Computational Models of Physics Problem Solving

Christopher Truxaw*

IRVINE COMPUTATIONAL INTELLIGENCE PROJECT
DEPARTMENT OF INFORMATION AND COMPUTER SCIENCE
UNIVERSITY OF CALIFORNIA, IRVINE 92717

Technical Report 87-20

August 11, 1987

Abstract

Solving typical textbook physics problems, such as those found in books used in high school and first year college physics courses, involves several subtasks. These subtasks can be described in terms of the way the problem is represented. Computational approaches to physics problem solving can be distinguished by the subtasks that they address and the types of representations that they use. A general descriptive framework of physics problem solving in terms of five different types or levels of representation that can be used in understanding and solving a physics problem is presented. Six computer programs that investigate various aspects of physics problem solving are presented and compared against the general descriptive framework. This comparison against a common framework makes clear certain differences among the reviewed programs in terms of what subtasks each is addressing. Important issues not reflected in the framework and only briefly addressed in the paper include learning and organization of knowledge. The systems reviewed include Novak's ISAAC, Bundy, Byrd, Lueger, Mellish, and Palmer's MECHO, de Kleer's NEWTON, Larkin and Simon's ABLE, Shavlik and de Jong's PHYSICS101, and Larkin, Reif, Carbonell, and Cheng's FERMI. These systems were chosen because they explicitly deal with problems typical of beginning physics textbooks. Related work on naive physics and qualitative reasoning about physical mechanisms and processes is not addressed in this paper.

*Supported by a graduate fellowship from ONR.

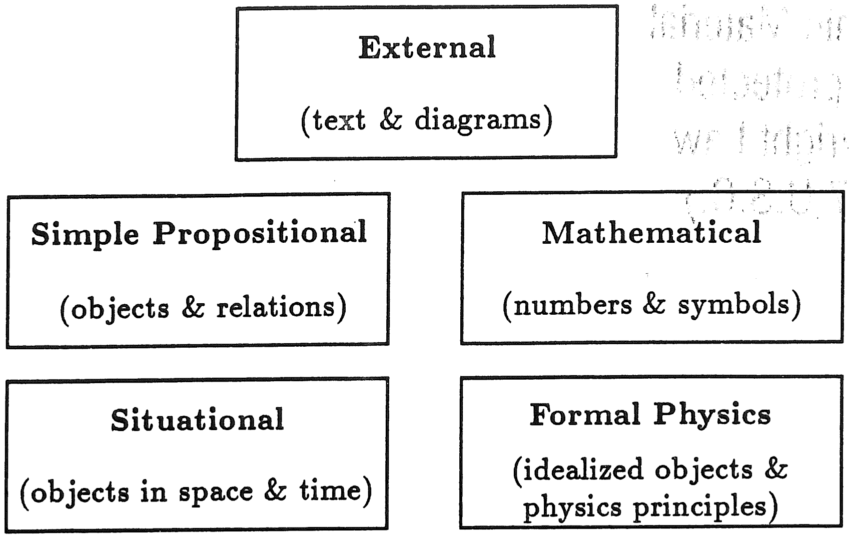


Figure 1: Problem representations for physics problem solving

1 Introduction

Successful solving of a typical textbook physics problem, such as might be encountered in the textbook for a high school or college introductory physics course, involves several steps. The written problem statement is read. The situation described in the problem statement is understood. The proper physics principles for interpretting the situation are chosen. Formulas are retrieved corresponding to the physics principles of the problem situation. Mathematical equations are set up based on the formulas and facts from the problem situation. These equations are solved and the answer is interpretted in terms of the physics principles and in terms of the objects in the problem situation and a statement of the answer is composed. Failure at any step may lead to backing up and reinterpretting one of the earlier steps. A strange result in the mathematics may lead the person solving the problem to rethink the formulas chosen, or to revise the physics principles applied, or to re-interpret the situation, or to re-read the text, or even to conclude that the text should be rewritten. Human problem solvers do not always follow all of these steps, but a competent human physics problem solver may have the ability to follow these steps.

Several attempts have been made to build computer models of physics problem solving. Each of these of these computational models addresses some but not all of the steps involved in human solving of textbook physics problems. We create a simple framework for comparing computational models of physics problem solving, and use this framework to compare six programs that solve physics problems.

2 A framework for physics problem solving

Figure 1 depicts a framework for describing examples of problem solving in the domain of physics. Problems can be represented in several different ways during problem comprehension and problem solving. This would often start with a written textbook form and go into other forms—eventually into mathematical equations—in order to reach a solution. Not every episode of physics problem solving would involve all of the levels of representation. For example, in some cases an answer could be generated without needing to use equations.

The stages in the problem representation that this framework considers include the *external* representation of text and diagrams, an internal *simple propositional* representation of the problem, perhaps based on the text comprehension, a *situational* problem representation based on qualitative inferences about the problem situation in time and space, a *formal physics* problem representation in terms of the idealized objects and principles of physics, and a *mathematical* problem representation in terms of numbers and symbols in equations and inequalities.

Processes must exist to transform information from one type of representation to another and to make inferences within a single representation. In the following sections we will describe and attempt to justify the breakdown of the physics problem solving process into these five stages or levels of representation.

A descriptive framework.

The purpose of this framework is to aid in describing instances of the physics problem solving process in computer programs and in humans. It is used to describe and compare existing A.I. physics problem solving systems, but it is partially based on analyses of human physics problem solving. A physics problem solving episode can be described in terms of which types of representation it uses.

Our choice of the five types of representation presented here are partly based on personal introspection about physics problem solving and partly based on descriptions in the literature of studies of human subjects solving physics problems [LMSS80a,RH82,Lar83,Lar81a,Lar81b]. They are also partly based on analysis of protocols of students working on problems in the related domain of algebra story problems [HKWT86], which indicate that human problem solvers often use other techniques besides setting up and solving equations. The problem solving process could be described in terms of a different set of representations. This set of representations is not the only way to describe physics problem solving and it certainly does not cover all types of representations that may be used in reasoning about physics problems, but it is does allow useful analysis of computational models of physics problem solving.

2.1 An Example: Five Levels

A physics problem solver that made use of all five of the representations that we are discussing, could potentially relate any type of representation with any or all of the others. Transformations between any two types of representation might be possible. For purposes of defining and explaining our five representations, we will use the example problem solving architecture depicted in Figure 2. In this figure the five representations are arranged in a sequence as levels, with each connected just to adjacent levels. Transformations are indicated in both directions between adjacent levels but not between any non-adjacent levels. This simplified example will be used to help explain the five types of representation. We are not proposing that this example architecture is the *correct* one. It is just an example for discussion purposes.

In this section we will examine each of the five levels of representation for physics problems depicted in Figure 2. We will try to define each level and discuss errors that could occur at

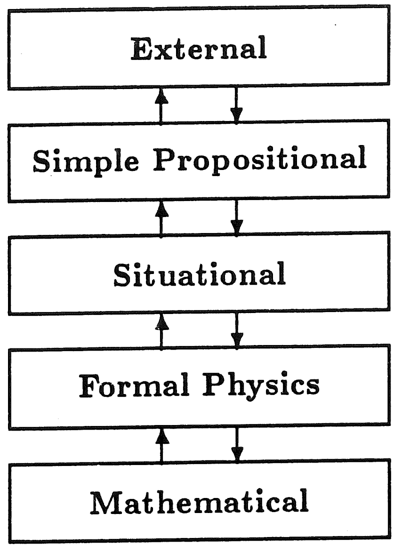


Figure 2: An example physics problem solving architecture

that level and possible means of reaching a solution to the problem at that level without going on to deeper levels. The sequence of levels was chosen because in the sequence each level is dependent on at least some of the levels before it, and not necessarily on any of the levels after it. For instance, the simple propositional representation depends initially on the external problem text, but not vice versa.

External Problem Representation—Problem Text

Textbook problem solving in any subject involves reading the problem from the textbook. The external written words of the problem statement make up this first level of representation. Some physics problems have to be solved by engineers and scientists without first appearing in written form in textbooks or on tests; but many problems do appear in this form when presented to a prospective problem solver and it is this type of problem that is of primary interest to us here. Diagrams and other pictures are often used in such problems, and any complete description of the external representation of physics problems would have to handle such pictorial representations as well as written language statements. If a solution is present at this level before working with any other levels, then the problem is an example rather than a problem to be solved. Errors initially present at this level would all be errors by the person who prepared the problems rather than by the problem solver, but a competent problem solver should be able to deal with (or at least recognize) misspellings, poorly worded problem statements, inappropriate diagrams, and the like.

Simple Propositional Problem Representation

When the problem text is read, the reader must construct an internal representation of the meaning of what was read. This level of the representation is meant initially to include just those propositions or relations which are explicitly stated in the written problem text without all of the inferences that can be made. It is possible to debate about just how much is explicitly stated and how much is inferred from background knowledge; but there has to be some internal representation that is just beyond the perceptual hardware before a lot of reasoning (especially conscious reasoning) takes place. A problem would only be directly solvable at this point if it is a problem that states its own answer. In this same simple propositional representation language many inferences may be possible that may lead to an answer, but this level does not include a simulation model of the situation. Errors that occur in the representation at this level correspond to a misreading of the problem statement.

Situational Problem Reprentation

The problem text is read and then, in order to fully comprehend it, a model of the situation described may be constructed. Many inferences are made based on the initial internal problem representation and related background knowledge and a qualitative situational understanding of the problem is built. It is problematic whether this elaborated problem representation is a separate level of representation from the simple propositional problem representation, or just an embellished version of the same level. We distinguish the two because the simple propositional problem representation is just a set of propositions or relations among objects, while the situational representation can be an qualitative model of the situation that can support simulations of the outcome of the situation, and has special expertise for reasoning about time and space. This can lead to a step by step simulation (usually through time) of some problems that can generate an answer without transforming the problem into abstract physics terms, or further into abstract mathematical terms. It may be possible to solve the problem at this level of representation if background knowledge can directly supply an answer or if a simple simulation of the situation can be performed based on background knowledge about this type of situation. Errors can occur in this stage of the representation if the background knowledge that is brought to bear is inappropriate or incorrect, which could cause an inaccurate simulation.

Physics Problem Representation

A problem can be translated from a model of a situation made up of "real" objects and into the abstract concepts and idealized objects of physics. Abstract physics concepts include such things as particles, forces, mass, energy, momentum, velocity, acceleration, gravitation, electromagnetism, electric potentials, and on and on. Some physics problems can be stated strictly in terms of such abstract physics concepts, and a problem has to be stated that way before the formal rules of physics will apply. The formal rules of physics don't say anything about cars, but they do say things about a particle with a certain mass and velocity and with forces acting upon it. If a car is viewed as a particle then such physics rules can apply. Once a physics problem is stated strictly in terms of abstract physics concepts, there may still be a lot of work to do to find a solution. The laws of physics may be sufficient for solving a physics problem given enough time to try the right combinations, but a lot of knowledge may be needed to properly apply the laws of physics in order to get to a solution efficiently. Errors could occur in this level of the representation if the physics knowledge that is applied is faulty, or if an improper choice of abstract physics concepts was made corresponding to the objects and relationships in the situational model. Some physics problems can be solved in this level

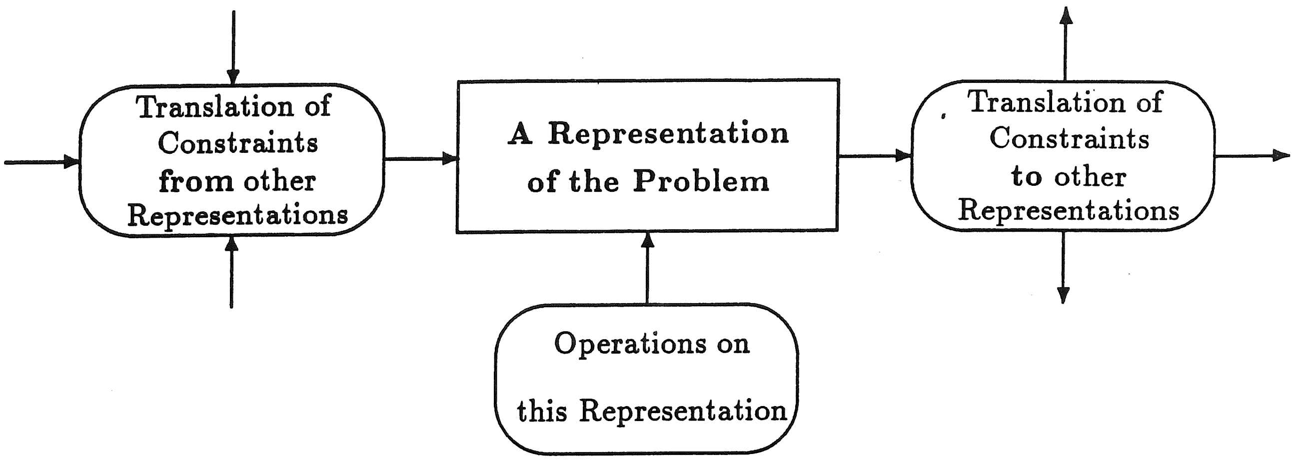


Figure 3: Transformations of a problem representation.

of representation by virtue of qualitative reasoning with the principles of physics, but often formal mathematical reasoning is required.

Mathematical Problem Representation

Physics problems encountered in high school and early undergraduate college courses most often require a quantitative solution that typically requires setting up equations and solving them with algebra or calculus. The equations that are chosen are based upon the physical situation of the problem, but the solution of the equations is done in the formal realm of mathematics. Once a problem has been transformed into a system of equations (and inequalities) it becomes possible to work on the problem as a formal mathematical exercise, and temporarily ignore the meanings of the formulas and variables. The power of mathematics allows solutions to many problems to be derived in this way at this level of representation. The results of this mathematical reasoning should be interpretted in terms of the physics concepts that provided the formulas, and further in terms of the "real" objects in the problem situation. Errors could occur in this level if the mathematical operations are improperly applied, or if the equations were not properly generated from the physical principles. Such errors should often be caught if the results are interpreted in terms of the other levels of representation. Students have been known occasionally to generate equations and solve them for an inappropriate numerical answer and fail to notice that the answer does not fit the situation.

2.2 Transformations of Problem Representation

If there are distinguishable stages or levels of representation that may be applied to physics problem solving, then there must also be some means of getting from one stage to another. Within the framework of Figure 2 we assume processes (represented by arrows) that can translate in either direction between any two adjacent levels. We also assume that processes (not shown) operate within each of the levels, elaborating and refining the problem representation without changing the language or terms in which it is expressed. See Figure 3.

For many types of problems it is possible to do successful problem solving while completely ignoring some of the levels. The STUDENT system by Bobrow [Bob68] and the CARPS system by Charniak [Cha68] used a purely syntactic analysis of the problem statements in algebra and calculus word problems to directly create equations without going through any of the possible intermediate levels of understanding the problems. These systems were able to solve a number of problems, but would fail if the problems were stated in a way that the programs could not map them into equations, such as if background knowledge had to be applied in order to set up some necessary equations. We maintain that any model of physics problem solving that skips over levels is limiting in its problem solving abilities.

The example framework of Figure 2 includes inverse transformations at every step, indicating that this is not limited to describing one-way processes. A problem solver can backtrack and revise its understanding at one level based on what it finds at another level. The abstract version of a problem in equation form is still related to the more concrete version, and insights at one level can be transferred to and evaluated at another. This framework allows for checking or verifying work. Taking the results from one level of reasoning and carrying them back to another to evaluate them is checking. Checking is appropriate only in circumstances where errors can be detected and possibly corrected. The transformations of a problem representation from one level to another will not result in a completely equivalent representation. Some information may be lost or gained in the transformation. Each representation level is a knowledge source for a distinct type of knowledge; therefore, the transformations between levels will not be a perfect mapping.

Between Words and Concepts

The first type of transformation of problem representation that a physics problem solver may have to address is at the point of input. The natural language understanding problem is one that must be overcome or avoided if the problem solver is even going to find out about a problem to work on. The problem solver has to have the ability to communicate with the external world in some form, at least to get a problem and to return an answer. Human problem solvers also use the external world for extending their memory and assisting in their reasoning by allowing them to use scratchpads and draw diagrams while they are working on a problem. A complete model of a physics problem solver ought to be able to handle the connection between external symbols and tokens and its own internal representations.

Between Simple Concepts and a Situation Model

Whether a problem was stated just in words or in words and pictures it often represents a complex physical situation that cannot be fully understood based on an understanding of just the words and pictures. Background knowledge about the objects and the type of situation described in the problem text must often be used to augment the information provided in the problem statement. Sometimes that background knowledge is sufficient to answer the question or solve the problem. A strong mental model of the situation may allow mental simulations that may provide additional constraints on the problem or even a solution. The problem solver should have the ability to apply background knowledge to elaborate the facts from the problem statement into a stronger mental model of the situation, and the problem solver should also have the ability to interpret its mental model in terms of the concepts and facts of the problem statement.

Between a Realistic Situation Model and a Formal Abstraction

A physics problem can be stated in terms of familiar objects and events or it can be stated in terms of abstract physics concepts or it can be stated directly in mathematical symbols and equations. (In the latter case you may have an algebra or calculus problem rather than a physics problem.)

Making the transition between a problem stated in terms of familiar objects and events, and a representation of the problem in terms of abstract physics concepts is not a trivial matter. Physics is supposed to be concerned with the real world, with finding regularities in it and with understanding how the universe works. It is also about abstract imaginary entities and forces that are not apparent to casual observers of the world. Its principles seem constrained to apply only in certain extreme conditions that generally do not occur. Frictionless surfaces or massless, inextensible strings are abstract concepts without physical realization. The only way to apply these physics concepts to the real world is to make simplifying assumptions and approximations. But the fact that the concepts of physics are abstract means that some very powerful, well-defined rules apply to those abstract objects.

When physics problems are stated in terms of the abstract objects instead of in terms of familiar objects then problem solving and reasoning can go on without any reference to any familiar real-world objects. There is still considerable skill to solving such problems, but they are formal problems like algebra or theorem proving. There is not necessarily a single abstract interpretation for a given familiar situation. It depends on the assumptions and approximations that you let yourself make. Deciding that the friction in a certain pulley can be ignored may be a useful simplification for solving a certain problem. Knowing when certain simplifying assumptions can be made and when they cannot is an important part of the knowledge of a physics problem solver. Knowing that the mass of strings in pulley problems can usually be allowed to default to zero is knowledge that a problem solver must have to be able to successfully solve such problems. Most if not all of the A.I. systems built to do physics problem solving essentially ignore or simplify the problem of mapping between a situation made up of familiar real world objects and an abstract situation made up of physics conceptual objects and principles.

Between Physics and Mathematics

A solution will usually require mathematical manipulation of equations produced by reasoning with the laws and principles of physics. Sometimes the answer to a problem can be determined just from qualitative reasoning with the principles of physics, but usually a solution requires quantitative reasoning. The principles of physics are often stated as equations or formulas. Once a problem has been understood in terms of abstract physics concepts, it is usually easy to come up with a number of equations that can apply to the problem. The goal is to be able to generate only a few appropriate equations, rather than every one that could apply, and for the choice of formulas to use to be based on a deep understanding of the problem in terms of the physical principles that it represents.

Other Transformations

Besides the forward transformations between levels of representation just described, similar transformations could be described in the backward direction. Although our diagram in Figure 2 shows connections only between adjacent levels, it is certainly easy to imagine transformations that go between nonadjacent levels. For simplicity in the diagram and in this discussion we did not include such connections.

2.3 Reasoning within a Representation

One of our criteria for defining a separate type of representation, is that there should be operations on this representation that can change the knowledge in this representation without reference to other levels. The center oval in Figure 3 represents such processes. This is in addition to the operations that transform knowledge from one representation into another (the right and left ovals in Figure 3). A type of representation that does not allow any operations on its contents, but only allows transformations to or from other types of representations is less interesting and could perhaps be eliminated or merged with another representation type.

If the five levels that we have outlined are legitimately distinct types of representation, then each should allow some reasoning within that level. A model of physics problem solving that does not allow reasoning at each of the representation levels can be considered incomplete. In this section we examine each of our five types of representation to see how they satisfy our requirement that they support reasoning.

Reasoning on Paper

The first level of representation in our framework is the external problem text. There may not be any reasoning that can be done purely with the external representation of the problem text, but some reasoning can be supported by making marks on paper. Paper is an extension to memory, and relationships among marks on paper can be perceived by the human problem solver giving insights into the problem. Diagrams come to mind as an example. By drawing a diagram a problem solver is sometimes able to discover relationships between objects in the problem that were not apparent without the diagram. Arithmetic algorithms often rely on paper and pencil as well. At its most extreme, external reasoning could involve solving a physics problem by physical experimentation—actually carrying out the actions of the problem to see what will happen. Any reasoning that relies on artifacts in the external world is generally ignored in computational models of physics problem solving. Two minor exceptions are Novak's [Nov76] ISAAC system which draws problem diagrams after the fact, and Larkin's [LMSS80b] ABLE system which used a working memory and a so-called "paper" memory to simulate internal and external memories.

Reasoning with Simple Propositions

The initial internal level of representation of the problem is made up of simple propositions that correspond to the problem statements. At the level of simple propositions, some reasoning can be accomplished. At this level it should at least be possible to notice if the problem states its own answer or to remember an answer if it was previously known. Perhaps logic can be applied, as well as some background knowledge in the same simple propositional form. We suggest that complex model-based reasoning is different and needs a different level of representation.

Reasoning with a Situation Model

The situational model level of representation should allow some sophisticated reasoning. This reasoning is often of the form of a simulation (usually over time) of some aspects of the problem to see what could happen. Sophisticated temporal and spatial reasoning occurs at this level. Background knowledge may provide constraints on the outcomes of events in the problem situation. The models in this level of representation may support thought experiments possibly leading to better understanding or solution of the problem.

Reasoning with Abstract Physics

Within the formal world of abstract physics concepts there is still a formidable problem of choosing the right principles to apply and the right order to apply them so that problems can be solved efficiently. An expert can recognize familiar patterns in the context that indicate that a certain principle, such as the conservation of energy can be usefully applied. Larkin has demonstrated that improvements in the application of formulas and principles can be achieved through composition of production rules, see Larkin and Simon [LS81]. This indicates that part of the difference between experts and novices is attributable to skill honed through practice, as opposed to raw factual knowledge. But for this practice effect to work, the problem solver has to know the basic laws, what they mean and how to apply them. A law may be stated in a simple formula, such as $F = ma$, but such simple statements hide a great deal of knowledge.

Some abstract physics concepts have strong connections to notions that people have about the everyday world. Velocity is related to and is easily confused with the everyday notion of speed. Other physics concepts may have good or bad analogs in common concepts. Thinking of electric current as if it were water in pipes is an analogy that is good for some aspects of electricity, but bad for others, see Gentner and Gentner [GG83]. In solving abstract problems it is likely that besides purely formal rules a problem solver would reason with analogies to other physical concepts and common sense concepts.

Reasoning with Mathematics

Most physics problem solving requires mathematical skills, at least at the level of algebra, and often calculus. Whether it is possible to have a correct qualitative understanding of physics principles without the supporting mathematics skills is doubtful. But it is clear that the laws and principles of physics are stated in the language of mathematics. Mathematics is a formally defined system that can be reasoned about in a formal symbolic way. The aspects of physics that are defined in mathematical formalisms can be reasoned about in the same way.

Mathematical reasoning can be used for various purposes in different stages of solving a problem. Sometimes mathematics may be used as a tool for carrying out some simple calculation about physics concepts, such as using arithmetic and a formula to calculate directly the value of an unknown, and other times portions of a problem may be translated into equations and a separate algebraic problem solving episode may take place.

Once a problem has been stated strictly in terms of formal mathematical statements, then there are formal rules that can apply to transform those statements (equations and inequalities) to try to reach a solution. There may still be a large number of ways to combine the available rules, or there may be no way of reaching a solution with the information given. Knowledge about strategies and tactics is needed for applying these rules to solve the formal mathematical problem efficiently. This formal mathematical reasoning process may be thought of as being completely contained in the formal rules of mathematics and the heuristics for applying those formal rules.

Comparing results from different representations

The various representation types are not completely equivalent and conclusions reached in one may not be appropriate when interpretted with respect to another representation type. The mathematics used in physics problem solving has a close relationship to the abstract physics concepts and principles that the formulas represent and it may be important to keep track of these relationships while doing formal mathematical operations. Unusual results in the mathematics may imply that the physics principles have been misapplied or that additional principles need to be applied, or they may at least require interpretation in terms of the physics concepts that they represent. Similarly, abstract physics concepts should not be completely divorced from the common objects that they represent. There are different levels on which a problem can be represented, but they are not completely independent.

In the next section we use the framework that we have just introduced as a basis of comparison to evaluate six computational models of physics problem solving.

3 Comparing Computational Approaches

The diagram of representations used in physics problem solving in Figure 1 does not capture many important aspects of physics problem solving, but it is useful as a framework for comparing and contrasting different computational approaches to the task. Computer programs that do physics problems solving can be distinguished based on how they fit this diagram.

Six computer programs that address some aspects of physics problem solving are illustrated in Figures 4, 6, 8, 9, 10, and 11, which show how they fit the general diagram of Figure 1. These illustrations allow a quick comparison of the systems in terms of the stages of the physics problem solving process that each one addresses. There may be other important aspects or emphases of each program that are not depicted in these diagrams, but the framework does provide some common terms for describing various systems and it is a point of departure for discussions about other aspects of each approach.

A model of physics problem solving can be compared with this framework and the following features of the model will immediately stand out.

- Which of the five types of representation the model handles.

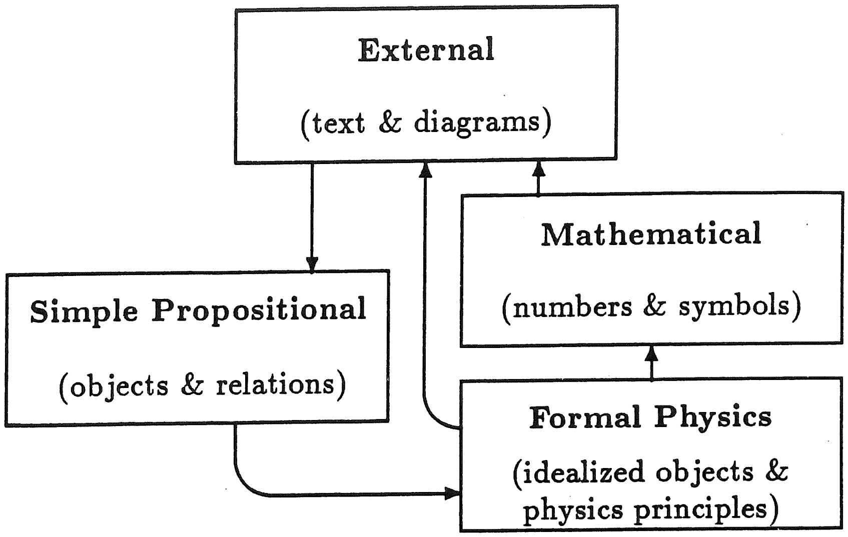


Figure 4: Four levels of problem representation in ISAAC.

- Which transformations between levels the model allows.

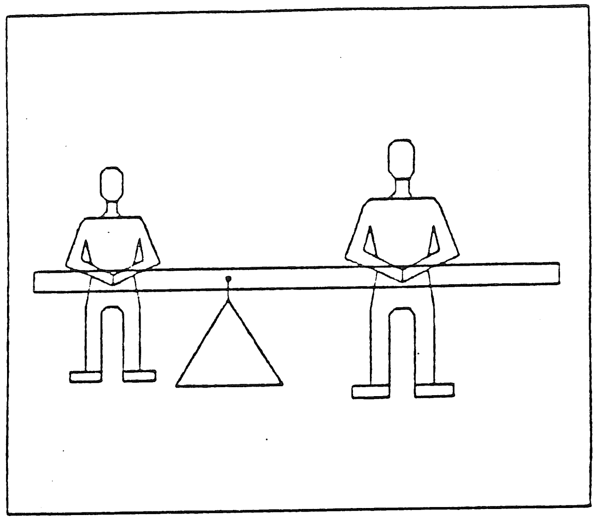
Some other points that are interesting to notice about models of physics problem solving but that are not illustrated in this framework include:

- Learning—whether and how it addresses issues of learning;
- Overall strategy—if it has a single overall strategy for solving problems, such as always using algebraic equations, or if it has a repertoire of methods to choose from;
- Knowledge structures—how knowledge is encoded and shared by various parts of the problem solving process;
- Performance—what types and difficulties of problems can the model solve;
- Psychological validity—whether it attempts to be a model of human abilities and how true a model it is.

The comparison between programs will chiefly focus on the way they fit the framework. A diagram is included for each one illustrating which parts of the framework the program addresses.

3.1 Novak's ISAAC

Novak's ISAAC system, as described in [Nov76,Nov77,NA80], deals with physics problems at four of the five levels of the general diagram introduced in this paper. English text is parsed into an initial problem representation which is translated into formal, or *canonical* physics concepts, which in turn help to generate a number of algebraic equations which are solved for an answer to the problem. The answer is given in English and a picture of the situation is also generated.



P7 SCHAUM PAGE 16 NUMBER 21

(A UNIFORM POLE 20 FT LONG AND WEIGHING 30 LB IS SUPPORTED BY A BOY 3 FT FROM ONE END AND A MAN 6 FT FROM THE OTHER END) (AT WHAT POINT MUST A 150 LB WEIGHT BE ATTACHED SO THAT THE MAN SUPPORTS TWICE AS MUCH AS THE BOY)

ANSWER: 7.40000 FT FROM THE BOY

Figure 5: Sample output from Novak's ISAAC system

There is no elaboration of the initial propositional problem representation other than the translation to canonical physics concepts. And there is no checking of work, nor reverse translation of results, aside from the formatting of the statement of the answer.

ISAAC does not address learning issues. It has a fixed overall strategy of deriving equations and solving them for an answer. It performs well on a very limited set of physics statics problems. The knowledge representation used varies from one part of the program to another. Semantic nets and frames with attached procedures are used. Tactical control knowledge generally is procedurally rather than declaratively encoded. The whole system appears to be a loose psychological model, but Novak does not make any strong claims about the psychological validity of the system.

Language understanding and problem transformation

The ISAAC program reads and solves high school or college level physics statics problems. It can be given problems in the exact words in which they are written in a physics book, and it understands them and solves them. It also draws a picture to illustrate the answer that it gives (see Figure 5). The picture is not used in solving the problem—it is drawn after the problem is solved—but it is an indication that there is more than just a superficial syntactic understanding of the problem text. The program builds a model of the situation.

In Novak's ISAAC program the problem representation goes through several stages. It starts out as an English problem statement. Syntactic parsing by an augmented transition net grammar (ATN) produces a case-structured semantic net representation for each sentence. Semantic processing is interwoven with parsing. Identification of referents of phrases and determination of types for the entities referenced are examples of semantic processing.

An internal model of the problem is incrementally built by the semantic processes. Specialist routines for each type of referent in the semantic nets help transform them into so-called *language-free* semantic frames for physical objects, features, and relationships. Novak claims that this model is language-free, and that it could be used for other purposes besides problem solving. For instance, it could be used as a basis for translation of the problem to another language, or for answering simple questions about the problem situation. He does not implement any of these claims. The semantic frames of the internal model represent physical objects and features of those objects and relationships among the objects. These semantic frames form the *Simple Propositional* representation of Figure 4.

A "Canonical Object Frame" must be associated with each physical object in the problem. This is an idealized or abstracted physical object such as a rigid body or a point mass. A canonical object is not an *isa* superset of a physical object, but rather it is a *view* of the object, in the sense of Bobrow and Winograd's KRL [BW77]. One object can be viewed in several ways or fill several different roles. A rock can serve the role of a chair or be viewed as a chair, or an automobile can serve the role of a point mass or be viewed as a point mass. This is not to say that all rocks are chairs or that all automobiles are point masses.

Procedures attached to semantic frames are used for setting defaults and abstracting problem features to match canonical problem solving requirements. Each physical object has a set of associated canonical objects *one* of which is chosen, e.g., a rigid body, or a point mass. Special functions map concrete physical to abstract canonical object frames (and vice versa for giving results in terms of the concrete objects in the problem). Certain features are required for a given canonical object, and these are identified with features of the physical object or else default values are set. Next a complete geometric model is built that has a single common coordinate system, instead of the many object-centered coordinate-systems in the earlier representations. A separate geometric problem solving program called EUCLID is called to take care of this task. The canonical object frames make up the physics problem representation in Figure 4. Procedures attached to the canonical objects generate equations using the geometric model. These procedures represent physical laws about the way canonical objects interact. From equations, an answer is generated with a small symbolic algebra package, and the result is displayed with a diagram of the problem situation. Special procedures arrange and scale pictures of objects and set up the proper form for the statement of the answer. The representation used for drawing the picture is similar to the geometric model, except that reasonable sizes for display have to be chosen for objects in the picture and reasonable choices have to be made for the points of attachment of objects. For example a man on a ladder should be attached to the ladder by his feet rather than by his ear.

Limitations and Strengths of ISAAC

Several limitations of the ISAAC program are worth noting.

- ISAAC does not have any mechanism for learning or improving its performance.
- Novak [NA80] has noted that ISAAC over-simplified the process of recognizing actual objects as instances of canonical objects, and making the mapping between a model of physical objects and the canonical object frames.

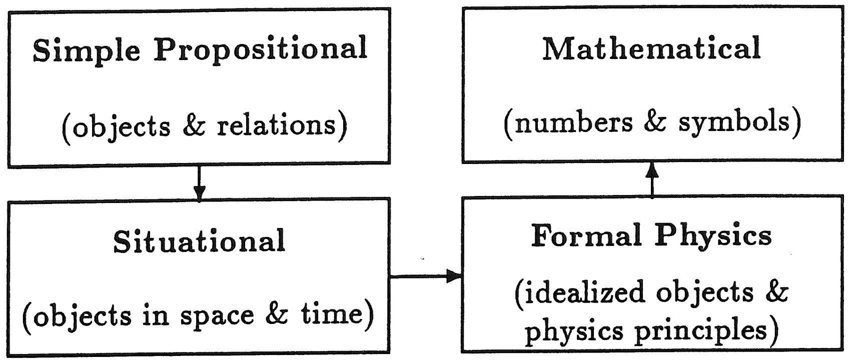


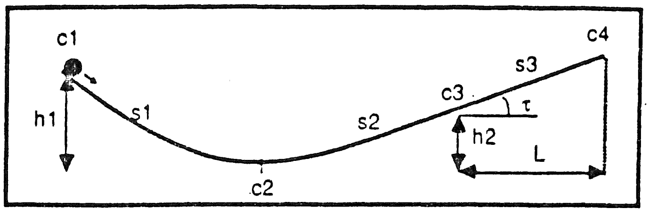
Figure 6: Four levels of problem representation in NEWTON.

- The system is locked in to a single strategy. Every problem is done by generating equations and solving them for an answer.
- The system generates many more equations than a human expert solving the same problems. It generates all related equations—nine to fifteen of them, instead of the one or two special case equations that a human expert would generate.
- There are no claims to psychological validity about the symbolic manipulation package that finds the algebraic solution, nor to the picture drawing functions that represent the problem and result pictorially.

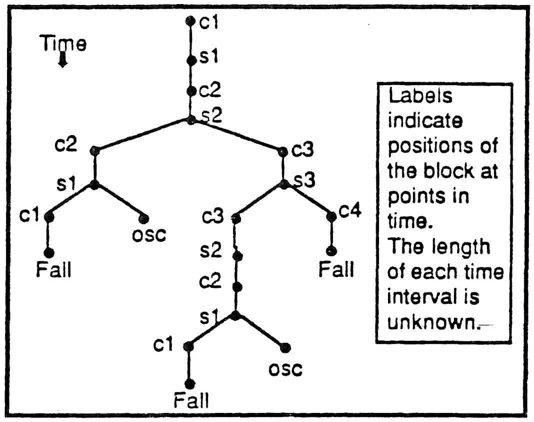
ISAAC is a performance system investigating aspects of problem comprehension and problem solving on a limited set of problems. The system pays a lot of attention to the problem of interpreting the English text input, and somewhat less attention to the problem solving that follows. It does address at least four of our five types of representation. In addition, the geometical model it generates could be considered to fit into the remaining type, a situational representation. We choose not to characterize it in that way because it does not support any kind of qualitative simulation. The idea that problem solving can involve repeated transformation of a problem representation is well-developed in this system.

3.2 de Kleer's NEWTON

de Kleer's NEWTON system, as described in [dK77,dK79], dealt with four of our five types of representation. English problem text is not handled, but all of the other levels are addressed. An initial propositional problem representation is provided to the program, and qualitative reasoning about this representation produces an elaborated situational representation (or *envisionment*) which may answer some questions directly, or may be used with physics principles to generate quantitative and mathematical representations of the problem and generate a solution algebraically.



A simple roller-coaster problem. The block starts at point c1.



NEWTON's Envisionment of the roller-coaster problem

Figure 7: The NEWTON system and Envisionment

Natural language processing is not addressed and no reverse transformation of results from any stage of representation in the problem to an earlier stage are included. Arithmetic calculations are not used in the elaboration of the initial problem description. The emphasis is on qualitative elaboration.

The NEWTON system does not address learning issues. It has some flexibility in its overall strategy, in that a problem may be solvable strictly based on qualitative analysis, or it may require quantitative analysis. Different knowledge representations apply to different steps in the problem solving process, including an envisionment state tree in the qualitative analysis and frames in the quantitative analysis and equations for the mathematical reasoning part. This early work only handled roller coaster problems, but the techniques have been extended to other complex situations. The work is definitely motivated by interests in modelling human cognitive abilities, but it is not closely tied to supporting psychological evidence.

Envisioning

The NEWTON program is able to solve physics problems in the restricted area of roller coaster problems. Part of the problem solving process involves making a qualitative evaluation of the situation described in the problem. For some problems the qualitative evaluation is enough to answer the question without doing any quantitative reasoning and for others the qualitative evaluation sets up the problem for a quantitative solution. Envisioning is a process whereby all possible outcomes in a qualitatively uncertain situation are generated in the form of a tree based on gross features of the situation. For example, see Figure 7. The tree diagram represents the envisionment: the result of the envisioning of all of the possible chains of events resulting if a small block is released from point c1 in the roller coaster diagram and allowed to slide along the frictionless surface s1, The envisionment tree is labeled with positions of the block as it slides on the roller coaster. Two scenarios end in oscillation and three end with the block falling off one of the ends of the roller coaster track. Quantitative information, such as the heights of certain points on the track, and equations based on physics principles can be used in quantitative analysis to decide which of the envisioned outcomes will actually happen. Some questions about the situation could be answered from the envisionment directly, without any quantitative reasoning, such as, will the block come to rest and stay at point c3? The reasoning that goes on is based on an abstract situation of a frictionless roller coaster. Commonsense reasoning in a more realistic situation would give a different envisionment.

The NEWTON program

In setting up the NEWTON system, de Kleer wanted to address the issue that easier problems should be solvable using simpler or at least different methods than hard problems. A problem solver that can solve difficult calculus based problems should also be able to recognize when a problem has a trivial solution. He thought there should be multiple representations for the problem with different reasoning techniques for each. Easier problems should be solvable with easier techniques.

The NEWTON system has four levels in its processing of a problem: Envisioning, Qualitative, Quantitative, and Mathematical. The envisioning builds a tree of possibilities. It must be a tree and not a graph because it represents possible chains of events over time. A state is never repeated, even in an instance of oscillation, because each state has a distinct time. Each leaf of the envisionment tree identifies a complete chain of events rather than just a final state. The tree is reasoned about qualitatively to make a plan for reaching a solution to the problem. Quantitative knowledge is represented in frames and symbolic reasoning is used to try to plan a path to a solution. The quantitative reasoning may generate equations that are passed to a mathematical reasoner based on MACSYMA that solves the equations and returns a result.

This system does qualitative reasoning on a model of the situation, but the situation already is described in terms of ideal objects. It can answer certain questions from the qualitative understanding that it develops, but it does not do time-step simulations or otherwise calculate any numeric answers at this stage. Physics formulas are present in a frame-based representation. Each formula frame has certain conditions to test to see if it applies. The quantitative reasoning selects equations to use to reach a solution, and keeps track of variable assignments in the equations. Equations are passed to the algebra package and numeric or symbolic results are passed back from the algebra package to the quantitative reasoning level. There is no other communication between the mathematical reasoning and the quantitative reasoning.

Limitations and Strengths of NEWTON

de Kleer mentions some failings of the NEWTON program in [dK77]

- "an insufficiently powerful envisioner;" The situation of roller coaster problems do not involve very complicated envisionments and more complex problems would need a stronger envisioner.

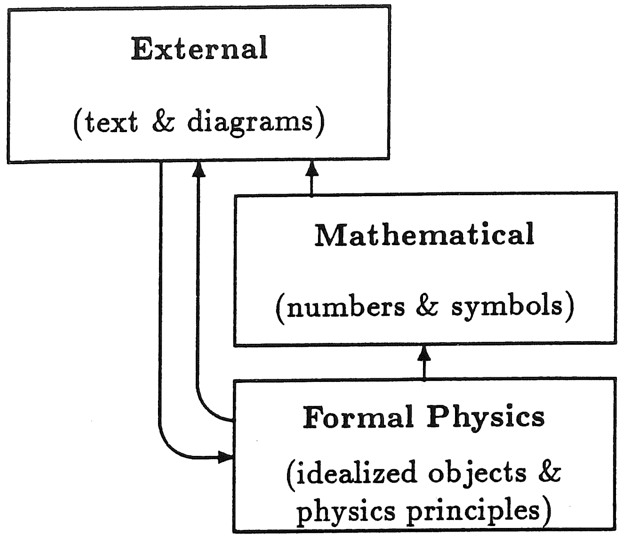


Figure 8: Three levels of problem representation in MECHO.

- "an inappropriate theory of mathematical expertise;" The black box MACSYMA mathematics routines do not provide the kind of interaction with quantitative knowledge that is necessary for properly solving some problems.

Some other limitations of the NEWTON program are worth noting.

- The system does not address any issues of *learning*.
- It does not address natural language comprehension.
- The translation to physics formalism is somewhat weak—the problem is stated in terms of standard objects not much different from idealized physics objects.
- The system does not seem to do checking of answers against earlier stages of problem representation.

The concept of *envisionment* has been refined and expanded in later work, such as [dKB81]. The NEWTON system is important because it represents an early attempt to show how physics problems can be solved in more than one way, choosing from multiple techniques that apply to different types of representation. In particular, it emphasizes the fact that qualitative reasoning alone can sometimes solve a problem, and quantitative reasoning depends on an understanding of the problem that is achieved in qualitative reasoning.

3.3 Bundy's MECHO

Bundy's MECHO system, as described in [BBL*79,BB83], is able to handle fairly difficult problems from a wide range of topics in mechanics, but it does this while working with only three of our five representation types, as illustrated in Figure 8. MECHO does natural language processing from English text, but it deals with problems that are already stated in terms of abstract physics concepts, so it skips directly to a physics problem representation. It then translates the problem into algebraic equations and performs the algebra or calculus needed to get a symbolic or numeric solution, which it returns.

Besides skipping the simple propositional problem representation and a situation model, this system does not do any reverse translation of results from any level of representation to an earlier level except for the output of an answer. It has a fixed overall strategy—always deriving and solving equations without considering any alternative strategies. It does have multiple approaches it can use to find equations. The knowledge that it uses is encoded in the predicate calculus of the PROLOG language, with an essentially declarative representation of both object level information about objects in the physics problem as well as *meta-level* information about the tactics and control strategy of the problem solving process. It does not address the issue of learning. The creators of this system do not make any claims about the psychological validity of the system, but they do have psychological aims in doing this research, hoping to better understand the process involved in getting from words to equations in order to improve the teaching of physics.

Meta-level Reasoning

The MECHO program is a physics problem solving system that its creators claim gets extra power from using *meta-level inference* in virtually every subtask of the problem solving process. By meta-level inference Bundy seems to mean declaratively encoding information and rules about any aspects of the problem solving process that do not directly involve objects of the physics problem. *Object-level* information refers to the objects in the physics problem, while meta-level information refers to everything else involved in the problem solving process.

In one way of looking at it, the meta-level information is the program itself, which looks like it is doing higher order reasoning about the problem solving process because it is written in PROLOG. The meta-level reasoning does not apply to choosing high level strategies for attacking the whole physics problem. Meta-level reasoning describes the processes applied to subtasks, such as how to parse the input text or carry out symbolic algebra or calculus. Tactics are selected based on the context in the particular subtask. The larger problem context does not affect the performance on a subtask. This use of the term *meta-level reasoning* may be misleading. It is clear that Bundy had to reason about reasoning in order to write his program, and he separated object-level rules from control rules in his program. But it is not so clear that his program is reasoning about reasoning.

The MECHO program

The MECHO system solves a wide range of mechanics problems. The domain or class of problems solvable by this system includes various mechanics problems with idealized objects such as light inextensible strings, frictionless pulleys, smooth planes. It solves statics problems, pulley problems, problems of motion on smooth complex paths and motion under constant acceleration. MECHO is constantly being extended to handle more problem types. One extension allows it to handle problems requiring the use of calculus over continuous measure systems, such as finding the radius of gyration of a spinning disk. But even in this extended version, the system fails to be able to solve many hard problems and avoids the "idealization" problem of choosing the proper ideal or abstract physical interpretations for objects in the problem.

The stages of problem solving in MECHO are divided into three major sections. These are:

1. interpreting the English text,

2. generating equations from the problem description, and

3. symbolically solving the equations for a solution.

MECHO has a limited ability to accept English text input, but can also accept PROLOG predicate calculus statements about a problem. When English text is accepted, it is parsed into predicate calculus assertions, but the parser cannot handle all of the problems that the problem solving part of the program can handle. Syntactic and semantic analysis converts English text into PROLOG predicate calculus statements about objects and their relevant properties and relationships, and what values are sought and what are given. The *meta-level* information in the parsing program represents techniques for rejecting false parses based on semantic constraints.

Problems are stated using idealized physics objects instead of "normal" objects, skipping over the problem of mapping from a real world situation to physics principles. Objects that may participate in the physics problems include simple zero and one dimensional objects which have types and properties and participate in relations. For example, particles, pulleys, spatial points, moments of time—all of type POINT; rods, strings, paths, and periods of time—all of type LINE.

The step of generating equations from the predicate calculus assertions of the problem description includes the use of schemata, meta-level information, inference rules, and physical formulas. A schema is a collection of facts and default values for quantities in a particular familiar type of problem situation. It contains the background knowledge needed to solve the problem that is not stated in the problem. A physical formula is just an equation that expresses a relation among abstract physics quantities, such as $F = m \times a$. Inference rules apply to object-level information and allow some new object-level conclusions to be drawn based on information from the problem statement or from previous inferences. Meta-level information is the encoding of the knowledge of the process of using schemata, inference rules, and physical formulas, as well as object level information. Meta-level information is supposed to allow a great reduction in search that would otherwise have to occur. In fact, the MECHO system does selectively generate a small number of useful equations, instead of producing every applicable equation as ISAAC does.

Cues for schemata are generated in the parsing of the English text by recognizing key words and certain object configurations, such as cueing a pulley-system schema when the problem includes a pulley, a string, and two solids, and the pulley supports the string, and the solids are attached to the string. A schema includes information that is necessary for solving the problem, but that is not stated in the problem, such as default values for mass of the pulley (=0) and facts such as "particles experience constant acceleration", or "the tension in both parts of the string are equal if there is no friction."

The last section of the system finds a symbolic solution to equations, either with algebra or with calculus, and produces a numeric or symbolic answer in the correct units. A symbolic answer is a mathematical expression containing symbols for quantities from the problem. The part of MECHO that does algebra and calculus reasoning is another system called PRESS. This subsystem was developed separately but fits nicely into the MECHO system in terms of its goals and methods. Most A.I. physics problem solving systems just borrow a symbolic algebra package that has nothing in common with the overall system, or skip the algebra altogether.

Problem solving in MECHO is a one-way process, where each step is assumed to be completely successful before going on to the next step, and no backtracking to earlier steps is necessary or possible. At the level of overall strategy, there is no meta-level reasoning. There is only one top-level strategy.

Virtually all aspects of the problem solving process are described in terms of meta-level reasoning. Bundy says that meta-level reasoning makes inferences from the database of PROLOG assertions more efficient. Meta-level reasoning generates only equations needed for solving the problem. In the algebra package, the meta-level rules suggest strategies such as to isolate a variable. In effect, the meta-level rules are setting subgoals. There are symbolic rewrite rules that can apply to try to reach these subgoals. Without the the meta-level rules setting these subgoals, there would be a combinatorial problem because of the many ways that the rewrite rules could be applied. Others might call all this *using heuristics*, rather than *meta-level reasoning*.

Limitations and Strengths of MECHO

Some limitations of the MECHO program are worth noting.

- There does not seem to be any sharing of background knowledge between the semantic part of the language understanding module and the part that sets up of equations, even though they deal with the same objects and relationships.

- This system does not address the issue of making the correct idealization of a situation and its objects to make a problem fit the solution paradigm. Objects are already idealized when they are given in the problem statement.

- This system does not address issues of learning.

- The comments about meta-level inference being central to this system seem to be a slightly contrived rationalization done after the fact. In [BBL*79] Bundy, et al admit that when they started doing the work they did not recognize the distinction between object-level and meta-level reasoning.

This work seems to be coming from a performance-centered perspective, and as such, they do achieve significant success. It seems to be the best performance system of any of the physics problem solving systems examined. It covers the widest range of problems and the hardest problems of any of the systems reviewed.

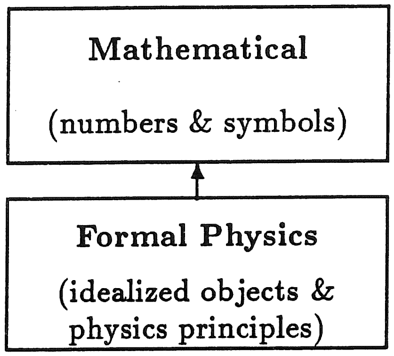


Figure 9: Two levels of problem representation in ABLE.

3.4 Larkin's ABLE

The ABLE system built by Larkin, as described in [LMSS80b,LS81,Lar81a,Lar81b], deals with only two of the five levels of problem representation. It deals with formal physics principles and does some of the reasoning for setting up equations. It does not actually set up or solve any equations. It merely keeps track of which quantities are unknown and which are known or derivable. It does no reverse translation of results. It focuses on the application of physics principles (or formulas) to a physics representation, and it *learns* to apply these principles in a more efficient fashion.

The ABLE system does address one aspect of the issue of learning—that of skill improvement in applying physics principles. The system encodes its physics knowledge in production system rules: condition–action pairs. It concentrates on the order of application of physics principles rather than actually solving physics problems. The work is presented as a valid psychological model based on evidence from psychological experiments, but it does not attempt to model the whole problem solving process, nor does it pretend to be a powerful performance system.

Improving skill in physics

The ABLE program solves certain physics problems and gets better at solving related problems with practice. The system is called 'barely ABLE' at first, when it knows the applicable laws of physics but does not have much knowledge about when to apply them. It develops into the 'more ABLE' system as it develops skill in using these laws.

The progression from barely ABLE to more ABLE is closely compared with differences between novice problem solvers and more expert problem solvers. Evidence is presented from analysis of protocols of human subjects solving physics problems, to support this comparison. The evidence involves changes in the order of application of various physics principles to solving a class of problems as the problem solver gets more experience. The change can be interpreted as indicating that the novice problem solver uses a backward chaining approach and the more experienced problem solver uses a forward chaining approach. Both work with the same set of physics principles, but the expert has organized and structured his/her knowledge of these principles to allow them to be successfully applied in a forward chaining fashion from the data in the problem, while the novice has to start with the desired unknown quantity and work backwards, finding principles that apply to find that quantity and then finding new principles to apply to find any new unknown quantities introduced. Larkin shows that the ABLE program is able to recreate this same effect with a simple computational model.

The ABLE program

ABLE is set up as an adaptive production system model of learning. Its knowledge is encoded in production rules—pairs of conditions and actions that should be carried out when the conditions are satisfied. The conditions of a production include the goal the production is working to solve. ABLE learns by adding new production rules that take precedence over existing rules because the new rules are more specific.

No algebraic solution is found, the system just keeps track of what is known and what is unknown. When exactly one symbol in a principle is unknown, then that symbol is marked as known, without representing its value in any way.

The barely ABLE system works backward from a desired quantity until it finds formulas for which all of the variables are known. The more ABLE system works forward, directly applying useful formulas to known quantities and deriving the desired answer more quickly. Of the various production rules involved in applying a general physics formula some propose the formula and others instantiate the variables in the formula, while still others solve the formula for an unknown. In the barely ABLE system a formula is proposed when it contains a variable for a quantity needed in the problem. It gets proposed before the other variables in the formula have known values. This causes other formulas to be proposed to solve for those quantities in a backward chaining fashion.

Solutions change gradually as the system learns specific productions to speed up the application of principles and reduce the backward-chaining search among principles. In this, as in any adaptive production system model, learning is the addition of new productions. New productions are produced when a principle is applied and generates new information. The actions taken to get the information and the conditions that made the actions possible are combined into the new production. This new production can fire if the same conditions arise again. These new productions are specific to these conditions. The new production can apply to any object that matches these conditions.

In the more ABLE system one very specific production will fire first based on the known quantities and the general desired quantity. This production will result in another quantity being known, and then another specific production will apply. The system will work forward until the desired quantity is known.

See Figure 9 for a picture of how the ABLE system fits into the general physics problem solving framework. Problems are given with variables identified to match specific physics formulas, avoiding language and common sense reasoning subtasks. There is no discussion of language parsing or other natural language issues. The quantities in the problem statement are labeled with the same names as the corresponding variables in physics formulas. The system does not have to do any interpretation of concrete objects into physics objects and variables. The problem is to find the right order of application of principles and to properly apply principles to the situation. In the barely ABLE system half of the productions fired involve

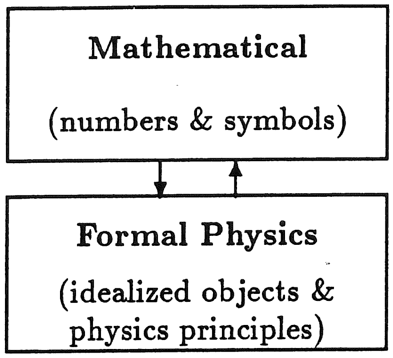


Figure 10: Two levels of problem representation in PHYSICS101.

interpreting equations in the current context, deciding if variables are known or desired or simply unbound.

Formal domains (geometry, physics) have "principles" that make problems solvable, but beginners do not have much knowledge about how and when to use these principles. They therefore end up using general search strategies. Conditions under which a principle is applicable may be taught, but often in an abstract unusable form. A simple mechanism of building new productions from actions taken in applying a single principle, may account for differences between novices and experts noticed in human subject protocols. Larkin suggests that composition of productions that apply different principles instead of just building specific productions that apply a single principle is a good idea, but that idea is not implemented in the ABLE system.

Limitations and Strengths of ABLE

- The ABLE system does learn, but it focuses on just one aspect of learning.
- It does not attempt to be a powerful performance system, and in fact it does not get answers. It just gets an ordered list of physics formulas to apply.
- It does not try to handle a variety of problem types.
- It ignores all aspects of problem understanding: text comprehension, problem elaboration or situational modelling, and problem abstraction into ideal physical objects.
- It ignores mathematical problem solving.

ABLE does illustrate the interesting effect of a change from backward to forward chaining based on the building of principle-specific production rules during practice.

3.5 Shavlik's PHYSICS101

Shavlik and de Jong described the PHYSICS101 system in [SD85,Sha85]. This system uses only two of our five types of representations. This program ignores natural language understanding, simple propositional reasoning, and situational modelling. It gets its problems in a formal physics form from the start. It applies principles that it knows about to generate equations and it reasons with calculus to solve them. If it does not know the principles needed to solve a problem it gets a solution from a tutor. It then reasons about the tutor's solution and generalizes to form new principles that the tutor must have used in his/her solution. It does translate in both directions between physics representation and mathematical calculus representation, but it does not address any other levels of the problem solving process at all.

The system focuses on a single method of learning, *i.e.* explanation-based generalization. The system does not attempt to be a powerful performance system. Its overall strategy for solving problems is based on generating equations and using calculus to solve them. Its calculus reasoning section is closely tied to the part that generates equations. These sections share knowledge and work together with the explanation generator to build new general principles that explain a teacher's example solution.

Learning new physics formulas from a teacher's example

The PHYSICS101 program does physics problem solving and learns to apply new principles, learning from an example given by a teacher using the technique of explanation-based learning. This technique was developed by de Jong and is more fully discussed in [DeJ81]. In explanation-based learning a learner has the basic axioms of a theory but needs help in making useful inferences from those axioms. It sees an example of a problem solution done by a teacher and it constructs an explanation of why the teacher's solution works. The explanation is a proof that the teacher's solution is valid based on the axioms. By generalizing the teacher's explanation the learner is able to make useful inferences, or build useful theorems, that make it able to solve a whole class of problems that it was previously unable to solve. The learner never attempts to complete an exhaustive search of the possible inferences from the axioms.

The PHYSICS101 program

The PHYSICS101 program has the basic physical laws that are necessary to solve every problem that it is presented with. It does not initially know how to apply those laws to solve a problem that it receives. When the PHYSICS101 program cannot solve a physics problem it asks the teacher to give a solution. Then it finds an explanation of why the teacher did it that way and generalizes this explanation to build new principles, such as learning about conservation of momentum. This builds new formulas. It does not introduce any new physical laws. Those are assumed known, because they must be known if the system is going to be able to explain the teachers solution. The system already knows about mass and velocity before it notices that their product, momentum, is a useful concept as well. The concept of momentum that it develops is strictly a new formula. It has no qualitative understanding of momentum. The explanation is of the form of a proof using the previously known physics laws as axioms. This system uses calculus in its attempts to solve problems and in its explanations of the sample solution given by the teacher. It distinguishes itself from some systems because it explicitly handles reasoning with calculus instead of relying on a black box mathematical manipulation package.

Figure 10 shows how this system fits the general physics problem solving framework. It is concentrated at one end of the process. There is no discussion of text comprehension. There is no discussion of a concrete situation model, nor of any common sense elaboration of the problem representation. There is no conversion into idealized physics concepts because the problem is apparently stated in terms of those concepts from the beginning, if not already in equation form. The whole emphasis of the problem solving part of this system is on the application of physics formulas to the problem representation to produce equations, and the calculus reasoning process on these equations to try to get a solution. The learning process is dependent on these interwoven capabilities.

Limitations and Strengths of PHYSICS101

The PHYSICS101 program has some limitations worth noting.

- This system concentrates on only one aspect of learning—that is, explanation-based learning. The ability to discover and prove significant principles from a single example is impressive, but it requires that the laws of physics be previously correctly known and it does not address how those are learned.
- It is not a powerful performance system, and it attempts to address only a narrow range of problems.
- It does not address natural language, nor problem elaboration.
- It does not deal with common objects, only abstract physics objects.

It does do some interesting reasoning with calculus, but its main reason for interest is because it can learn about new useful concepts, such as momentum, based on explaining and generalizing a teacher's example.

3.6 Larkin and Carbonell's FERMI

Larkin, Reif and Carbonell [LRC86] and Cheng and Carbonell [CC86] discussed two different versions of the FERMI system. This system does physics problem solving but apparently addresses only one of our five types of representation. Despite this lack of fit with our framework, the system is very interesting. This points out the fact that our framework is a high level one and it does not describe some important details of the physics problem solving process.

For FERMI the physics problem is directly encoded in an abstract physics representation that includes all of the solution methods needed to generate an answer. It emphasizes the organization of the knowledge in the physics problem representation and ignores all other levels of the problem solving task except techniques for grinding out an answer. In the second version of the system reported in [CC86] the learning of iterative macro-operators is an important

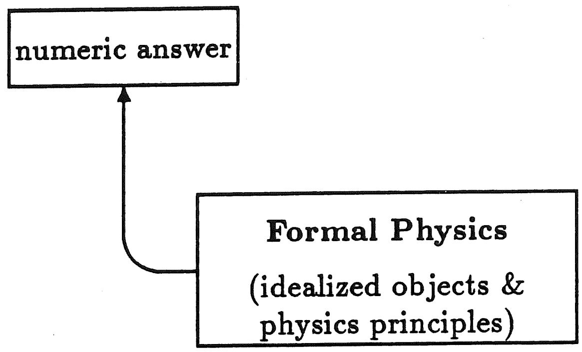


Figure 11: One level of problem representation in FERMI.

research topic. The system has been set up with three different high level control strategies: an "augmented means-ends method" which does backward chaining, analogical transfer, and rule-based forward chaining. The learning of iterative macro-operators is only discussed in terms of the rule-based method. The most detailed discussion of the system [LRC86] is in terms of the backward chaining method, and we will discuss FERMI primarily in terms of that control strategy.

Organized knowledge: concepts and methods

The FERMI program solves problems by traversing a hierarchical knowledge structure starting with the desired unknown and working until the desired unknown can be derived in terms of known quantities. Physics entities are represented in an object hierarchy linked to a hierarchy of solution methods. The methods include direct calculation if necessary related quantities are known, and decomposition of the problem otherwise. This version of FERMI works only with problems that are well-suited to a decomposition into parts.

The FERMI program

FERMI has its procedural and declarative knowledge organized in two major hierarchies: an action hierarchy that includes all methods and other actions involved in solving a physics problem, and an entity hierarchy that includes all objects and quantities that are involved in the representation of any physics problem. Each node in a hierachy is called a *frame* or *schema*. Each schema has a name and contains slots corresponding to features of that action or entity or relations of that node to other nodes in either hierarchy. Each hierachy is organized along the lines of inclusive class membership—an *isa* hierarchy. The methods refer to submethods and more primitive actions that are also in the same action hierachy. The two hierarchies are interconnected. The quantities in the entity hierarchy have slots that point to methods that can be used to find values for those quantities. The knowledge hierarchies feature automatic inheritance of all features from higher level concepts. Methods for classifying a problem are not discussed. All of the quantities and objects of a problem are directly assigned to specific classes in the entity hierarchy, such as a pressure drop.

The system solves physics problems in the domains of elementary DC circuits, planar center of mass calculations, and pressure drops in fluids. It has also been applied to solve linear independent equations in multiple unknowns using the same techniques. The authors of the FERMI system claim that the knowledge base is easily extensible to allow it to apply to other areas in physics. They say that other types of physics problems will use many of the same actions and entities. Only a few new ones will have to be added. They claim that the FERMI knowledge representation framework provides power and robustness, and cross-domain generality. The generalization hierarchy leads to transfer of knowledge between domains.

With the backward chaining control strategy, the problems to be solved must have one primary unknown designated as the desired quantity. The problem must fit into the entity hierarchy representation that exists. Units are not handled so the problem must be stated in values corresponding to the system's implied units. The solution methodology involves backward chaining from the desired quantity through the problem representation with backtracking to alternative methods when a chosen method fails. It does not set up equations, but uses formulas for direct calculation of results and for setting up subgoals. The solution is in the form of a scalar number without units. The methods used for solution often include decomposition of the problem into easier subproblems. General decomposition methods are used on various different problems. This exploration of decomposition methods is one of the most interesting features of the FERMI program.

The major intentions of the authors of the FERMI system seem to be directed towards building expert systems with greater problem solving and explanatory power using a generalization hierarchy and by separation of factual and strategic knowledge. They also expect their work to provide a hypothesis of how human experts structure knowledge and to improve the communication of knowledge (teaching systems) by appealing to a useful generalization hierarchy. In the latest work, they also show an interest in the learning of iterative macro-operators.

Iterative macro-operators in FERMI reduce repetitive sequences from a problem trace into a single operator that can include some internal conditionals. The work on this topic is just beginning, and it shows signs of producing some significant results.

Limitations and Strengths of FERMI

There are a few notable limitations of the FERMI system as it stands.

- The system does not use algebra to solve physics problems. The same architecture has been used to solve algebra problems, but not yet as a part of a physics problem. In physics problem solving no equations are set up. Arithmetic operations are directly encoded in the actions/methods hierarchy. It seems to be restricted to problems that have a numerical solution.

- It avoids the issue of understanding the text of a problem or doing any qualitative or conceptual elaboration.

- It does not deal with any issues of transformation or elaboration of a problem representation.

- This system addresses only a single specific type of learning, and none at all in its initial version.
- It is not yet a powerful performance system, despite the claims of its creators that its hierarchical knowledge base allows easy extension to to other domains.

This system is still in early stages of development. It has already achieved interesting results in knowledge organization and in learning of complex macro-operators. As the project matures, there should be additional interesting results.

3.7 Summary of Computational Models

Each of the computational models of physics problem solving that we have just reviewed has its own particular emphasis. ISAAC does natural language understanding and can build a geometric model of a problem situation as well as solve the problem in the area of statics. NEWTON uses the technique of *envisioning* the possible sequence of events to help understand simple rollercoaster motion problems. MECHO attempts to be a performance system that can solve many difficult problems and it also tries to do some natural language understanding. ABLE tries to account for certain differences between novice and expert performance by a form of skill learning based on composition of production rules. PHYSICS101 uses explanation based learning techniques to suggest a way that some physics concepts might be learned based on analysis of a teacher's example. FERMI illustrates a hierarchical organization of knowledge and a technique for learning macro operators to replace iterative sequences in a problem solving episode.

None of the computational models handled all of the levels of representation that we propose, but each of our levels is addressed by at least one system. Three of the systems emphasized certain aspects of learning. All but NEWTON, and in some sense FERMI, had a single fixed strategy for problem solving that was applied to every problem. NEWTON could solve some problems with qualitative reasoning, while others required quantitative reasoning. FERMI had various methods in its action hierarchy that might be applied to generate an answer. MECHO is the only one that attempts to be a performance system, and it is mildly successful, solving some reasonably difficult problems. It does not seem to be ready to be used as a tool for physicists or engineers. It can not handle really hard problems.

FERMI did not fit well into our framework. By our analysis it worked completely within the abstract physics representation. Its solutions used arithmetic, but it did not set up and solve equations. Despite the fact that its action fits into one little box on our diagram, it is still doing interesting problem solving. This points out that our framework does not look at details of problem solving. It breaks the physics problem solving task up into five major levels. The individual levels are very complex and interesting on their own. How the knowledge at each level is represented, organized, used, and learned are important questions that our framework does not address. In the following sections we will discuss these issues briefly.

Knowledge Representation and Organization

The knowledge a problem solver uses must be stored somehow and somewhere such that it can be accessed again in ways consistent with the problem solving process. Our descriptive framework said nothing about the knowledge representation, except that a problem representation can undergo tranformations and be expressed in more than one representation language, for example in a physics formalism as well as a calculus formalism. Is there a common store of knowledge that all subtasks draw on, or are there separate special rules and techniques for dealing with separate representations of different stages in the problem solving process? This is a question that our framework does not address, but some of the computational models do.

In the FERMI system the main emphasis was on the structured representation of knowledge about abstract concepts in the domain and about methods for determining quantities associated with them. All knowledge including knowledge of actions and methods was declaratively encoded in a hierarchical knowledge structure with a fixed procedure for interpreting the knowledge structure. ABLE on the other hand used a production system model for the procedural knowledge of the system. MECHO used PROLOG predicate logic for all of its knowledge. ISAAC used different representations at each level of processing but much of its knowledge was recorded in a frame-based manner with procedures attached to the frames. None of the systems seemed concerned with psychological memory issues such as explaining mechanisms for retrieval of information from memory.

People solving textbook problems do have a limited memory and have to resort to externally writing down notes about a problem. Drawing diagrams and using the paper as a memory aid and a calculation aid are issues that are not addressed in this framework. But they are involved in human physics problem solving.

Learning

Three of the systems that were described pay attention to some aspects of learning. All of these involve starting from a base of consistent laws or principles and learning better ways of applying the knowledge implicit in those laws or principles. Larkin's ABLE system composes production rules to speed up performance on application of formulas and in so doing changes the character of the solution steps from backward working goal-directed behavior to a forward working more expert behavior. Shavlik's PHYSICS101 system learns new principles that are derivable from laws and principles that it already knows. It learns in the situation where it has been unable to solve a problem, and it gets a sample solution from a teacher. It uses explanation-based learning to generalize the sample solution and come up with general principles that the teacher was using, but these general principles must be derivable from previously known principles in order for an explanation to be possible. In one version of the FERMI system, Cheng and Carbonell address methods of building macro-operators that handle iterative and conditional sequences from a problem solving trace. This is a more elaborate form of rule composition than that used in ABLE. Both ABLE and PHYSICS101 address the issue of improving the ability to apply known physics laws or formulas to physics problems, although in the case of PHYSICS101 this involves the learning of new useful concepts that are constructed from known concepts. FERMI's learning is not specifically centered on the use of physics laws, but it finds iterative cycles in the problem solving trace and builds macro-operators that accomplish the same thing.

Many other skills take part in the physics problem solving process. Human problem solvers have to learn these skills before they can use them. This includes learning language, learning abstract physics concepts, learning rules for elaborating and making inferences about a problem situation, learning how to map one representation into a more formal representation, discovering new laws, and learning from instruction. A complete psychological model would have to be able to learn these things also. These were not addressed in the systems reviewed.

Higher-level strategies – the control mechanism

The flow of activity between processes in the framework diagram is not specified. There has to be some high level control over this flow, but it could be a fixed algorithm or it could be something that is itself subject to meta-level reasoning. There are control decisions that have to be made in operating within this framework. The framework itself does not say anything about how those decisions are made.

All of the systems reviewed either had a fixed high level algorithm, or else avoided the issue entirely by concentrating on only one of the processes in the framework. Bundy's MECHO claimed to do meta-level reasoning, but it did meta-level reasoning only on individual subtasks such as algebraic manipulation of equations or semantic filtering of language parses. It had a fixed algorithm at a higher level and it never considered using another solution method besides solving equations. For example, it never ran a step-by-step time simulation of a problem to get an answer.

People obviously have multiple strategies and go back and forth between them. They also go back and forth between different steps in the problem solving process. They go back and reread the problem, they try some equations and go back and check if the answer is reasonable. This high level control is not specified in our framework and is not strongly addressed in any of the systems we reviewed.

4 Conclusions and Future Directions

Physics problem solving is a task that involves several subtasks and different types of representation that can be used to describe the same problem. Many of these subtasks have been addressed by computational models implemented in computer programs. A general framework for the representation types of physics problem solving was presented as a basis for comparison among six different computer simulations. This framework was useful in making clear distinctions among the systems presented.

Work on models of physics problem solving is still in its infancy. There has been interesting work done, but there remains much to do. Ongoing projects, such as the MECHO and FERMI systems, hold promise of delivering substantial results in the years ahead. At this point it appears that two areas in physics problem solving stand out as needing more research. These are learning and problem comprehension. The topic of learning the skills and knowledge needed for competent problem solving in physics has not been fully explored. The specific knowledge and processes that are involved in the transformation from a situation described in common terms into an abstract physics and mathematical model also need further study.

The connection between common sense, qualitative reasoning about objects and events in the world and abstract physics reasoning or mathematical reasoning has not yet been well addressed. Some work on qualitative reasoning and envisioning has made progress in understanding the reasoning behind a common sense understanding of the world. There is still much to be done in this area. Most of the work so far is concentrated on formal methods of applying formulas based on abstract physical principles. Psychological studies, such as those discussed by Reif [Rei86] indicate that qualitative understanding of abstract physics concepts, in addition to knowledge of formulas and quanitative rules, is essential to high expertise in physics problem solving. The systems we have reviewed have focused on one type of reasoning or the other, or have glossed over the process of tying the two together. Experts who can reason easily with abstract physics concepts and apply them to physical situations with great ease are often said to have *physical intuition.* They may be able to visualize how a complex situation will turn out, or they may simple *know* what underlying principles apply in a certain situation. They have a great qualitative understanding of abstract physical concepts. Research that provides more understanding about physical intuition and how it might be developed would be valuable.

The systems reviewed were impressive in the areas that they addressed, but none is close to being a complete model of physics problem solving. By our framework a complete physics problem solving model should address all five types of problem representation and should be able to coordinate the various types of representation effectively. It is probably not yet time to try to build a complete model of physics problem solving, because so much is still unknown about how each type of representation should work.

Physics can be treated as a formal system with physical laws taken as axioms and other principles considered as theorems derived from those axioms. However, all of these formal abstract concepts represent an explanation of real events observed in the world (or in the laboratory). They represent one hypothesis about how the world *really* works. Over many years physicists have been gradually discovering regularities and conceiving of laws and theories to summarize and explain the regularities that have been discovered. Physics researchers are not just figuring out better ways to apply principles learned from a book. They are experimenting and theorizing to discover more basic laws and underlying principles that account for the ways things work in the world. A system that can do flawless physics problem solving based on assuming that current laws are complete and correct would be a wonderful tool, but it would not be a physicist. Perhaps research into the scientific discovery process will lead to insights that will allow for better or self-improving physics problem solving systems.

5 Acknowledgements

The author wishes to thank Dennis Kibler, Rogers Hall, and Etienne Wenger for discussions that influenced this paper. Dennis Kibler and David Benjamin gave many helpful comments on earlier drafts. The author also gratefully acknowledges the support received in the form of a graduate fellowship from the American Society for Engineering Education and the Office of Naval Research.

References

[BB83] Alan Bundy and Lawrence Byrd. Using the method of fibres in Mecho to calculate radii of gyration. In Dedre Gentner and Albert L. Stevens, editors, *Mental Models*, pages 253–266, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

[BBL*79] Alan Bundy, Lawrence Byrd, George Luger, Chris Mellish, and Martha Palmer. Solving mechanics problems using meta-level inference. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 1017–1027, Tokyo, Japan, August 1979.

[Bob68] Daniel G. Bobrow. Natural language input for a computer problem-solving system. In Marvin Minsky, editor, *Semantic Information Processing*, M.I.T. Press, Cambridge, Mass., 1968.

[BW77] Daniel G. Bobrow and Terry Winograd. An overview of KRL, a knowledge representation language. *Cognitive Science*, 1(1):3–46, 1977.

[CC86] Patricia W. Cheng and Jaime G. Carbonell. The fermi system: inducing iterative macro-operators from experience. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 490–495, Philadelphia, Pennsylvania, August 1986.

[Cha68] Eugene Charniak. *CARPS, a Program which Solves Calculus Word Problems.* Technical Report MAC-TR-51, M.I.T., July 1968.

[DeJ81] Gerald DeJong. Generalizations based on explanations. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 60–67, Vancouver, B.C., Canada, August 1981.

[dK77] Johan de Kleer. Multiple representations of knowledge in a mechanics problem-solver. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 299–304, M.I.T., Cambridge, Massachusetts, August 1977.

[dK79] Johan de Kleer. The origin and resolution of ambiguities in causal arguments. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 197–203, Tokyo, Japan, August 1979.

[dKB81] Johan de Kleer and John Seely Brown. Mental models of physical mechanisms and their acquisition. In John R. Anderson, editor, *Cognitive Skills and Their Acquisition*, pages 285–309, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1981.

[GG83] Dedre Gentner and Donald R. Gentner. Flowing waters or teeming crowds: mental models of electricity. In Dedre Gentner and Albert L. Stevens, editors, *Mental Models*, pages 99–129, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

[HKWT86] Rogers P. Hall, Dennis Kibler, Etienne Wenger, and Chris Truxaw. *Exploring the Episodic Structure of Algebra Story Problem Solving.* Technical Report 86-24, U.C.I., November 1986.

[Lar81a] Jill H. Larkin. Cognition of learning physics. *American Journal of Physics*, 49(6):534–541, June 1981.

[Lar81b] Jill H. Larkin. Enriching formal knowledge: a model for learning to solve textbook physics problems. In John R. Anderson, editor, *Cognitive Skills and Their Acquisition*, pages 311–334, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1981.

[Lar83] Jill H. Larkin. The role of problem representation in physics. In Dedre Gentner and Albert L. Stevens, editors, *Mental Models*, pages 75–98, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

[LMSS80a] Jill H. Larkin, John McDermott, Dorothea P. Simon, and Herbert A. Simon. Expert and novice performance in solving physics problems. *Science*, 208(20):1335–1342, June 1980.

[LMSS80b] Jill H. Larkin, John McDermott, Dorothea P. Simon, and Herbert A. Simon. Models of competence in solving physics problems. *Cognitive Science*, 4:317–345, 1980.

[LRC86] Jill H. Larkin, Frederick Reif, and Jaime Carbonell. FERMI: a flexible expert reasoner with multi-domain inference. *Cognitive Science*, 9, 1986.

[LS81] Jill H. Larkin and Herbert A. Simon. Learning through growth of skill in mental modeling. In *Proceedings of the Third Annual Conference of the Cognitive Science Society*, pages 106–111, Berkeley, California, August 1981.

[NA80] Gordon S. Novak, Jr. and Augustyn A. Araya. Research on expert problem solving in physics. In *Proceedings of the First National Conference on Artificial Intelligence*, pages 178–180, Stanford, California, August 1980.

[Nov76] Gordon S. Novak, Jr. *Computer Understanding of Physics Problems Stated in Natural Language*. PhD thesis, The University of Texas at Austin, March 1976.

[Nov77] Gordon S. Novak, Jr. Representations of knowledge in a program for solving physics problems. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, pages 286–291, M.I.T., Cambridge, Massachusetts, August 1977.

[Rei86] Frederick Reif. Interpretation of scientific or mathematical concepts: cognitive issues and instructional implications. 1986. unpublished manuscript.

[RH82] Frederick Reif and Joan I. Heller. Knowledge structure and problem solving in physics. *Educational Psychologist*, 17(2):102–127, 1982.

[SD85] Jude W. Shavlik and Gerald F. DeJong. Building a computer model of learning classical mechanics. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 351–355, Irvine, California, August 1985.

[Sha85] Jude W. Shavlik. Learning about momentum conservation. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 667–669, Los Angeles, California, August 1985.