in the left-hand side of table 2's calculation appear in figure 10. This graph records the role of each variable in the determination of the desired property of the unknown. The variable α does not appear in the cancellation graph because it serves no purpose in the achievement of the goal. One of the important features of the cancellation graph process is that irrelevant variables in the left-hand expression can be detected.

4. GENERALIZING SOLUTIONS IN PHYSICS 101

Once the solution to a problem is understood (via the construction of a cancellation graph), the solution must be generalized so that it can be used to help solve similar problems in the future. In

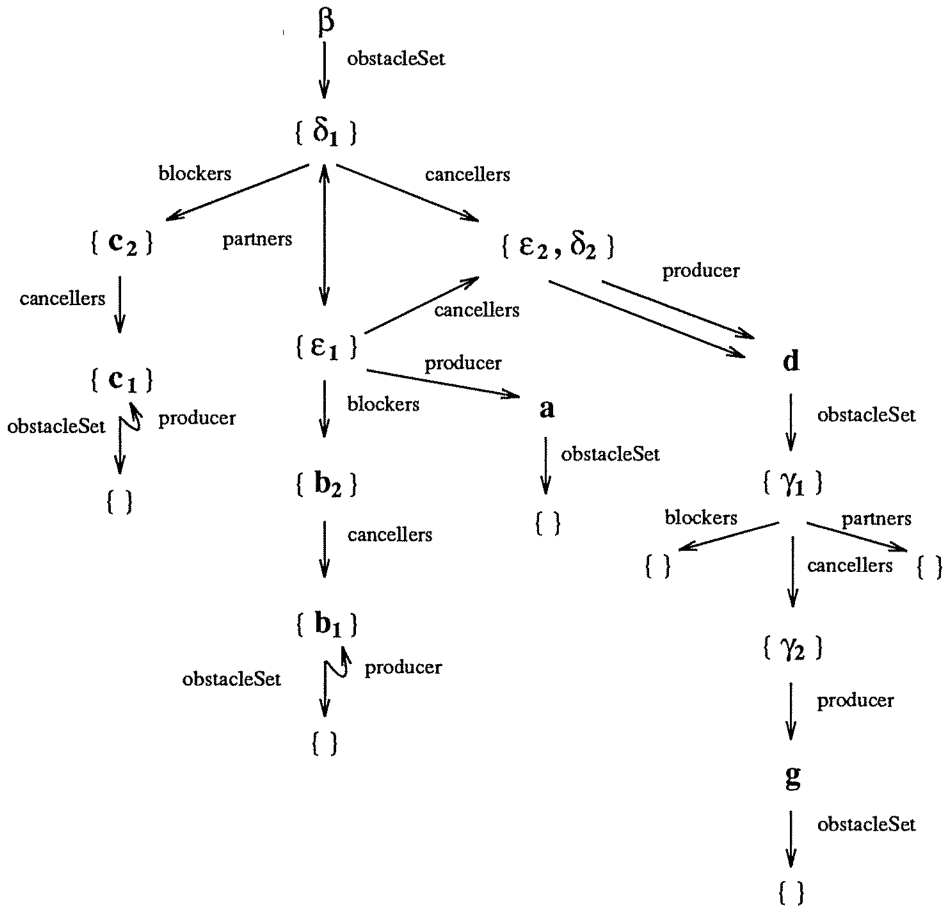

Figure 11. The Cancellation Graph for the Calculation in Table 2

Physics 101 this involves generalizing the structure of the specific problem's solution, as well as generalizing the constants in the specific example. This section presents the process by which the general version of a specific problem's solution is constructed. Also in this section the generalization produced by standard explanation-based learning algorithms is presented, the construction of special-case schemata is discussed, and performance improvement after learning is empirically analyzed.

4.1. The Generalization Algorithm

Physics 101 performs generalization by using its explanation of the specific solution (the *cancellation graph*) to guide the determination of the problem's unknown in the general case. This entails reconstructing the specific solution in its general form under the guidance of the cancellation graph. In other words, the structure of the explanation of the specific problem's solution is generalized. This section presents the algorithm and uses the three-body collision example to illustrate the generalization process.

Figure 12 presents the algorithm for reconstructing the calculation. This algorithm is very similar to that used for building the cancellation graph. Obstacles and their blockers, partners, and cancellers are again the focus. Basically, the algorithm works as follows. First, the general version of the unknown is used as the initial left-hand side of the calculation, and then the general versions of the equation schemata used in the specific calculation are applied until the primary obstacles are introduced in their general

```
procedure AnalyzeGeneralObstacles (variable)
    for each obstacle in variable.obstacleSet do IntroduceIntoGeneralCalculation(obstacle)
    for each obstacle in variable.obstacleSet unless OR(obstacle.notUsed? obstacle.eliminated?)
        do EliminateInGeneral(obstacle)
    for each obstacle in variable.obstacleSet unless OR(obstacle.notUsed? obstacle.eliminated?)
        do EliminateSecondaryObstacles(obstacle)
procedure EliminateInGeneral (variable)
    EliminateGeneralBlockers(variable.blockers)
    IntroduceGeneralPartners(variable.partners)
    IntroduceGeneralCancellers(variable.cancellers)
    RecordGeneralCancellation(variable.generalVersion)
procedure EliminateGeneralBlockers (blockers)
    for each blocker in blockers unless blocker.eliminated? do EliminateInGeneral(blocker)
procedure IntroduceGeneralPartners (partners)
    for each partner in partners unless partner.eliminated?
        do      IntroduceIntoGeneralCalculation(partner)
                EliminateGeneralBlockers(partner.blockers)
procedure IntroduceGeneralCancellers (cancellers)
    for each canceller in cancellers
        do      IntroduceIntoGeneralCalculation(canceller)
                EliminateGeneralBlockers(canceller.blockers)
procedure EliminateSecondaryObstacles (variable)
    for each blocker in variable.blockers do EliminateSecondaryObstacles(blocker)
    for each partner in variable.partners
        do      AnalyzeGeneralObstacles(partner.producer)
                for each blocker in partner.blockers do EliminateSecondaryObstacles(blocker)
    for each canceller in variable.cancellers
        do      AnalyzeGeneralObstacles(canceller.producer)
                for each blocker in canceller.blockers do EliminateSecondaryObstacles(blocker)
procedure IntroduceIntoGeneralCalculation (variable)
                <described in text>
procedure RecordGeneralCancellation (generalVariable)
                <described in text>
```

```
AnalyzeGeneralObstacles(unknown)        /* Reconstruct the calculation. */
SimplifyCalculation( )                  /* Simplify the new calculation. */
```

Figure 12. Building the General Version of the Calculation

form.[7] Next, the general versions of any blockers are eliminated, which may lead to the introduction of more general variables in the left-hand side of the calculation. Following this, any partners of the obstacle are introduced, also in their general form, and their blockers are eliminated. The general versions of the cancellers are then introduced and their general blockers eliminated. After this, the cancellation is recorded and the obstacles and their cancellers are eliminated from the calculation. Finally, secondary obstacles are recursively analyzed.

Introducing a variable involves finding its *producer* in the general case, inserting the producer into the left-hand side of the calculation, then performing the general versions of the substitutions performed in the specific calculation into the general version of the variable appears in the general calculation.

Recording a cancellation involves altering the ranges of the variables involved in the cancellation, removing the cancelled possibilities. The necessary unifications are also recorded. If the range of a subscript becomes empty, the general variable is marked as being eliminated. To illustrate this, assume that the range of one instantiation of the variable X_j is the singleton i, while the range of its canceller is from 1 to n. After the cancellation is recorded, the first instantiation is cancelled and the second has the new range of 1 to n except i.

In the remainder of this subsection, the momentum example is used to illustrate the generalization process. The system starts with the generalized unknown, $V^{?n}{}_{?s,?c}(?arg)$. It then performs the general versions of the specific formulae substitutions that produced the first of the primary obstacles. This can be seen, for the collision problem, in the equation below. (Appendix B presents the details on how the equations in this section are derived by the system.)

$$V_{?s,?c}(t) = \frac{1}{M_{?s}} \int (F_{ext,?s,?c}(t) + \sum_{\substack{j \varepsilon ObjectsInWorld \\ j \neq ?s}} F_{j,?s,?c}(t))\, dt$$

While the general equation schemata are being applied, a global unification list is maintained, in the manner of the **EGGS** system [21]. This process determines how the terms in the new general formulae used must relate to ones already in the general calculation. For example, $?arg$ in the generalized unknown is constrained to be t and $?n$ is constrained to be 1, since the first step of figure 6 applies the second equation of figure 4 to the unknown. Unifications that are needed to satisfy the preconditions of the equation schemata are also maintained.

Recall from section 3.3 that the inter-object forces are *additively* cancelled in the specific case. Hence, the next generalization step is to additively isolate each inter-object force. $M_{?s}$ is introduced into the left-hand side of the general calculation in order to accomplish this isolation. The next equation contains this generalization step.

$$M_{?s}\, V_{?s,?c}(t) = \int (F_{ext,?s,?c}(t) + \sum_{\substack{j \varepsilon ObjectsInWorld \\ j \neq ?s}} F_{j,?s,?c}(t))\, dt$$

At this point the general versions of the primary obstacles are isolated for additive cancellation. To perform this cancellation, those terms that will cancel the intra-object forces must be introduced into the

[7] The unexpanded forms of equations are used. Recall that in solving the specific problem, indefinite summations ($\sum$) and products ($\prod$) are expanded into ordinary sums and products.

general calculation. The system determines that in the specific solution each inter-object force acting on ball 1 is cancelled by the equal-but-opposite inter-object force specified by Newton's third law.

In the general case, *all* of the other objects in a situation exert an inter-object force on object $?s$. *All* of these inter-object forces need to be cancelled. In the specific case, $M_2 \times V_2$ produced and isolated the additive canceller of $F_{2,1}$ while $M_3 \times V_3$ produced and isolated the additive canceller of $F_{3,1}$. So to cancel object $?s$'s inter-object forces, an $M_j \times V_j$ term must come from every other object in the situation. The following equation contains the introduction of the summation that produces the terms that cancel object $?s$'s inter-object forces. Notice how the goal of cancellation motivates generalizing the number of objects involved in this expression.

$$M_{?s}\, V_{?s,?c}(t) + \sum_{\substack{j \,\varepsilon\, ObjectsInWorld \\ j \neq ?s}} M_j\, V_{j,?c}(t)$$

$$= \int (F_{ext,?s,?c}(t) + F_{int,?s,?c}(t))\, dt + \sum_{j \neq ?s} \int (F_{ext,j,?c}(t) + F_{int,j,?c}(t))\, dt$$

Once all the cancellers of the generalized primary obstacle are present, the primary obstacle itself can be cancelled. This is shown in the next equation.

$$M_{?s}\, V_{?s,?c}(t) + \sum_{\substack{j \,\varepsilon\, ObjectsInWorld \\ j \neq ?s}} M_j\, V_{j,?c}(t) = \int (F_{ext,?s,?c}(t) + 0\, \frac{kg\ m}{s^2} + \sum_{j \neq ?s} [F_{ext,j,?c}(t) + \sum_{k \neq j,?s} F_{k,j,?c}(t)])\, dt$$

Now that the primary obstacles are cancelled, the system checks to see if any secondary obstacles have been introduced. As can be seen in the previous equation, the inter-object forces *not* involving object $?s$ still remain in the expression. These are secondary obstacles. Figure 13 graphically illustrates these remaining forces in a situation containing N objects. All of the forces acting on object $?s$ have been cancelled, while a force between objects j and k still appears whenever neither j nor k equal $?s$. This highlights an important aspect of generalizing to N. Introducing more entities may create new interactions that do not appear and, hence, are not addressed in the specific example. Only when the specific example illustrates how to circumvent all of the potentially troublesome interactions, does **Physics 101** learn a new schema.

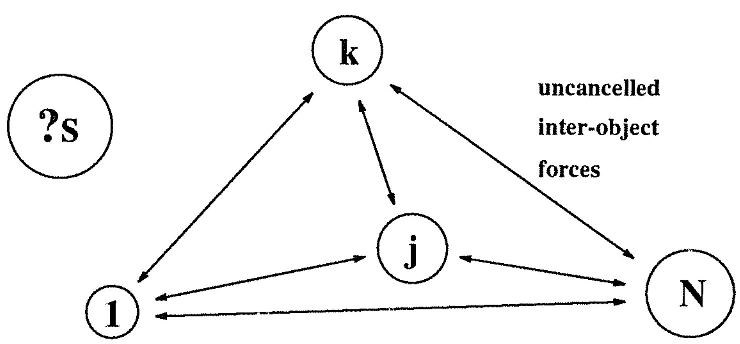

Figure 13. The Remaining Inter-Object Forces

Physics 101 cannot eliminate the remaining inter-object forces if the specific example only involves a two-object collision. The system does not detect that the remaining forces all cancel one another, since in the two-object example there is no hint of how to deal with these secondary obstacles. A collision involving three or more must be analyzed by the system to properly motivate this cancellation. (More details on the reasons for this are given in chapter 7 of [35], which presents a two-body collision problem.) In the three-body collision problem, the system continues, producing

$$M_{?s}\, V_{?s,?c}(t) + \sum_{\substack{j \in ObjectsInWorld \\ j \neq ?s}} M_j\, V_{j,?c}(t) = \int (F_{ext,?s,?c}(t) + \sum_{j \neq ?s} F_{ext,j,?c}(t))\, dt$$

Once all possible obstacle cancellations of the cancellation graph have been produced, **Physics 101** produces the final result. The preconditions of each equation schemata are collected, the global unification list is used to determine the final form of each variable in these preconditions, and the final result is simplified. This process produces the restrictions that the masses of the objects be constant over time (since each was factored out of a temporal integral — see figure 6), and that the objects cannot have zero mass (since their masses appear in the denominator of expressions). The final result is shown in figure 14. The new equation is recorded, along with its preconditions. In addition, the terms cancelled in the general calculation are recorded. Although not implemented in **Physics 101**, the eliminated terms could be used to help index the acquired formula. For example, when the inter-object forces are not specified, this equation schema could be suggested as possibly being appropriate.

In addition to trancending situations containing exactly three objects, the newly-acquired formula applies to situations where the external forces are not all zero. An appreciation of how the external forces effect momentum is obtained. This process also determines that there is no constraint that restricts this formula to the x-direction. It applies equally well to the y- and z-components of V. Hence, the acquired formula is a vector law. Notice that those physics variables whose values are used in the specific solution (e.g., the F_{ext}) remain in the general formula. The final equation is added to **Physics 101**'s collection of general formulae. (If **Physics 101** generalizes the two-body collision it would produce in an expression still containing those inter-object forces that do not involve object i. However, this formula would not be kept.) The new formula says: *The rate of change of the total momentum of a collection of objects is completely determined by the sum of the external forces on those objects.* Other problems, which involve any number of bodies under the influence of external forces, can be solved by the system using this

---

Equation

$$\frac{d}{dt} \sum_{i \in ObjectsInWorld} M_i\, V_{i,?c}(t) = \sum_{i \in ObjectsInWorld} F_{ext,i,?c}(t)$$

Preconditions

IsaComponent($?c$) $\wedge \forall\, i \in ObjectsInWorld$ NOT(ZeroValued(M_i)) $\wedge \forall\, i \in ObjectsInWorld$ IndependentOf(M_i, t)

Figure 14. The Final Result

---

generalized result. For example, it can be used to solve the three-dimensional collision problem involving four objects, where there are external forces due to gravity.

Not all of the preconditions of the equation schemata in a calculation appear in the final result. Each equation schema providing support in a calculation may have associated with it propositions that are known to be true. *Known facts* are filtered out of the final result. For example, if $\sum_{\substack{j \varepsilon ObjectsInWorld \\ j \neq ?i}}$ appears in an equation schema, that schema's known facts include $Member(j, ObjectsInWorld)$ and $j \neq ?i$. Preconditions of the equation schemata used in a calculation that match any of the known facts do not appear in the final collection of preconditions. For instance, a precondition of the equation $F_{j,?i} = -F_{?i,j}$ is $j \neq ?i$. This precondition does not appear in figure because the use of another schema leads to this precondition being a known fact.

4.2. The Result of Standard Explanation-Based Learning

Standard explanation-based learning generalization algorithms (e.g., [20, 21]) can be applied to the explanation structure underlying a calculation. The result obtained by doing this with the calculation of figure 6 is presented in figure 15.

A number of characteristics of the sample problem are generalized, many with the same result as in the **Physics 101** system. For example, the x-component of the velocities need not be used. The technique applies to any vector component. Also, because of the equation schemata used, the masses of the three objects must be non-zero and constant with respect to time. The cancellations of the inter-object forces

---

Equation

$$M_{?i_1} V_{?i_1,?c}(?t_1) + M_{?i_2} V_{?i_2,?c}(?t_1) + M_{?i_3} V_{?i_3,?c}(?t_1)$$

$$= M_{?i_1} V_{i_1,?c}(?t_2) + M_{?i_2} V_{i_2,?c}(?t_2) + M_{?i_3} V_{i_3,?c}(?t_2)$$

Preconditions

IsaComponent($?c$) $\wedge$ $?t_1 \neq ?t_2$ $\wedge$

NOT(ZeroValued($M_{?i_1}$)) $\wedge$ NOT(ZeroValued($M_{?i_2}$)) $\wedge$ NOT(ZeroValued($M_{?i_3}$)) $\wedge$

IndependentOf($M_{?i_1}$, t) $\wedge$ IndependentOf($M_{?i_2}$, t) $\wedge$ IndependentOf($M_{?i_3}$, t) $\wedge$

$?i_1 \neq ?i_2 \wedge ?i_1 \neq ?i_3 \wedge ?i_2 \neq ?i_3$ $\wedge$ Permutation($\{?i_1\ ?i_2\ ?i_3\}$, $ObjectsInWorld$) $\wedge$

ZeroExpression(ValueOf($F_{ext,?i_1,?c}$) + ValueOf($F_{ext,?i_2,?c}$) + ValueOf($F_{ext,?i_3,?c}$))

Figure 15. The Result of Standard Explanation-Based Learning

(line 10 of figure 6) leads to the requirement that the general objects be distinct.[8] The property of the specific example that each external force is zero is generalized. In the general case these forces need not *individually* be zero. What is needed is that they *sum* to zero. Line 15 of figure 6 produces this constraint because the integration rule used requires that the integrand be zero.

However, because the *structure* of the explanation is not generalized using standard generalization algorithms, the result obtained in this manner only applies to situations where there are *three* objects in the problem's world. A separate rule must be learned for situations containing four objects, five objects, etc. In addition, the acquired schema is not relevant when the external forces do not sum to zero. No appreciation of the effect of external forces on a system's momentum is obtained. By recognizing and analyzing obstacle cancellations, then reconstructing the explanation in the general case, **Physics 101** overcomes these shortcomings.

4.3. Learning Special-Case Schemata

One would expect that acquired schemata should be as general as possible so that they might each cover the broadest class of future problems. Indeed, explanation-based learning research has been primarily aimed at the acquisition of such maximally general schemata. However, an intermediate level of generalization is often appropriate. The class of intermediate generality schemata improves the performance of a system's problem solver by supplying ''appropriately general'' schemata instead of forcing the system to rely on its maximally general schemata. Storing both the maximally general schemata and the intermediate level schemata results in much improved efficiency with no loss of generality. This section describes how **Physics 101** produces intermediate level schemata, which are called *special cases*.

A major issue in explanation-based learning concerns the relationship between *operationality* and *generality* [7, 12, 20, 29, 32]. A schema whose relevance is easy to determine may only be useful in an overly-narrow range of problems. Conversely, a broadly-applicable schema may require extensive work before a problem solver can recognize its appropriateness. Most approaches to selecting the proper level of generality involve pruning easily-reconstructable portions of the explanation structure. In **Physics 101**, operational rules are produced by constraining a general schema in such a way that its relevance is easily checked. Special cases are the result of composing the new, general equation schema with a problem-solving schema. A successful composition results in a specialization which is guaranteed to work using the composed problem-solving technique. This frees the problem solver from performing the planning that would otherwise be required to elaborate the general schema to fit the current problem-solving episode. The system can, of course, always resort to its collection of maximally general schema when no special case is appropriate.

As illustrated by the result in figure 14, the explanation-based generalization of a sample collision problem leads to a physics formula that describes how external forces change a system's momentum. This general schema is broadly-applicable, but ascertaining that it will lead to the solution of a given problem requires a good deal of work. The external forces must be summed, then integrated, and the constant of integration must be determined by using some initial conditions. As will be seen, all of this work can be pre-packaged into a special-case schema, if assumptions about the external forces are made.

---

[8] This constraint does not appear in the result of **Physics 101** because it is a *known fact*.

The intermediate level schemata generated by **Physics 101** are similar in scope of applicability to those that human experts appear to possess. For example, the conservation of momentum problem results in a special-case schema characterized by the absence of external forces and the specification of a *before* and *after* situation. These features are those cited by experts as the relevant cues for the principle of conservation of momentum (see table 12 of [5]).[9]

Although the motivation for this intermediate level of generalization is computational, the use of this level helps to reconcile the approach with a variety of psychological evidence showing that problem solvers use highly specific schemata [11, 28, 37]. Much of expertise consists of rapidly choosing a tightly-constrained schema appropriate to the current problem. However, the difference between the knowledge of an expert and a novice cannot be explained on the basis of number of schemata alone. The scope and organization of these schemata have been shown in psychological experiments to be qualitatively different [5, 15, 28]. In representing a problem, novices make great use of the specific objects mentioned in the problem statement, while experts first categorize according to the techniques appropriate for solving the problem.

Physics 101's special-case algorithm is presented in figure 16. As in the basic model (figure 1), in the extended model a known problem-solving schema is initially used to understand a solution to a specific problem. The explanation-based analysis of the solution may lead to the construction of a new broadly-applicable schema. Interestingly, the generalization process often produces a new schema that, in its fullest form, is not directly usable by the originally applied problem-solving schema. For example, the acquired momentum law describes how the external forces affect a physical system's momentum. It is *not* a conservation law, although the original calculation involved the problem-solving schema for conserved quantities. Constraining the general result so that the originally-used problem-solving schema does apply produces a special case. In the special-case schema, the constrained schema, its constraints, and the original problem-solving schema are packaged together to produce a specialized equation schema.

Let *primary* = the primary problem-solving schema used in the specific example
equation = the newly-acquired equation schema
facts = specific facts about the current example

Step 1: Satisfy the preconditions of *primary* using *equation* and *facts*, producing a proof tree.

Step 2: Traverse the proof tree, locating subtrees whose root is a new equation (i.e., one not already known to the system).

Step 3: Generalize the resulting subtrees using a standard EBL algorithm, extracting the general version of the new special-case equation and its preconditions.

Figure 16. The Algorithm for Constructing Special-Case Equations

[9] It should be noted that it was not the explicit intent to model this psychological data. Rather, computational efficiency considerations led to a system that produced results matching this empirical data.

In the approach taken, automatically acquiring schemata of the intermediate level of generality requires that the system's schemata be organized into two classes:

(1) Schemata that represent general problem-solving knowledge, which apply across many application domains (e.g., a schema for utilizing a conserved quantity to solve a problem). In **Physics 101**, these are the *problem-solving schemata.*

(2) Schemata that represent declarative knowledge of the domain of application (e.g., Newton's laws). In **Physics 101**, these are the *equation schemata.*

Humans with mature problem-solving backgrounds possess the first type of schema and are told schemata of the second type when introduced to a new domain. Through study they acquire a large collection of schemata that combine aspects of both types, thereby increasing their problem-solving performance in the domain. Combining general problem-solving techniques with domain specific knowledge produces schemata that, when applied, lead to the rapid solution of new problems. This performance is modelled in the **Physics 101** system.

Special case concepts maintain pointers to the general concepts from which they arose and the properties that distinguish the special cases from their general cases are recorded. These special-case cues are checked first when selecting an appropriate schema during problem solving. If no special case is applicable, the general concepts are then accessed. Besides being constructed when the general case is acquired, a new special case can be created whenever the general case is used to solve a later problem.

In addition to improving a problem solver's efficiency, special cases also indicate good assumptions to make. For instance, if the values of the external forces are not known, assume they are zero, as this will allow solution to an otherwise impossible problem. Physics problems often require one to assume things like "there is no friction", "the string is massless", "the gravity of the moon can be ignored", etc. Problem descriptions given in textbooks contain cues such as these, and students must learn how to take advantage of them. Facts in the initial problem statement suggest possible problem-solving strategies, while any additional requirements of the special case situations indicate good assumptions to make (provided they do not contradict anything else that is known).

The schema used to solve the three-body collision problem is shown in figure 17. This schema makes use of conserved quantities during problem solving. It says that one way to solve for an unknown is to find an expression, containing the unknown, that is constant with respect to some variable, instantiate this expression for two different values of the variable, create an equation from the two instantiated expressions, and then solve the equation for the unknown. If the values of all but one variable at these two points are known, simple algebra can be used to easily find the unknown.

Recall that the result in figure 14 is *not* a conservation law. It describes how the momentum of a system evolves over time. Although this new formula applies to a large class of problems, recognizing its applicability is not easy. The external forces on the system must be summed and a possibly complicated differential equation needs to be solved. Applying this law requires more than using simple algebra to find the value of the unknown.

Preconditions

CurrentUnknown(*?unknown*) $\wedge$ ConstantWithRespectTo(*?expression*, *?x*) $\wedge$
SpecificPointOf($?x_1$, $?x$) $\wedge$ SpecificPointOf($?x_2$, $?x$) $\wedge$ $?x_1 \neq ?x_2$ $\wedge$
?leftHandSide = InstantiatedAt(*?expression*, $?x_1$) $\wedge$
?rightHandSide = InstantiatedAt(*?expression*, $?x_2$) $\wedge$
?equation = CreateEquation(*?leftHandSide*, *?rightHandSide*) $\wedge$
ContainedIn(*?unknown*, *?equation*)

Schema Body

SolveForUnknown(*?equation*, *?unknown*)

Figure 17. Conserved Quantity Schema

A portion of the proof that the originally used problem-solving schema (figure 17) can be used with the new general formula appears in figure 18. (Arrows run from the antecedents of an inference rule to its consequents.) In order for the conserved quantity schema to be applicable to this new formula, it must be the case that momentum be constant with respect to time. This means that the derivative of momentum be zero, which leads to the requirement that the external forces sum to zero. This requirement is satisfied in the specific solution because each external force is individually zero, and this property is used to characterize the special case. When this occurs, the momentum of a system can be equated at *any* two distinct states. The special case schema for momentum conservation is contained in figure 19.

$$\text{ConstantWithRespectTo}\Big(\sum_i \mathbf{M}_i\, \mathbf{V}_{i,\,?c(t)}\ ,\ \mathbf{t}\Big)$$

$$\frac{d}{dt}\sum_i \mathbf{M}_i\, \mathbf{V}_{i,\,?c(t)} = \mathbf{0}\ \frac{\mathbf{kg\,m}}{\mathbf{s^2}}$$

$$\frac{d}{dt}\sum_i \mathbf{M}_i\, \mathbf{V}_{i,\,?c(t)} = \sum_i \mathbf{F}_{i,\,ext,\,?c(t)} \qquad\qquad \sum_i \mathbf{F}_{i,\,ext,\,?c(t)} = \mathbf{0}\ \frac{\mathbf{kg\,m}}{\mathbf{s^2}}$$

IsaComponent(**?c**) $\wedge$
$\forall$ iε **ObjectsInWorld** NOT(ZeroValued($\mathbf{M}_i$)) $\wedge$
$\forall$ iε **ObjectsInWorld** IndependentOf($\mathbf{M}_i$, t)

$$\forall\, \mathbf{i}\ \ \mathbf{F}_{i,\,ext,\,?c(t)} = \mathbf{0}\ \frac{\mathbf{kg\,m}}{\mathbf{s^2}}$$

Figure 18. Satisfying the Second Precondition of the Conserved Quantity Schema

---

Equation

$$\frac{d}{dt}\sum_i M_i \, V_{i,?c}(t) = 0 \, \frac{kg\ m}{s^2}$$

Preconditions

IsComponent($?c$) $\wedge$
$\forall\, i \in ObjectsInWorld$ NOT(ZeroValued(M_i)) $\wedge$
$\forall\, i \in ObjectsInWorld$ IndependentOf(M_i , t)

Special Case Conditions

$\forall\, i \in ObjectsInWorld \quad F_{ext,i,?c}(t) = 0$

Problem Solving Schema Used

conserved-quantity-schema

Figure 19. The Special-Case Momentum Law

Physics 101's problem solving improvement after learning is reported in the remainder of this subsection. Performance on several collision problems, differing as to the number of physical objects involved, is measured before and after learning the concept of momentum conservation. The goal in each case is to determine the velocity of one of the objects after the collision. In all of the problems, there are no external forces, and sufficient, randomly-generated mass and velocity values are provided to make each problem soluble. Nothing is stated about the inter-object forces. As mentioned in the previous section, the standard system cannot solve collision problems before learning due to the incompleteness of its problem solver. However, to gather the data presented below, the problem solver was slightly extended. This was done by allowing as many ''blind'' substitutions as necessary. Ordinarily a search path terminates before two of these unmotivated substitutions would occur consecutively.

The time to solve problems as a function of the number of objects in the situation is graphed in figure 20. The solid line represents the system's performance without learning, while the dashed line represents its performance after learning. Without benefit of learning, the system uses only its mathematical knowledge and the physics equations of figure 4. The special case law for momentum conservation is used to solve the problem after learning. All points result from averaging five measurements, and the choice of the number of objects is made randomly in order to reduce the effect of inter-sample dependencies. The standard deviation for each point is less than 10% of the solution time. It takes about 280 seconds to learn the general momentum evolution law and its special-case concept of momentum conservation from the teacher's solution to the three-body collision problem.

This graph demonstrates one of the main advantages of explanation-based learning, especially that of a system that can generalize number. Even when a problem solver has sufficient knowledge to solve a problem, an exponential amount of time may be needed, making solution infeasible. The increase of efficiency provided by grouping inter-related schemata brings previously intractable classes of problems within the capabilities of the problem solver.

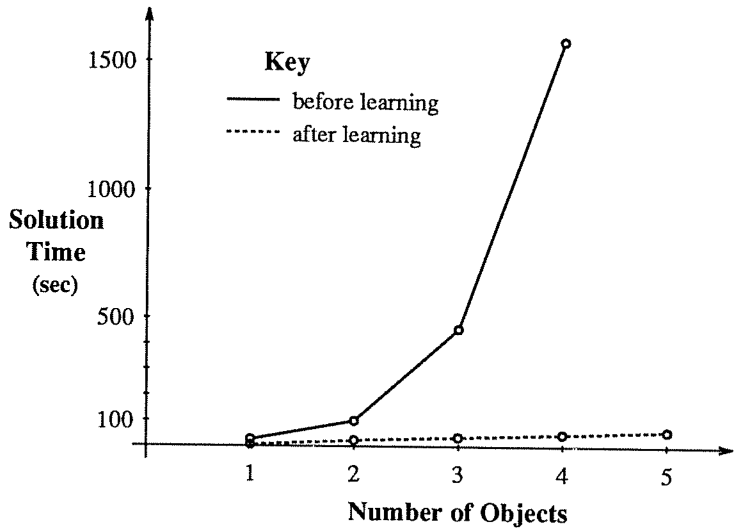

Figure 20. Performance on N-Body Collision Problems (averaged over five runs)

5. CONCLUSION

Physics 101 is a mathematical reasoning system that performs explanation-based learning in mathematically-oriented domains. This system's understanding and generalization processes are guided by the manner in which variables are cancelled in a specific problem. Attention focusses on how *obstacles* are eliminated in the specific problem. Obstacles are variables that preclude the direct evaluation of the unknown. Cancelling these variables allows the determination of the value of the unknown. The explanation of a specific calculation closely guides the construction of a general version of the calculation, from which a new general concept is extracted. New schemata are only learned when all of the obstacles in the general version of a problem can be eliminated.

Physics 101 presents a different perspective on the process of explanation-based generalization. Rather than directly using the explanation of the specific problem's solution as is done in more standard algorithms, the explanation of a specific calculation closely guides the construction of a general version of the calculation, from which a new general concept may be extracted. The new calculation is often substantially more general, in terms of its structure as well as its variables, than the specific calculation. Special cases of the generalized calculation, which can be more efficiently used during problem solving, are also constructed.

Because explanation-based learning requires extensive domain knowledge, it clearly is not appropriate for all learning in a new domain. Nevertheless, it may be useful even in early learning if the domain relies heavily upon a domain for which the learned does have substantial knowledge. Because mathematics underlies many other domains, a learner with some mathematical sophistication may be able to make use of explanation-based techniques without extensive knowledge of the new domain.

The foundation of mathematics underlies many application domains. In such domains, a kind of domain independence can be achieved by formalizing mathematical reasoning and strictly separating the system's initial application domain knowledge from its mathematical knowledge. **Physics 101** offers a conceptual vocabulary together with algorithms using that vocabulary to support explanation-based learning in mathematical domains. The concepts of primary and secondary obstacles and the notion of a cancellation graph are demonstrated in the domain of elementary Newtonian physics. But since these concepts are rooted in mathematics, they apply equally well in many other technical domains. Associated algorithms that construct solutions to new problems, explain a teacher's solution, and generalize explanations into new concepts are also mathematically centered and, therefore, independent of application domain. Together, these algorithms demonstrate how a system can acquire new application domain concepts (such as conservation of momentum) which greatly enhance problem-solving capabilities in the application domain.

ACKNOWLEDGMENTS

Comments by two anonymous reviewers greatly improved this paper. This research was partially supported by the Office of Naval Research under grant N00014-86-K-0309, by the National Science Foundation under grant NSF IST 85-11542, by a University of Illinois Cognitive Science/Artificial Intelligence Fellowship, and by a grant from the University of Wisconsin Graduate School.

APPENDIX A - SAMPLE MATHEMATICAL KNOWLEDGE IN PHYSICS 101

Below are some of the general mathematical rewrite rules known to the system. Note some formulae belong to more than one category (figure 3), depending on use. Terms beginning with a question mark are universally quantified variables.

$?expression - ?expression = 0$

Problem-Solving Schema: SubstAddIdentities

$?expression + 0 = ?expression$

Problem-Solving Schema: RemoveIdentities

$?expression \, / \, ?expression = 1$

Problem-Solving Schema: SubstMultIdentities
Preconditions: NOT(ZeroValued($?expression$))

$1 * ?expression = ?expression$

Problem-Solving Schema: RemoveIdentities

$\int (?independent * ?expression) \, d?x = ?independent \int ?expression \; d?x$

Problem-Solving Schema: ConstantsOutOfCalculus
Preconditions: IndependentOf($?independent$, $?x$)

$\int (?expression_1 + ?expression_2) \, d?x = \frac{d}{d?x} ?expression_1 + \frac{d}{d?x} ?expression_2$

Problem-Solving Schema: SeparateCalculus

$\int ?expression_1 \, d?x + \int ?expression_2 \, d?x = \int (?expression_1 + ?expression_2) \, d?x$

Problem-Solving Schema: CombineCalculus

$\frac{d}{d?x} ?expression^{?n} = ?n \; ?expression \; \frac{d}{d?x} ?expression^{?n-1}$

Problem-Solving Schema: Differentiate
Preconditions: Number($?n$)

$\int ?expression \; d?x = ?independent * ?x + constant$

Problem-Solving Schema: Integrate
Preconditions: IndependentOf($?independent$, $?x$)

APPENDIX B - GENERALIZING THE SOLUTION TO THE SAMPLE MOMENTUM PROBLEM

This appendix provides the details of the derivations of the equations in section 4.1. These calculations are the general version of the solution to the three-body collision problem. The cancellation graph in figure 9 guides the derivation. The figures in this and the next appendix are verbatim transcriptions of actual outputs of the implemented **Physics 101** system. The numbers associated with each line refer to the calculation steps of figure 6.

First, the primary obstacles are introduced into the calculation.

$$V_{?s,?c}(t)$$

(1) $= \int A_{?s,?c}(t)\,dt$

(2) $= \int \frac{F_{net,?s,?c}(t)}{M_{?s}}\,dt$

(3) $= \frac{1}{M_{?s}} \int F_{net,?s,?c}(t)\,dt$

(6) $= \frac{1}{M_{?s}} \int (F_{ext,?s,?c}(t) + F_{int,?s,?c}(t))\,dt$

(7) $= \frac{1}{M_{?s}} \int (F_{ext,?s,?c}(t) + \sum_{\substack{j \in ObjectsInWorld \\ j \neq ?s}} F_{j,?s,?c}(t))\,dt$

Next $\mathbf{M_{?s}}$ is introduced into the calculation to additively isolate the primary obstacles.

$$V_{?s,?c}(t)$$

(1) $= \int A_{?s,?c}(t)\,dt$

(2) $= \int \frac{F_{net,?s,?c}(t)}{M_{?s}}\,dt$

(3) $= \frac{1}{M_{?s}} \int F_{net,?s,?c}(t)\,dt$

(6) $= \frac{1}{M_{?s}} \int (F_{ext,?s,?c}(t) + F_{int,?s,?c}(t))\,dt$

(7) $= \frac{1}{M_{?s}} \int (F_{ext,?s,?c}(t) + \sum_{\substack{j \in ObjectsInWorld \\ j \neq ?s}} F_{j,?s,?c}(t))\,dt$

Following this, the cancellers of the primary obstacles are introduced.

$$M_{?s}\ V_{?s,?c}(t) + \sum_{\substack{j \varepsilon ObjectsInWorld \\ j \neq ?s}} M_j\ V_{j,?c}(t)$$

$$(1) \quad = M_{?s} \int A_{?s,?c}(t)\,dt + \sum_{j \neq ?s} M_j \int A_{j,?c}(t)\,dt$$

$$(2) \quad = M_{?s} \int \frac{F_{net,?s,?c}(t)}{M_{?s}}\,dt + \sum_{j \neq ?s} M_j \int \frac{F_{net,j,?c}(t)}{M_j}$$

$$(3) \quad = \frac{M_{?s}}{M_{?s}} \int F_{net,?s,?c}(t)\,dt + \sum_{j \neq ?s} \frac{M_j}{M_j} \int F_{net,j,?c}(t)\,dt$$

$$(4) \quad = 1 \int F_{net,?s,?c}(t)\,dt + \sum_{j \neq ?s} 1 \int F_{net,j,?c}(t)\,dt$$

$$(5) \quad = \int F_{net,?s,?c}(t)\,dt + \sum_{j \neq ?s} \int F_{net,j,?c}(t)\,dt$$

$$(6) \quad = \int (F_{ext,?s,?c}(t) + F_{int,?s,?c}(t))\,dt + \sum_{j \neq ?s} \int (F_{ext,j,?c}(t) + F_{int,j,?c}(t))\,dt$$

$$(7) \quad = \int (F_{ext,?s,?c}(t) + \sum_{j \neq ?s} F_{j,?s,?c}(t))\,dt + \sum_{j \neq ?s} \int (F_{ext,j,?c}(t) + \sum_{k \neq j} F_{k,j,?c}(t))\,dt$$

The primary obstacles are then cancelled.

$$(9) \quad = \int (F_{ext,?s,?c}(t) + \sum_{j \neq ?s} F_{j,?s,?c}(t) + \sum_{j \neq ?s} [F_{ext,j,?c}(t) + \sum_{k \neq j} F_{k,j,?c}(t)])\,dt$$

$$(10) \quad = \int (F_{ext,?s,?c}(t) + 0\,\frac{kg\ m}{s^2} + \sum_{j \neq ?s} [F_{ext,j,?c}(t) + \sum_{k \neq j,?s} F_{k,j,?c}(t)])\,dt$$

Finally, the secondary obstacles - introduced during the cancellation of the primary obstacles - are eliminated.

$$(10) \quad = \int (F_{ext,?s,?c}(t) + 0\,\frac{kg\ m}{s^2} + \sum_{j \neq ?s} [F_{ext,j,?c}(t) + 0\,\frac{kg\ m}{s^2}])\,dt$$

$$(11) \quad = \int (F_{ext,?s,?c}(t) + \sum_{j \neq ?s} F_{ext,j,?c}(t))\,dt$$

References

1. Ahn, W., Mooney, R. J., Brewer, W. F. and DeJong, G. F., "Schema Acquisition from One Example: Psychological Evidence for Explanation-Based Learning," *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, WA, July 1987, 50-57.

2. Bennett, S. W., "Approximation in Mathematical Domains," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, 239-241.

3. Brown, J. S. and VanLehn, K., "Repair Theory: A Generative Theory of Bugs in Procedural Skills," *Cognitive Science 4*, (1980), 379-426.

4. Bundy, A., Byrd, L., Luger, G., Mellish, C. and Palmer, M., "Solving Mechanics Problems using Meta-Level Inference," in *Expert Systems in the Micro-Electronic Age*, Michie, D. (ed.), Edinburgh University Press, Edinburgh, Scotland, 1979, 50-64.

5. Chi, M. T., Feltovich, P. J. and Glaser, R., "Categorization and Representation of Physics Problems by Experts and Novices," *Cognitive Science 5*, 2 (1981), 121-152.

6. Cohen, W. W., "Generalizing Number and Learning from Multiple Examples in Explanation-Based Learning," *Proceedings of the Fifth International Conference on Machine Learning*, Ann Arbor, MI, June 1988, 256-269.

7. DeJong, G. F. and Mooney, R. J., "Explanation-Based Learning: An Alternative View," *Machine Learning 1*, 2 (1986), 145-176.

8. Ellman, T., "Generalizing Logic Circuit Designs by Analyzing Proofs of Correctness," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, 643-646.

9. Forbus, K. D. and Gentner, D., "Learning Physical Domains: Toward a theoretical Framework," in *Machine Learning: An Artificial Intelligence Approach, Vol. II*, Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (ed.), Morgan-Kaufmann, Los Altos, CA, 1986, 311-348.

10. Greiner, R., "Learning by Understanding Analogies," *Artificial Intelligence 35*, 1 (1988), 81-126.

11. Hinsley, D. A., Hayes, J. R. and Simon, H. A., "From Words to Equations: Meaning and Representation on Algebra Word Problems," in *Cognitive Processes in Comprehension*, Just, P. A. Carpenter and M. A. (ed.), Lawrence Erlbaum and Associates, Hillsdale, NJ, 1977, 89-105.

12. Keller, R. M., "Defining Operationality for Explanation-Based Learning," *Artificial Intelligence 35*, 2 (1988), 227-241.

13. Laird, J., Rosenbloom, P. and Newell, A., "Chunking in Soar: The Anatomy of a General Learning Mechanism," *Machine Learning 1*, 1 (1986), 11-46.

14. Langley, P., "Data-Driven Discovery of Physical Laws," *Cognitive Science 5*, 1 (1981), 31-54.

15. Larkin, J. H., McDermott, J., Simon, D. P. and Simon, H. A., "Models of Competence in Solving Physics Problems," *Cognitive Science 4*, 4 (1980), 317-345.

16. Lenat, D. B., "AM: An Artficial Intelligence Approach to Discovery in Mathematics as Heuristic Search," Ph.D. Thesis, Department of Computer Science, Stanford University, Stanford, CA, 1976.

17. Mitchell, T. M., Utgoff, P. E. and Banerji, R., "Learning by Experimentation: Acquiring and Refining Problem-solving Heuristics," in *Machine Learning: An Artificial Intelligence Approach*, Mitchell, R. S. Michalski, J. G. Carbonell, T. M. (ed.), Tioga Publishing Company, Palo Alto, CA, 1983, 163-190.

18. Mitchell, T. M., "Learning and Problem Solving," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 1983, 1139-1151.

19. Mitchell, T. M., Mahadevan, S. and Steinberg, L. I., "LEAP: A Learning Apprentice for VLSI Design," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985, 573-580.

20. Mitchell, T. M., Keller, R. and Kedar-Cabelli, S., "Explanation-Based Generalization: A Unifying View," *Machine Learning 1*, 1 (1986), 47-80.

21. Mooney, R. J. and Bennett, S. W., "A Domain Independent Explanation-Based Generalizer," *Proceedings of the National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986, 551-555.

22. Mooney, R. J., "Generalizing the Order of Operators in Macro-Operators," *Proceedings of the Fifth International Conference on Machine Learning*, Ann Arbor, MI, June 1988, 270-283.

23. Mostow, D. J., "Machine Transformation of Advice into a Heuristic Search Procedure," in *Machine Learning: An Artificial Intelligence Approach*, Mitchell, R. S. Michalski, J. G. Carbonell, T. M. (ed.), Tioga Publishing Company, Palo Alto, CA, 1983, 367-404.

24. Novak, G. S., "Computer Understanding of Physics Problems Stated in Natural Language," Technical Report NL-30, Ph.D. Thesis, Department of Computer Science, University of Texas at Austin, March 1976.

25. O'Rorke, P. V., "Explanation-Based Learning Via Constraint Posting and Propagation," Ph.D. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, January 1987. (Also appears as UILU-ENG-87-2239, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

26. Palmer, M. S., "Inference-Driven Semantic Analysis," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August 1983, 310-313.

27. Prieditis, A. E., "Discovery of Algorithms from Weak Methods," *Proceedings of the International Meeting on Advances in Learning*, Les Arcs, Switzerland, 1986, 37-52.

28. Schoenfeld, A. H. and Herrmann, D., "Problem Perception and Knowledge Structure in Expert and Novice Mathematical Problem Solvers," *Journal of Experimental Psychology: Learning, Memory, and Cognition 8*, 5 (1982), 484-494.

29. Segre, A. M., "On the Operationality/Generality Trade-off in Explanation-Based Learning," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987.

30. Shavlik, J. W. and DeJong, G. F., "Building a Computer Model of Learning Classical Mechanics," *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, Irvine, CA, August 1985, 351-355.

31. Shavlik, J. W. and DeJong, G. F., "A Model of Attention Focussing During Problem Solving," *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, Amherst, MA, August 1986, 817-822.

32. Shavlik, J. W., DeJong, G. F. and Ross, B. H., "Acquiring Special Case Schemata in Explanation-Based Learning," *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, WA, July 1987, 851-860.

33. Shavlik, J. W. and DeJong, G. F., "BAGGER: An EBL System that Extends and Generalizes Explanations," *Proceedings of the National Conference on Artificial Intelligence*, Seattle, WA, July 1987, 516-520.

34. Shavlik, J. W. and DeJong, G. F., "An Explanation-Based Approach to Generalizing Number," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, 236-238.

35. Shavlik, J. W., "Generalizing the Structure of Explanations in Explanation-Based Learning," Ph.D. Thesis, Department of Computer Science, University of Illinois, Urbana, IL, January 1988. (Also appears as UILU-ENG-87-2276, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

36. Silver, B., *Meta-Level Inference*, North-Holland, New York, NY, 1986.

37. Sweller, J. and Cooper, G. A., "The Use of Worked Examples as a Substitute for Problem Solving in Learning Algebra," *Cognition and Instruction 2*, 1 (1985), 59-89.