MULTIPLES REPRESENTATIONS OF KNOWLEDGE IN A MECHANICS PROBLEM-SOLVER

Johan de Kleer

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract

Expert problem-solving programs have focused on working problems which humans consider difficult Oddly, many such problem-solvers could not solve less difficult versions of the problems addressed by their expertise This shortcoming also contributed to these programs' inability to solve harder problems. To overcome this 'paradox' requires multiple representations of knowledge, inferencing schemes for each, and communication schemes between them.

This paper presents a program, NEWTON, applying this idea to the domain of simple classical mechanics. NEWTON employs the method of envisionment, whereby simple questions may be answered directly, and plans produced for solving more complex problems. Envisioning enables NEWTON to use qualitative arguments when possible, with resorts to mathematical equations only if the qualitative reasoning fails to produce a solution.

Introduction

Expert problem-solving programs have focused on working problems which humans consider difficult Charniak demonstrated the expertise of CARPS [68] by using it to solve problems from a freshman calculus text. The achievements of MACSYMA [Mathlab 74] are touted by exhibiting some complicated expressions it can integrate. There is no doubt that these two problem areas are difficult, but before we attribute expertise to these programs we should first examine the depth of their understanding of the problem areas I propose one criterion: *As well as being able to solve the difficult problems, (he expert problem-solver should be able to solve simpler versions of a problem with qualitatively simpler techniques.* We will see that the inability to solve simple problems is the source of many of the difficulties problem-solvers encounter in attacking harder problems.

My experience teaching electrical network theory suggests that this criterion also applies to students Consider the problem of determining the average power dissipated by a network. A general technique which solves this problem is to write down Kirchoff's current law for all the nodes and solve the resulting equations. A poor student always applies this technique immediately, while the better student looks first to see whether some simplifications are possible. For example, if the network consisted of only capacitors and inductors, the poor student still sets up equations while the bet'er student immediately replies zero since capacitors and inductors do not dissipate power. The reason the student who immediately sets up equations does worse, is that without first simplifying the problem the resulting equations are often unmanageable. If the topology of the network was unspecified, the poor student would be unable to even set up the equations while the better student would stilt be able to solve the problem.

To determine whether an the object released at A reaches B on the roughened track requires a careful analysis of the shape and frictional properties of the track.

The slightly different problem where friction is zero can be solved with the same technique. This would be a rather stupid since a simple comparison of the relative heights of A and B would solve the problem directly. The mark of an expert is that such qualitatively simpler problems are attacked by qualitatively different techniques.

Theory

I propose that an expert problem-solver should be able to employ multiple representations for the same problem Within each representation radically different reasoning techniques can be used. By employing the different representations, the problem-solver can solve problems of varying difficulty and, more importantly, use only those reasoning techniques which are appropriate to the difficulty of the problem. By definition, such a problem-solver meets the above criterion.

In mechanics, a useful distinction can be made between qualitative and quantitative knowledge. Qualitative knowledge represents the scene in terms of gross features such the general type of curve and the relative heights between points. Quantitative knowledge represents the same scene in terms of mathematical equations describing the shapes of the curves and the numerical distance between points. A simple qualitative rule uses the relative heights to determine whether an object released at one point can reach the other Quantitative reasoning, on the other hand, symbolically manipulates the mathematical equations and numerical quantities to obtain the same result.

A problem-solver which employs these two representations for solving the same problem has a number of distinct advantages:

(1) It solves simpler problems with drastically simpler techniques.

(2) Even when qualitative analysis fails, it sets up specific plans which greatly simplify the quantitative analysis of the problem

(3) Qualitative analysis can handle indeterminacies in problem specification.

The first advantage has already been discussed

Even if we were only interested in problems requiring equations, the qualitative analysis still performs a crucial role in the problem-solving. By itself a mathematical equation contains no useful information. To make use of an equation the meanings of each of the variables must be specified and the conditions of its applicability must be known There are many equations describing the motion of moving objects and relating the dimensions of physical objects, but how do we determine which of these equations are relevant to the problem at hand? Although the qualitative analysis of the problem may fail, requiring quantitative analysis, the qualitative analysis determines the kind of event happening thus providing a concise suggestion as to which equations are relevant.

The qualitative analysis also provides an overall structure for the solution of the problem A problem can involve a number of independent parts each requiring quantitative solution. Consider the problem of a block sliding over a hill, first you must determine whether the block can make it to the top of the hill, and then you must determine whether or not the block flies off the hill because it is too steep on the other side. The qualitative analysis first presents the problem of reaching the top for quantitative analysis, and if the top is reachable it presents the problem of whether the object falls off the hill on the other side.

The qualitative argument does not require a completely described scene. For example, qualitative analysis tells you that a ball will roll down an inclined plane without needing to know the slope of the incline or the radius of the ball. To determine the velocity of the ball at the bottom of the incline these other quantities must be incorporated into equations which are subsequently solved for the final velocity

A difficulty introduced by multiple representations is communication. One problem is that the way one representation refers to a particular entity is often radically different from how the other representations refer to this same object. Another problem is the format of the queries and replies between representations.

The Program NEWTON

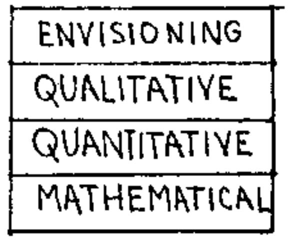
Although the theory suggests that multiple representations are useful, it does not provide any information about the details of the representations or how to utilize them In order to explore the theory further an expert problem-solver NEWTON has been constructed. It solves problems in the mechanics mini-world of "roller coasters" (the kinematics of objects moving on surfaces). NEWTON is not intended to be a general mechanics problem-solver, but is only used to demonstrate the above principles. Within its mini-world it can handle a wide range of problems. It uses qualitative aigumeiits when possible, but will resort to equations if necessary It recognizes nonsensical problems. It does not become confused or inefficient when given irrelevant facts, objects or equations.

One extremely important piece of qualitative knowledge is the ability to roughly predict what will happen in a given scene. For example, qualitative knowledge tells you that an unsupported object will fall or that a rubber ball will bounce. We will call this *envisioning.* More formally. envisioning means generating a progression of snapshots which describe what might or could happen The most important feature of a scene in the roller coaster world is the position of the moving object, and so the snapshot is best described by the position of the object An entire event is described by a tier. each node being a possible position for the object and each aic indicating an action which moves the object from one position to another.

The quantitative knowledge uses a FRAME like organization to package together mathematical equations There are an extremely large number of different equations that could be applicable to any problem Fortunately, since equations tend to come in groups, an individual decision need not be made about the relevance of every single equation The relationships between the angles and sides of a triangle form one such group or FRAME. Another possible FRAME is the kinematic equations which hold for uniformly acxeleiatnn; objects. With this representation only a small number of decisions are required to determine which equations are relevant. It also provides a convenient way to provide meaning to the variables in the equations Since the FRAME only applies when certain conditions are met, the variables in the equations of the FRAME can refer to these conditions to provide a meaning to the variables which is general amonj> all the FRAMEs.

In order to actually solve mechanics problems two other distinctly different representations become important One of the representations originates from the qualitative knowledge and the other originates from the quantitative knowledge. The envisioning can often solve the problem directly, however, if it fails it must be able to articulate its difficulty for the FRAME-rcpresentations This requires carefully analyzing and transforming the trees generated by the envisionment utilizing separate analysis and transformation rules. This analysis determines which top-level FRAME is relevant to the problem and which variables need to be solved for. The result of instantiating the FRAMEs is an information structure which relates all the equations and variables relevant to the original problem. This dependency network is then analyzed to identify possible paths to the solution. Symbolic mathematical techniques are then applied to check the possible path. If the manipulation is unsuccessful or intractable with the symbolic techniques available other paths are tried.

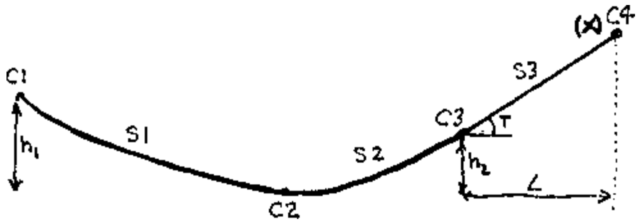
With these added refinements the organization of NEWTON is as follows:



Envisioning represents the original scene in terms of its gross features determining what might possibly happen and recording this information in a tree. Its knowledge is represented by rules which look at gross features and then extend the envisioning tree. The remaining qualitative knowledge represents the original problem in terms of the tree generated by the envisionment. Its knowledge is represented as transformation and analysis rules whose application eventually results in a plan to solve the problem The quantitative knowledge *is* represented in FRAMEs, a small subset of which is instantiated for any particular problem It maintains a CONNIVER-like [McDermott & Sussman 74] database to represent the semantics of variables. After instantiation the FRAMEs are examined for possible paths to the solution, and symbolic techniques are invoked to evaluate these possible paths.

The only entities which can be referred to in messages between representations are physical objects, instances of time, and variables describing them. Every representation refers to these primitive entities in the same way. The problem of the format of the messages has been resolved by having special purpose experts between each two representations that ever wish to communicate. This is not as unsatisfactory as it might seem since our theory imposes very rigid constraints on the nature of this communication. A query is generated only after the problem is discovered to be unsolvable in one representation, and thus the query asks a very precise and restricted question. The answers generated by queries are limited by the kinds of questions asked.

To get an overall idea of the different representations consider the following simple mechanics problem:



A small block slides from rest along the indicated frictionless surface. Will the block reach the point marked X?

The following is a possible protocol of a human solving the problem.

"The block will start to slide down the curved surface without falling off or changing direction After reaching the bottom it starts going up. It still will not fall off, but it may start sliding back. If the block ever reaches the straight section it still will not fall off there, but it may reverse its direction. To determine exactly whether the block reaches X we must study the velocity of the block as it moves along the surface. The velocity at the bottom can be computed by using conservation of energy:

$$v_1 = (2 g h_1)^{1/2}$$

Similarly, this velocity and conservation of energy can be used to set up an equation which can be solved for the velocity *(v)* at the start of the straight section.

$$1/2\, m v_2^2 = 1/2\, m v_1^2 - m g h_2$$

If the solution for v_2 is imaginary, then the straight segment is never reached. At the straight section we can use kinematics to find out whether the block ever reaches *X*. The acceleration of the block along the surface must be:

$$a = g \sin T$$

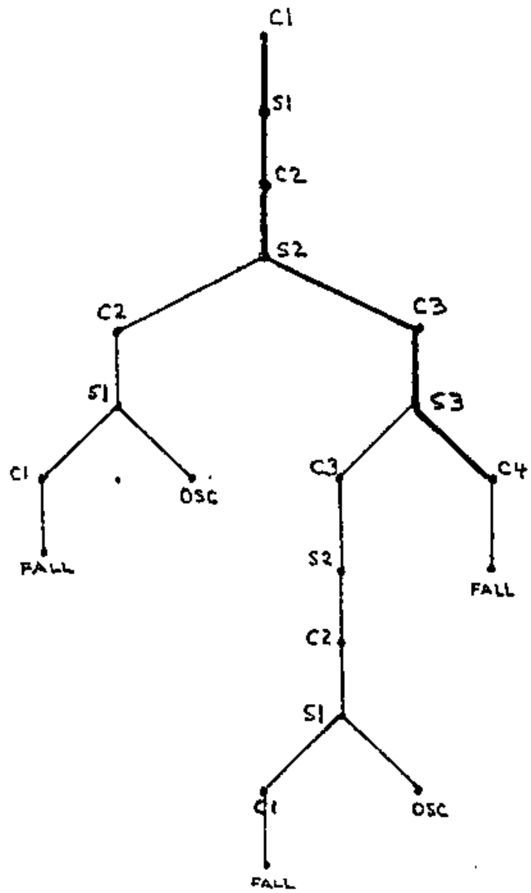
The length of the straight segment is $L / \cos T$ so by the well known kinematic equation relating acceleration, distance and velocities:

$$v_3^2 = v_2^2 - 2 L g \tan T$$

Again if v_3 is imaginary, X is not reachable"

The first part of the protocol which involved identifying a possible path to reach *X* we call envisioning Envisioning describes (and so limits) the problem space foimed when we ignore the specific values of the variables The problem had variables *h1*, h_2, *T* and *L* which were requited for the solution but the protocol (before the decision to calculate velocities) held true for a wide range of values for the^e variables. The reasoning depended on $h_1 > 0$, $h_2 > 0$, $L > 0$, $0 < T < 90$ and the facts that all the curves were concave fiom the perspective of the object and that the curves were differentiable everywhere. All this information could be assumed from the diagram. Everything that was predicted by the envisioning was achievable for some values of the variables and every possible assignment of values to variables was described.

The envisionment NEWTON generates for this problem is described by the following figure. The object starts at corner CI, slides through segment SI, reaches corner *C2,* slides through segment $2, either slides back on segment S2 or reaches corner C3, and so forth.



Many questions can be answered directly from the envisionment. For the above problem the question "Will it reach S2?" can be answered without further reasoning. Envisioning fails to answer the question when it predicts a number of possibilities. When the block was sliding up the hill, it could not be determined when or if it would start sliding back. It is in identifying these multiple possibility points tint the envisioner sets up specific subproblems Even when further reasoning is required to resolve such a qualitative ambiguity, envisioning identifies those possibilities it must distinguish between. Although there was the problem of determining whether the block would or would not slide back on the curve, the possibility of the block falling off had been eliminated by envisioning. In summary, envisioning gives local and very specific problems for further analysis and, on a global scale, envisioning provides an organization and plan to solve the entire problem. The trace of the possibilities through time provides the basis for such a plan.

Quantitative knowledge is used to disambiguate between the possibilities occuring at each fork It is important to note that FRAMEs are not procedures, but describe dependencies and assignments between variables These dependencies are searched for solutions to the goal variable. A FRAME is first examined to determine whether it has the solution for the goal variable. If this search is unsuccessful all equations which reference the goal variable are examined, and a subgoal is generated to find the remaining unknown variables in these equations. All the sub FRAMEs referenced by the FRAME are eventually included in this search Viewed from another perspective, the instantiation of a top-level FRAME results in a and-or graph which must be analyzed for possible solutions. Each equation which references the goal variable contributes to a disjunction since each such equation could possibly yield its solution. Unknown variables in equations referencing the goal variable contribute to conjunctions since every unknown in the equation must be determined to achieve the goal

The MASS-MOVEMENT FRAME knows about movements of objects on surfaces, but is not concerned about the possibility that the objects may fall off the surfaces The following is a description of MASS-MOVEMENT and the other FRAMES it references A FRAME uses two kinds of variables; the names of the objects it is concerned about and mathematical variables describing properties of these objects such as velocity or acceleration.

```
FRAME mass-movement OF object surface t1 t2
    VARIABLES:
        (a     : ACCELERATION OF object,
         theta : ANGLE OF surface)
    IF surface IS flat THEN
        (USE right-triangle ON surface,
         USE kin ON object surface t1 t2,
         a = G sin(theta)).
    USE energy ON object surface t1 t2.

FRAME energy OF object surface t1 t2
    VARIABLES:
        (vi : VELOCITY OF object AT TIME t1,
         vf : VELOCITY OF object AT TIME t2,
         h  : HEIGHT OF surface)
    vf² - vi² = 2 G h.
```

```
FRAME right-triangle OF triangle
     VARIABLES:
         (h      : HEIGHT OF triangle,
          base   : BASE OF triangle,
          hyp    : DISTANCE OF triangle,
          theta1 : ANGLE1 OF triangle,
          theta2 : ANGLE2 OF triangle)
     hyp = sqrt(h^2 + base^2)
     sin(theta1) = h / hyp
     sin(theta2) = base / hyp.

FRAME kin OF object surface t1 t2
     VARIABLES:
         (vf : VELOCITY OF object AT TIME t2,
          vi : VELOCITY OF object AT TIME t1,
          d  : DISTANCE OF surface,
          t  : TIME BETWEEN t1 AND t2,
          a  : ACCELERATION OF object)
    vf = vi + a t
    vf^2 = vi^2 + 2 a d
    d = vi t + .5 a t^2.
```

NEWTON would solve this problem without FRAMES by comparing the heights of the endpoints directly In order to demonstrate the quantitative knowledge, this possible solution (among many others) will be ignored. The FRAMES NEWTON uses are much more sophisticated than these presented here; they are not shown since they involve many features which are not relevant to this presentation In order to get a better understanding of how FRAMEs interact in the problem-solving process we will artificially make */» h^, T* and *L* unknown and only provide their values when necessary

Envisioning determines that there is never a possibility that the object will fly off. Since the envisionment tree for this problem has a fork at S2 and S3, the problem is decomposed into the two subproblems of first disambiguating what happens at S2 and then disambiguating what happens at S3. This is the plan the qualitative knowledge develops fot solving the problem. In order to analyze what happens on S2. the velocity of the object at the beginning of S2 must be determined. The velocity at the beginning of S2 is the same as at the end of SI, so the first problem to be solved by the quantitative knowledge is to find the velocity at the end of SI

To find the velocity at the end of SI MASS-MOVEMENT must be invoked:

(MASS-MOVEMENT (BI SI TIME1 TME2))

Before MASS-MOVEMENT is invoked, variables must be assigned values and meanings:

(VELOCITY BI TIMED -known
(VELOCITY BI TIME2) - desired goal

When MASS-MOVEMENT is invoked IFRAME attempts to find a value for this variable. There are two places in M ASS MOVEMENT where possible assignments to VF rake place IFRAME discovers that the assignment in the conditional cannot be reached since SI is not flat. The only alternative is the ENERGY FRAME. Using ENERCY is unsuccessful since HEIGHT is unknown. Every possible attempt to achieve a value for VF having now failed, the alternative is to generate subgoats of discovering the variables which are blocking a solution to the desired goal variable. The path ro VF is blocked by HEIGHT, but there are no other accessible references to HEIGHT in ENERGY or MASS-MOVEMENT Problem-solving now halts until HEIGHT is given, after which IFRAME reexamines ENERGY and returns a value for VF This value is remembered and the segment S2 is examined in much the same way using the VF on SI as VI on S2:

(MASS-MOVEMENT (BI S2 T1ME2 TIME3))

Note that ENERGY returns an "impossible" result if the equations result in an imaginary solution, thus indicating that the object cannot traverse S2.

On S3 IFRAME has two possible paths to a solution If ENERGY is tried it fails because HEIGHT is unknown Since S3 is flat KIN can be tried for a solution For KIN to succeed either D or T must be known. Again every path to VF is blocked. Finding a value for either HEIGHT, D or T would be sufficient to solve for VF. There are no other references to T in the FRAMEs so T cannot be achieved in MASS-MOVEMENT. The two variables HEIGHT and D can be found by the FRAME RTRI. RTRI is then invoked on segment S3. There is not enough information to solve for these variables. The values for TI (angle of surface) and L (base length of surface) in the instantiation of RTRI on S3 aie given and IFRAME proceeds. Now RTRI returns with values for both D and HEIGHT. IFRAME has a choice between reexamining KIN or ENERGY to solve the problem Depending on whether this results in an "impossible'' solution or a particular value for VF, the question of whether OI is reachable has been answered.

In order to demonstrate the importance of envisioning and to illustrate the nature of the communication between representations, a simplified version of envisioning and its interaction with quantitative knowledge will be discussed in the next section. For a more complete discussion of this topic and the other representations the reader is referred to [de Kleer 75].

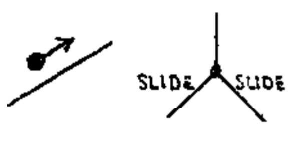
Envisioning and Planning

The most basic and primitive knowledge about physics is envisioning For NEWTON, envisioning means generating a progression of scenes encoded in a symbolic description which describe what could happen Both in NEWTON and in people, the envisionment of the event is the first step in the understanding of the problem

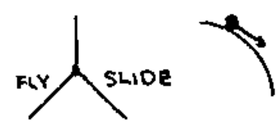
In fact, envisioning is necessary to understand the event at all. To understand that a pencil might roll off a table requires the ability to envision that event without actually seeing it happen. Envisioning is pre-physics knowledge and its presence is independent of the goal of solving mechanics problems. Envisioning is, however, the fundamental tool for understanding mechanics.

The envisioner requires that the original path be described in terms of segments for which the slope and concavity don't change sign, and achieves the envisionment by applying general rules which describe object motions on segments whose concavity and slope don't change sign As a consequence points at which the slope or concavity are zero or discontinuous are identified. In NEWTON envisionment starts with the initial situation, identifies what is happening in that situation, generates changed situations, and recursively analyzes these new situations until all possibilities are exhausted It may seem that it would be more efficient to only explore the situations that lie on some path to the goal [Bundy 77] Unfortunately, this goal directed reasoning requires some measure of how close it is to the goal, and this requires a separate analysis of the scene. An extra pre-analysis of the scene to identify paths to the goal (nearness to goal) must involve as much work as the envisioning itself. Simple strategies such as "when the goal is left, move left" are insufficient when the path turns around.

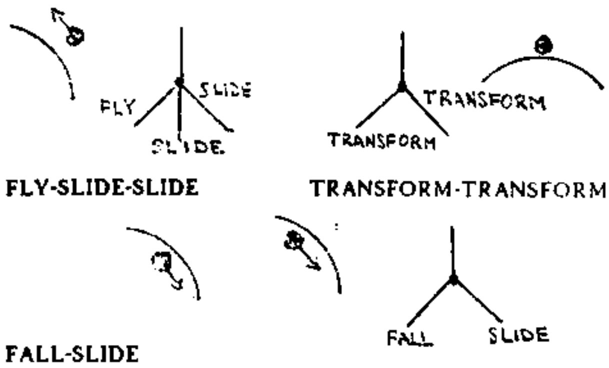
The envisionment results in a tree, every fork of which indicates a qualitative ambiguity which needs to be resolved. The forks for the roller coaster domain can be classified into the actions that occur there:



SLIDE-SLIDE



FLY-SLIDE

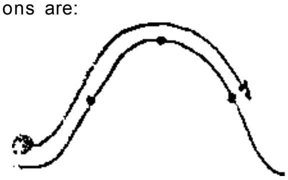


For each kind of qualitative ambiguity there exists a top-level of FRAME which can be used to resolve that ambiguity. Many of the FRAMEs for different ambiguities reference each other. We have already seen one Mich rollcamn of FRAMES for SLIDE-SLIDE: MASS-MOVEMENT

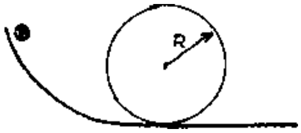
After the envisionment is completed, NEWTON accumulates a list of ambiguities which must be resolved to reach the desired goal This primitive plan is then analyzed to optimize the disambiguations that must be done The main optimization is the deletion of plan steps Plan steps can be deleted if succeeding plan steps implicitly repeat it The plan for the example sliding block problem is:

Solve SLIDE-SLIDE on S2 and fail if C3 is not reached.
Solve SLIDE-SLIDE on S3 and fail if X is not reached

The second plan step completely determines the relevance of the first step to the goal. Hence it can be eliminated Some other eliminations are:



SLIDE-SLIDE, FLY-SLIDE-SLIDE, FLY-SLIDE -> FLY SLIDE-SLIDE



SLIDE-SLIDE, FALL-SLIDE, FALL-SLIDE(symmctrical) -> FALL-SLIDE

Envisioning looks at only one point or segment at a time and the possible actions it determines pertain only to thru particular point or segment. Since the envisioner uses local information to determine local actions, the primitive plan it generates is restricted to employing local quantitative techniques With the help of these transformation rules which eliminate unimportant local plan steps, the important global structure of the problem can be identified. The resulting plan can take maximal advantage of the global quantitative technique of conservation of energy.

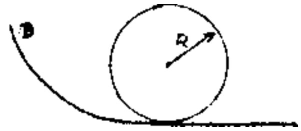
Problem Formulation

NEWTON contains quantitative techniques ro handle every possible qualitative ambiguity which can be generated by the envisioner. Given the absolute numerical position of every point and the shape of every segment. NEWTON can always determine what will happen if it has the FRAMEs to deal with the segment types. There are many other qualitative questions which can be asked besides "What happens?" The qualitative formulation of a problem leaves out many details in both the description of the scene and the query about that scene. NEWTON attempts to reduce questions to the simple "What happens?" type as much as possible. NEWTON also has some techniques to deal with other question types, but these are severely limited and it is difficult to characterize what class of problems they solve.

The question can involve a quantitative request about a qualitative predicate: "What is the velocity of the object when it falls off?" NEWTON first tries to satisfy the qualitative predicate, and if that predicate is satisfied attempts to determine the velocity when it falls off

If the original problem contains symbolic parameters, and all the qualitative disambiguations can be marie, it returns the final result in terms of these parameters. NEWTON fails when an unknown parameter makes a qualitative disambiguation impossible. (A smarter system might give a disjunctive answer.) If NEWTON is provided with an explicit numerical range for a particular parameter, it will attempt to do the disambiguation with this limited information

A problem can specify a resultant effect and ask what initial conditions lead to that effect. NEWTON solves this type of problem by hypothesizing variables for all the relevant initial conditions and the comparisons which otherwise would be used in disambiguation are used to accumulate inequality constraints on these initial conditions. The loop the-loop problem is an example of this type:



A small block starts at rest and slides along a frictionless loop-the-loop as shown in the figure. What should the minimal initial height be so that the block successfully completes the loop-the-loop?

The problem can implicitly refer to points which are not present in the original figure. NEWTON can introduce points in a figure which are not zeroes or singularities

A small block slides from rest from the top of a frictionless sphere. How far below the top does it lose contact with the sphere?

There are many questions about the roller coaster world that NEWTON cannot solve:

"What is the period of an object on a cycloid?"
"Would the object still make it if the height of the hill was increased?"
"What is the curve of shortest time between two points?"
"If the fall-off velocity is determined what must the initial height be?"

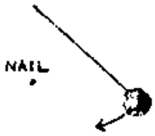
Concluding Remarks

We have seen how NEWTON can solve a wide range of problems in its mini-world The power of thp techniques it uses are appropriate to the difficulty of the problem it is asked to solve. When qualitative arguments will work it uses them, and otherwise resorts to mathematical equations. The organization of multiple representations allows it to limit the calculations as the problem-solving progresses.

Envisioning is an extremely powerful tool for problem-solving and deserves further attention In order to do any problem-solving about moving physical objects some kind of envisioning is required. Clearly, an envisioner is needed in order to build a general mechanics problem-solver Envisioning would be useful in other domains such as electronics Which depend on mechanical intuitions In general, envisioning might be useful in every problem-solving domain for which models exist. An example of a domain whci^ envisioning plays no role is pure mathematics or the solving of equations independent of some domain

NEWTON has three weaknesses which should be the subject of future research in this area; a lack of a theoiy of question types, an insufficiently powerful envisioncr, and an inappropriate theory of mathematical expertise

The current envisioner is relatively simple since it works with a single point object in only one dimension More general envisioning requires the manipulation of more than one object at a time and an analysis in more tha.n one dimension. Funt [76] describes a program WHISPER which can do certain kinds of two dimensional envisioning He is more interested in exploring the use of analogues than problem-solving in general. A slightly extended world for NEWTON would be a world with two moving objects. The difficulty here is that the envisioner needs a notion of tim*[3] A naive solution would be to envision the movement of rath object independently, and then assume a possible collision at each intersecting point. Each such point again yields two new trees. Two dimensional envisioning is far more complex since it introduces a large variety of new objects and possible interactions between them. Consider the problem of computing the period of this pendulum:



Two dimensional envisioning is needed to see that the nail shortens the length of the pendulum string thus shortening the the period.

NEWTON has no good theory of question types NEWTON can solve all "What happens?" questions which are totally described, and fortuitously slight modifications of these techniques can solve many other problem types. Unfoitunately. these other problem types cannot be characterized very well There is a need for a more powerful theory of question types, and techniques to deal with these

One of the unexpected difficulties encountered in implementing NEWTON was the interaction between quantitative knowledge and mathematical expertise NEWTON's mathematical expertise is provided by routines culled from MACSYMA. The problem in using these routines is that they are Just black boxes, and the quantitative knowledge needs other kinds of interactions than those normally provided by these routines. A simple example is the occurrence of multiple roots. In general MACSYMA does not generate appropriate explanations for why it fails to achieve some particular manipulation. NEWTON should have had another representation between the quantitative and the mathematical which knew about mathematics and about MACSYMA.

These difficulties originate from a far more serious shortcoming of NEWTON: it treats equations purely as constraint expressions. It is solving a physical problem for which an inherent duality exists between the symbolic structure of the expressions and the actual physical situations. Each mathematical manipulation of the equations reflects some feature of the physical situation. The manipulation should be under constant control and observation such that any unusual features or difficulties should be immediately reported to the rest of the problem-solver which then decides how to proceed This requires that the problem-solver have much more control over the symbolic manipulation routines and constantly monitor transformations on the expressions to what import they have on the physical situation. One possibility for the manipulation routines are those suggested by Bundy [Bundy 75]

Acknowledgements

This research was conducted at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology Support for the Laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract number N000I4-75-C-0643.

References

[Bobrow, 68]
Bobrow, CD., "Natural Language Input for a Computer Problem Solving System", in Minsky (ed.), *Semantic Information Processing,* Cambridge: MIT. Press, 1968.

[Bundy 75]
Bundy, Alan, "Analyzing Mathematical Proofs (or reading between the lines)", Department of Artificial Intelligence, Research Report o2, University of Edinburgh, 197*1

[Bundy 77]
Bundy, Alan, "Will it Reach the Top? Prediction in the Mechanics World", Department of Artificial Intelligence, Research Report «31, University of Edinburgh, 1977

[Charniak 68]
Chamiak, E., CARPS, "A Program which solves Calculus Word Problems", Project MAC TR-51, Cambridge: MIT, 1968.

[de Kleer 75]
de Kleer, Johan, "Qualitative and Quantitative Knowledge in Classical Mechanics', Artificial Intelligence Laboratory TR-'352, Cambridge: MIT, 1975.

[Kleppner &■ Kolenkow 73]
Kleppner, Daniel, and Robert J. Kolenkow, *An Introduction to Mechanics,* New York: McGraw-Hill, 1973

[Mathlab 74]
Mathlab Group, "MACSYMA Reference Manual", Cambridge MIT, 1974.

[McDermott & Sussman 74]
McDermott, Drew, and Gerald Jay Sussman, "The Conniver Reference Manual", Artificial Intelligence Laboratory, A1M-259a, Cambridge. MIT., 1974.

[Minsky 73]
Minsky, Marvin, "A Framework for Representation of Knowledge", Artificial Intelligence Laboratory, AIM-306, Cambridge: M.I.T., 1973.

[Novak 76]
Novak, Gordon Shaw, "Computer Understanding of Physics Problems Stated in Natural Language", The University of Texas at Austin, Ph.D. Thesis, 1976.