WORKING PAPER 88

QUALITATIVE AND QUANTITATIVE KNOWLEDGE IN CLASSICAL MECHANICS

A Master's Thesis Proposal And Progress Report

Johan de Kleer

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

January, 1975

Abstract

Scientific knowledge can be roughly divided into the qualitative and the quantitative. The qualitative is usually concerned only with qualities such as gross behavior, while the quantitative requires recourse to equations and mathematics. This proposal examines the nature of this dichotomy and uses it to develop a representation for qualitative and quantitative knowledge. This theory is applied to classical mechanics, and a program is proposed which can solve problems in a simple roller coaster world.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defence and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0005.

TABLE OF CONTENTS:

1.0 INTRODUCTION	1
1.1 Qualitative and Quantitative Knowledge in Science	1
1.2 Discussion of Previous Work	2
1.3 Classical Mechanics as an Interesting Domain	3
2.0 REPRESENTATION OF QUALITATIVE KNOWLEDGE	6
2.1 Qualitative Knowledge about Mechanics	6
2.2 The Basic Principle of Qualitative Knowledge	11
2.3 The Model as a Representation of Qualitative Knowledge	12
3.0 THE UTILITY OF QUALITATIVE KNOWLEDGE	21
3.1 The Necessity for More Knowledge	21
3.2 Answering Qualitative Questions	21
3.3 The Necessity of Quantitative Knowledge	24
3.4 Communication Between Qualitative and Quantitative Knowledge	25
4.0 REPRESENTATION OF QUANTITATIVE KNOWLEDGE	28
4.1 General Overview	28
4.2 A Scenario using Quantitative Knowledge	30
4.3 Frames as a Representation	34
4.4 Frame Syntax and Semantics	38
4.5 Frame Control Structure	40
4.6 A Critique of this Theory	43
5.0 CONCLUDING REMARKS	46
5.1 Discussion of the Theory	46
5.2 Future Work	46

1.0 INTRODUCTION

1.1 Qualitative and Quantitative Knowledge in Science

Scientific knowledge can be roughly divided into the qualitative and the quantitative. The qualitative is usually concerned only with qualities such as gross behavior, while the quantitative requires recourse to equations and mathematics. The intent of this thesis is to study this dichotomy and to represent these different kinds of knowledge in order to solve problems classical in mechanics.

Insight into the dichotomy can be gained by studying the development of mechanics in history and the acquisition of mechanics understanding by people. In people we see two phenomenon; there is a great resistance to the use of quantitative knowledge; and that which was once quantitative knowledge can become qualitative knowledge. A beginning high school student might not know that a ball rolling down a (frictionless) hill will reach the same height on the next hill. After learning about potential and kinetic energy he discovers that the heights must be equal. Now certain kinds of questions can be answered without resorting to equations. The question, "Will the ball make it over the next hill?", can be answered by examining a quality: the comparison of the two heights.

Historically, the science of mechanics developed for two reasons: firstly, to obtain a better understanding of what actually happens, and secondly, to deal with the more complex situations where qualitative knowledge breaks down. Often, the interest is still in qualities, but the situation is too complex and a transition to quantitative knowledge is necessary to obtain the result, after which quantities are compared and a qualitative result is obtained. Similarily, in students quantitative knowledge is acquired very late and although the quantitative knowledge is widely applicable it is only used in cases of extreme necessity.

As with qualitative knowledge certain principles underlie the structure of quantitative knowledge. This is elegantly discussed in Ernst Mach's book *The Science of Mechanics* <Mach, 60> in the chapter dealing with the formal development of mechanics. The development of mechanics is described as passing through the three periods of observation, deduction and formalization. The

last period in which all the quantitative knowledge of mechanics now lies is characterized as: "Here it is sought to put in a clear compendious form, or system the facts to be reproduced, so that each can be reached and pictured with the least intellectual effort."

From a practical, educational, and historical viewpoint the quantitative knowledge comes last and is used as a last resort. However, it is the natural outgrowth of the qualitative knowledge.

These two kinds of knowledge play different but important interacting roles in understanding mechanics. It is the qualitative knowledge of the situation that guides the quantitative analysis. The qualitative knowledge identifies relevant details and gives the analysis common sense. Qualitative knowledge helps construct analogies, by identifying qualitatively what happens. Perhaps most important is the goals the qualitative knowledge can provide. The quantitative knowledge is the natural outgrowth of the deficiencies of the qualitative knowledge, hence the qualitative knowledge can give a clear perspective for the analysis.

1.2 Discussion of Previous Work

Much of current Artificial Intelligence research into the representation of knowledge is relevant to this thesis. The current interest in frame-systems influences this proposal <Minsky, 74>. But, only a few researchers have ever tried to represent physical events and solve equations about them in the sense this thesis proposes.

One such investigation was Bobrow's work with STUDENT <Bobrow, 68>. This was a program which could solve algebra word problems. Its Deductive Model solves the problems once they have been transformed from their original English formulations. The major effort in this work was in the transformation, not in the Deductive Model which was in essence very simple.

A later work by Charniak was CARPS. It could solve calculus rate problems expressed in English. Charniak used a more sophisticated description for events than Bobrow, but the major emphasis was still in the understanding of natural language.

Recently, Charniak pursued an investigation into representing elementary electrostatics knowledge <Charniak,71> (chapter one of <Purcell, 65>). Although the interest was again to develop

"a firm semantic basis for a natural language understanding program", this work went further than the previous investigations in that he introduced a great deal more structure to the understanding. Many of the observations he presented in his paper parallel ideas presented here, particularly the idea of a model. His arguments failed to be convincing since his domain and knowledge structure were so limited that there was rarely any choice in what to do next. He also failed to recognize the importance and necessity of qualitative knowledge.

Currently, research is being done in understanding electronic circuits <Brown, 74> <Brown & Sussman, 74>. Although this research is more concerned with debugging these circuits than with solving problems about them, they are using qualitative knowledge to understand the circuits.

1.3 Classical Mechanics as an Interesting Domain

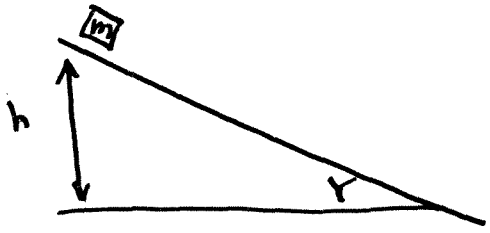
We will use these ideas of qualitative and quantitative knowledge to investigate representing knowledge about simple mechanics. The main interest is in representing the knowledge of this domain, not in natural language understanding of the problem description. This is in contrast to much of the previously mentioned research whose primary goal was to represent knowledge so that it could be used in a natural language understanding system.

This choice of domain was not arbitrary. Simple mechanics is very complex and requires a great variety of distinct techniques to understand it. Yet, almost everyone understands it in a simple qualitative sense when they walk, drive a car, ski, etc. This provides a wealth of situations in which to study their thinking about mechanics. Simple mechanics has been understood quantitatively for a long time and a variety of quantitative structures exist for it. In the sense we are interested in, simple mechanics is completely understood by modern physics. Having such a complete theory makes it easier to study quantitative representations for mechanics. Many theories already exist about its structure. Any mechanics textbook implicitly presents the author's theory of this stucture <den Hartog, 48> <Levinson, 61> <Kleppner & Kolenkow, 73>. There also exist textbooks written by authors who clearly did not understand certain aspects about mechanics such as Galileo <Galileo 00>. This gives us further insight. Also, many authors have given explicit theories about its

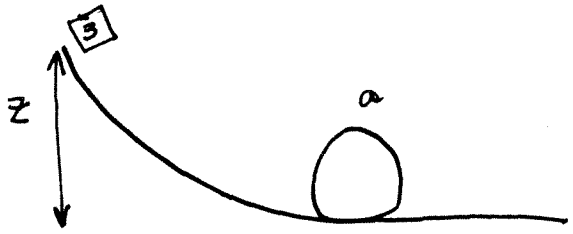
structure such as Mach <Mach, 60>. In summary, many sources of ideas and insights into the problem of representing mechanics knowledge are readily available.

The investigation of this problem is in two directions. One is the observation of people (including myself) doing mechanics and studying other theories of mechanics understanding in the aforementioned text books. The other approach which the remainder of this paper outlines is the construction of a program which embodies these ideas to solve simple mechanics problems.

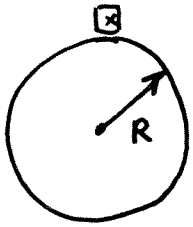
The following are some elementary mechanics problems which we would like this program to solve.



A block of mass M slides down a plane of angle Y. The problem is to find the speed of the block after it has descended through height h, assuming that it starts from rest and the coefficient of friction is zero.



A small block of mass m starts from rest and slides along a frictionless loop-the-loop as shown it the figure. What should be the initial height z, so that m pushes against the top of the track (at a) with a force equal to its weight.



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

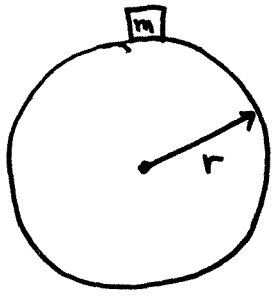
A bead of mass m is constrained to move on a frictionless wire in the shape of a cycloid. If the bead starts from rest at point O, (a) find the speed at the bottom of the path and (b) find the period.

2.0 REPRESENTATION OF QUALITATIVE KNOWLEDGE

2.1 Qualitative Knowledge about Mechanics

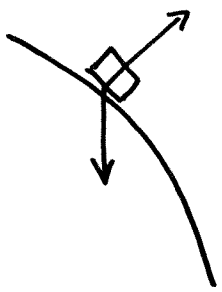
For the purpose of understanding the interplay between qualitative and quantitative knowledge let us observe how a typical student might solve some elementary mechanics problems.

The first example was presented earlier in section 1.3. It is a typical freshman physics problem:



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

"The block will slide downwards. At the very start of its motion it will remain in contact with the sphere since the surface is horizontal at the top. If the block ever did reach the equator it would have to immediately fall off as it would start sliding underneath the sphere. So the block loses contact with the sphere somewhere between the pole and the equator. At any point the block will either continue sliding along the surface or fly off, we know it will fly off before it reaches the equator, but where is that point? The forces on the block must be examined. There are only two forces acting on the block, one is gravity and the other is the reaction of the sphere.



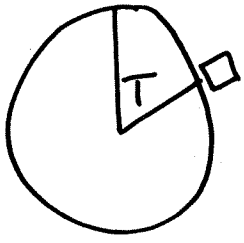
To remain in contact with the sphere the component of the total force in the direction of the center of the sphere must be sufficient for the block to continue in circular motion:

$$m v^2 / r$$

Since the direction of the reaction of the sphere is normal to its surface and thus away from its center we can ignore that force and just solve for the point at which the the component of the force of gravity in the direction of the center is insufficient to maintain circular motion. This happens when:

$$m v^2 / r = m g \cos T$$

Where the angle T is as defined in the diagram:



The velocity of the block can be computed by conservation of energy from the distance the block has already dropped through:

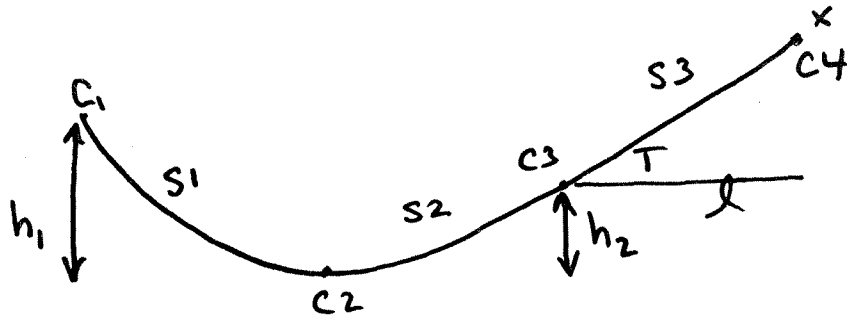
$$1 / 2\ m v^2 = m g r (1-\cos T)$$

These two equations can be easily solved for T:

$$\cos T = 2 / 3."$$

Until the questions about the forces on the block were asked, the student had only used

qualitative reasoning. After that, the solving involved continuous interaction between qualitative and quantitative reasoning. The second example is much simpler. Although it is somewhat contrived, it is at the level of difficulty of a high school student. It illustrates even more than the previous example the purpose and structure of qualitative knowledge:



A small block slides from rest along the indicated frictionless surface. Will the block reach the point marked *X*? (We will assume that the student does not understand qualitatively the principle of conservation of energy.)

"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 won't 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 change the direction of its movement. 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 using this velocity and conservation of energy we can set up an equation which can be solved for the velocity (v_2) 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, we know that the straight segment is never reached. At the straight section we could use kinematics to find out whether the block ever reaches *X*. The acceleration of the block along the surface must be:

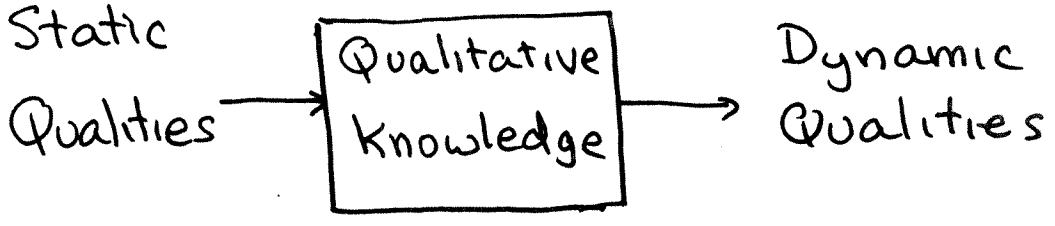
$$a = g \sin T$$

The length of the straight segment is $L / \cos T$, so using 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."

In these two protocols the structure and purpose of qualitative knowledge elucidates itself. Qualitative knowledge describes (and so limits) the solution space formed when we 'ignore' the values of the variables (i.e. let the variables have arbitrary values). The second problem had variables h_1, h_2, T and L which were required for the solution but the qualitative discussion (before the decision to calculate velocities) held true for a wide range of values for these variables. The qualitative reasoning depended on $h_1 > 0$, $h_2 > 0$, $L > 0$, $0 < T < 90$ and the facts that all the curves were concave from 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 qualitative reasoning was achievable for some values of the variables and every possible assignment of values to variables was described. Here we see what qualitative reasoning consists of: it considers only qualities (such as the sign of the slope of the surface) and then qualitatively describes all and only those possibilities which could possibly occur with those qualities being true. We now have two kinds of qualities, static and dynamic:



In general it is very difficult to define what qualities are, but the previous discussion presents many examples of them. The property of a surface being concave or convex is a quality.

The acuteness of an angle is a quality. The differentiability of a curve is a quality of that curve. The two protocols demonstrate that the language of dynamic qualities does not require the sophistication of static qualities. The language of dynamic qualities essentially consists of the qualities of moving in a direction, remaining at rest, changing direction of movement and falling. It also requires a concept of time which is not present in static qualities. Static qualities remain fixed throughout the entire problem while dynamic qualities are only true for a certain period of time.

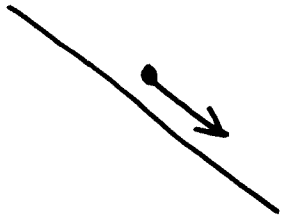
The previous paragraphs describe what qualitative reasoning is, we will now examine the role it plays in problem solving. In both protocols we see that the initial approach to the problem is that of qualitative reasoning. One would expect that if the question was simpler it could be answered by qualitative reasoning alone. Questions such as "Will it go past the equator?" or "Will it reach the bottom?" are answerable from the qualitative discussions. Qualitative reasoning fails to answer the problem 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 that the qualitative knowledge sets up specific problems for the quantitative knowledge. Even when quantitative reasoning is required to resolve such a qualitative ambiguity the qualitative knowledge identifies those possibilities it must distinguish between. Although there was the problem of determining whether the block would slide back or not on the curve, the possibility of the block falling off had been eliminated by qualitative reasoning. In summary, the qualitative knowledge gives local and very specific problems for the quantitative knowledge to solve and on a global scale the qualitative knowledge provides an organization and plan to solve the entire problem. The trace of the possibilities through time is such a plan.

To complete this general discussion about qualitative knowledge it should be noted that it must also have the ability to communicate. It must be able to communicate effectively with the quantitative knowledge. This means it must know about the structure of the quantitative knowledge. Qualitative knowledge also requires the ability to understand problems presented to it.

2.2 The Basic Principle of Qualitative Knowledge

In both static and dynamic qualities we see the principles of continuity and discontinuity. A curve is described as being either concave, convex or straight. These qualities capture the mathematical variable of the rate of change of slope -- mathematical concavity. The qualitative description has identified the single discontinuity and described those regions of continuity. Similarly for velocity; the object remains at rest, slides up or slides down.

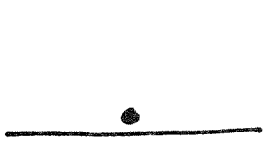
As an example of these principles we will study the qualitative knowledge needed to understand the statics of an object at rest on a surface. All curves are linear along a small segment, so only one kind of slope needs to be understood:



Case I

It does not matter how steep or how flat these slopes are, the action will be the same. A further problem only arises when the extremes of vertical and horizontal are considered; these are the two singularities of the slope. The idea of singularity is one of the powerful concepts of qualitative analysis: in between the singularities of some quantity affecting an action, the effect varies continuously. More simply, the singularities are the points at which violations of conventional rules are most likely to occur and furthermore there are usually radical changes in effects before and after a singularity (i.e. a discontinuity).

Although there is enough information in Case I to understand the vertical and horizontal cases; the singularity points are sufficiently important to merit separate attention:



Case II Case III

2.3 The Model as a Representation of Qualitative Knowledge

Qualitative reasoning involves observing static qualities and determining the dynamic qualities which describe the subsequent action. The principles of continuity and discontinuity can be used to represent the necessary knowledge. They suggest a very simple representation will suffice: productions with the static qualities on the left and dynamic qualities on the right. These productions will be organized into groups concerned with one aspect of the situation. Such groups are called *models*.

The proposed structure and representation of qualitative knowledge will be developed in an application to the simple mini-world of roller coasters. In chapter 4 the quantitative knowledge for this mini-world will be discussed.

The domain consists of a roller coaster on a track. This track can be arbitrarily shaped. The roller coaster is allowed to travel above or below the track, so loop-the-loops are legal. The roller coaster is, however, not attached to the track. The roller coaster will be assumed to be a point mass and moves without friction on the track. The only force acting is gravity, but collisions and trajectories are not included.

At first sight this mini-world may seem artificial. It is in fact only a slight simplification of the mini-world in which many freshman and high school physics problems are framed. The two problems of section 2.1 in fact fit into this domain. Furthermore, this domain is interesting because it is both simple and rich. It is simple enough to be understood by everyone. Yet, it requires a variety of different quantitative techniques to handle it: Newton's Laws, calculus, variational

calculus, differential equations. Each of these quantitative techniques is completely understood with current physical theories and thus its interaction with qualitative knowledge can be precisely studied.

The three different kinds of slope have already been established. The only other matter to consider about the statics of this simple domain is the relationship between the surface and the object, the object is either above or below the surface, and it is either on or off the surface. The resulting actions are either falling freely under gravity, remaining stationary, or sliding along the surface. The relations between these quantities can be expressed in a few simple productions:

below → fall

off → fall

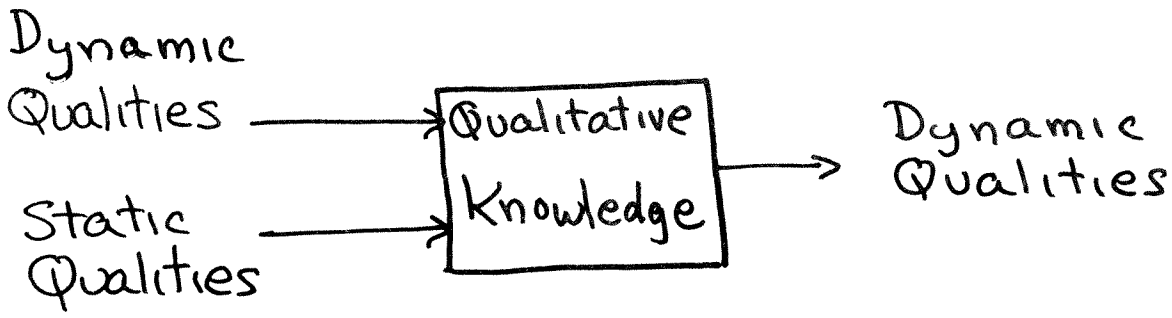
incline ∧ above ∧ on → slide-d

horizontal ∧ on → ϵ

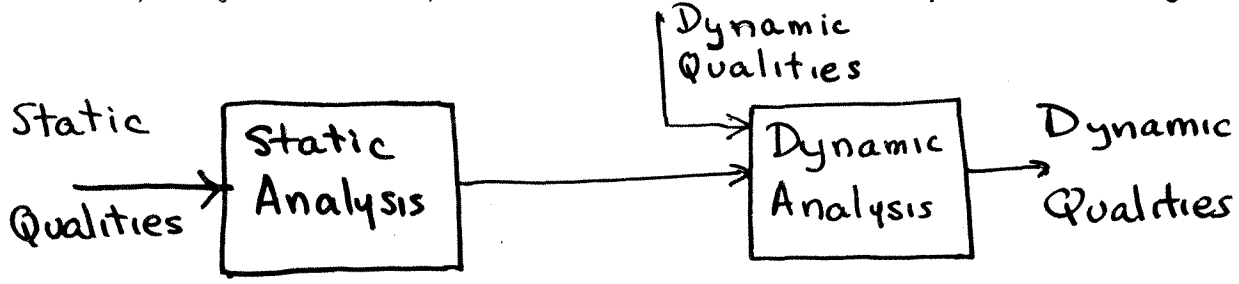
vertical ∧ on → fall

Here incline, horizontal and vertical indicate the three different kinds of slope cases I,II and III; ϵ indicates no action.

The point usually has a non-zero velocity and these rules do not apply. When the point has a velocity we must ask how does this velocity change. In order to do this the qualitative analysis must consider the velocity of the point (i.e. dynamic qualities).



The static analysis actually described how the velocity would change for any value of the velocity, not just for a velocity of zero. This clarifies the structure of qualitative knowledge:



The static model can be easily modified to take into account that the actions act on velocity.

(1) below → fall

(2) off → fall

(3) incline ∧ above ∧ on → possible[slide-d]

(4) horizontal ∧ on ∧ velocity-? → possible[slide-?]

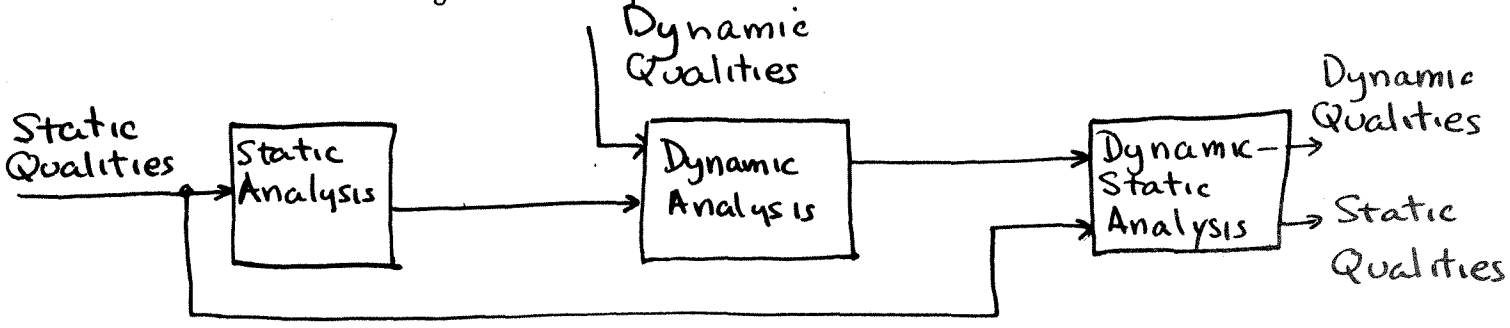
(5) vertical ∧ on → fall

The question mark can match either a d, z (zero velocity) or a u, but all later question marks in that particular production must be bound to that matched value. The function **possible[*x*]** puts *x* on the possibilities list, making *x* one of the possible final outcomes. The items put on the possibilities list can match in later productions.

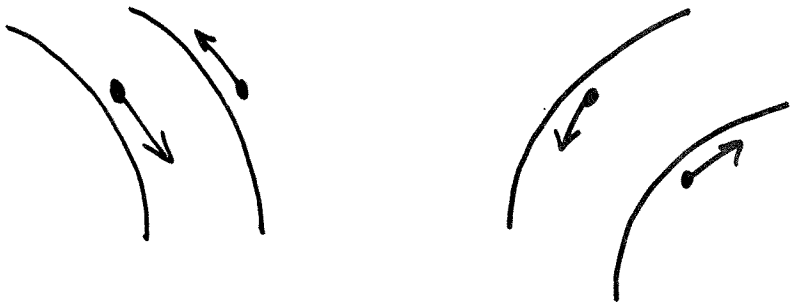
The actions of these rules can affect the velocity. The slope of the surface can either speed up or slow down the velocity of the point mass. If the object is being decelerated the velocity may change at some point and and object may start acelerating downwards. This is stated by rule (3). The other possibility that the velocity might not change must also be taken into account:

(6) velocity-u ∧ slide-d → possible[slide-u] .

To study the dynamics of this domain the model must contain a rudimentary understanding of time. The basic notion is that as time passes, the actions produced by the static analysis will modify the environment (position) of the point mass. This change in environment may or may not cause an immediate change in the static qualities.



We will first consider the case where immediate changes in static qualities occur. A change in slope is not reflected in the relations. The slope relation of incline,horizontal,vertical reacts only to singularities of the slope. The model does not know that a point travelling on top of a convex surface may fly off and a point travelling on the bottom of a concave surface may stay on.



may fall off may stay on

This is similar to the problem of the corner which will be discussed later. Two other rules need to be added to accommodate this knowledge.

(7) convex ∧ on ∧ above ∧ slide-u → possible[fall]

(8) convex ∧ on ∧ above ∧ slide-d → possible[fall]

(9) concave ∧ on ∧ below ∧ velocity-? → possible[slide-?]

In the case where the actions determined by the static analysis do not cause a change in static qualities a simulation must be done. To accomplish a small time step simulation all that remains to do is to apply the actions to obtain a succeeding environment. The only possible actions are **slide**, **fall** and ϵ(no action). The meanings and effects of these actions are obvious.

In order to complete a prediction, the model must be able to handle more than just infinitesmal time changes. One way to do this might be by doing the time step analysis repetitively. However, to treat the passage of time, induction can be used. The basic idea is to do a time step simulation, apply the actions and then check if the relations between the surface and point mass remain unchanged. Again the principle of continuity and discontinuity can be used. It can be assumed by induction that the actions for every time step analysis will be the same until the actions cause a prerequisite for the time step analysis to change. The environment can then be searched using the repeating actions to find when the prerequisite will be violated. Such points usually occur at the singularities of the surface. The description of the environment used by the model will facilitate this kind of search.

This time step analysis can be seen more than just the initial time step simulation on the segment. Rather, the time step analysis examines a typical section of the segment. The analysis may sometimes predict more than one possibility and these possibilities remain true over the entire segment. This myriad of possibilities can be simplified by considering that the event might occur once but an arbitrary time.

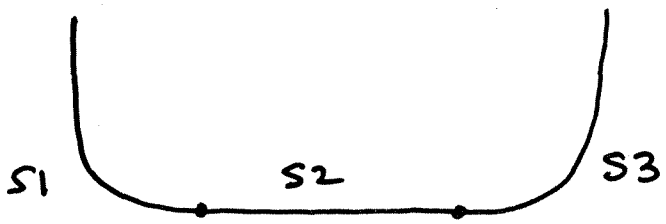
The model assumes that the surface is described by a sequence of segments having the same kind of slope and concavity. The zeros and mathematical discontinuities of the slope and concavity are treated as the singularities of the surface. Each such corner has a description associated with it. A segment is described by:

(SEGMENT name slopetype concavity)

slopetype∈{incline,horizontal,vertical}

concavity∈{minus,zero,plus}

The segments of a simple surface are described. (left to right)

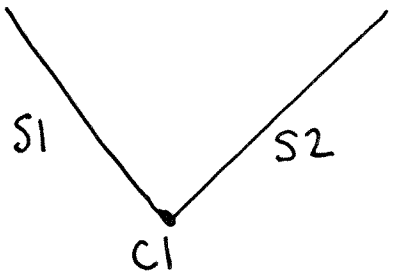


(SEGMENT S1 INCLINE PLUS)

(SEGMENT S2 HORIZONTAL ZERO)

(SEGMENT S3 INCLINE PLUS)

The corners need only be described by the change of slope occuring there. A complete description of a simple scene:



(SEGMENT S1 INCLINE ZERO)

(CORNER C1 MINUS)

(SEGMENT S2 INCLINE ZERO)

With the environment described in this way, the concept of induction can be used very profitably to simulate an event taking place over a segment; if anything is going to happen, it will happen during the first time step simulation. Variations in the situation occur only because of singularities in the prerequisites, and most of these are incorporated into the explicit description of the environment. So, to simulate the movement of the point mass, a time step simulation is done, and then the next corner is selected. As collisions between the point mass and the surface are not handled, only certain kinds of corners are understood.

(10) dslope-zero → ∈

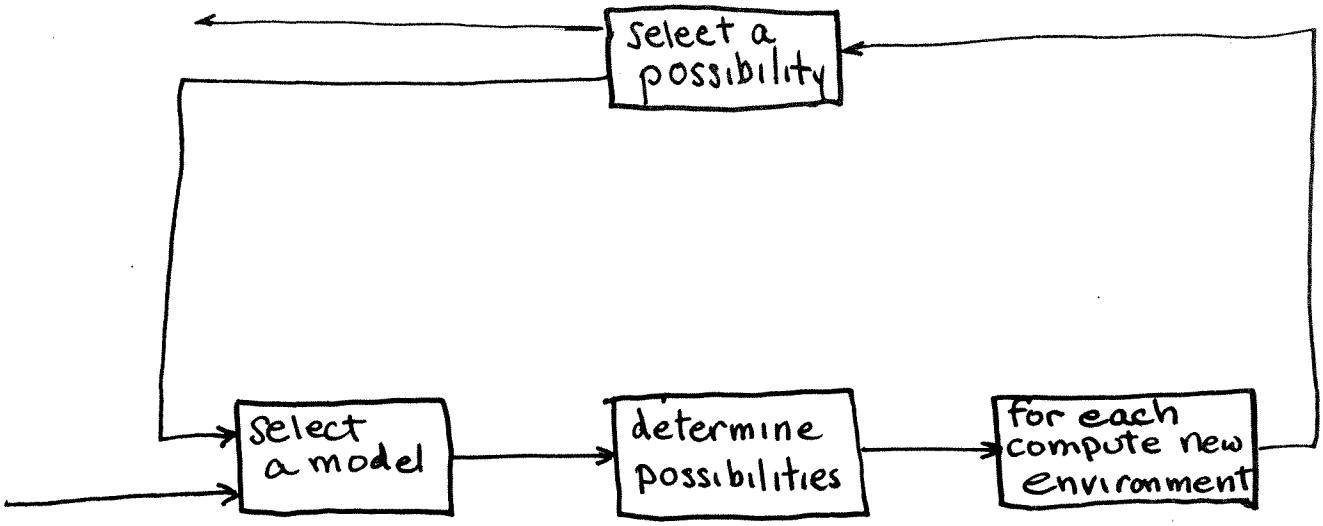
(11) above ∧ dslope-plus → fall

(12) above ∧ dslope-minus → collide

(13) below ∧ dslope-minus → fall

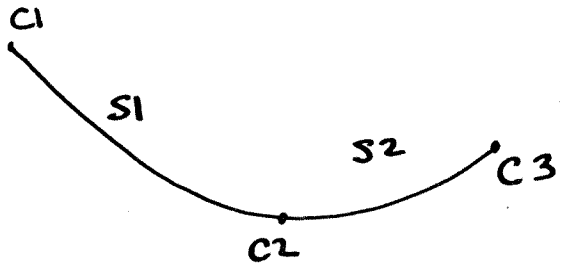
(14) below ∧ dslope-plus → collide

The models for corner and segment have been described. The interaction of these two models predicts all those possibilities which might occur with only the originally given qualities being true. The analysis proceeds as shown in the diagram:



A complete qualitative analysis results in a tree indicating all the possibilities.

Consider the problem of section 2.1:



The mass is placed at the start of segment S1 with zero initial velocity. The initial state is then:

{on,above,velocity-z}

(SEGMENT S1 INCLINE PLUS)

In the static analysis production (3) applies, adding to the state and possibility list the quality slide-d. No other productions in the static or dynamic analyses apply. This single possibility results in the movement to corner C2.

{on,above,velocity-d}

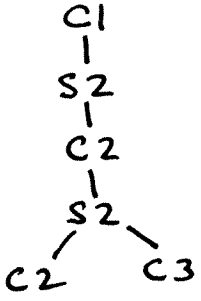
(CORNER C2 ZERO)

Only production (10) of the corner model applies, so the next segment is selected. Since a downward velocity from S1 and across C2 implies an upward velocity on S3, that quality must be changed.

{on,above,velocity-u}

(SEGMENT S3 INCLINE PLUS)

Only production (3) of the static immediately analysis applies, however, the production (6) of the dynamic analysis then also applies. This results in two possibilities **slide-d** and **slide-u**. These two possibilities result in the selection of C2 or C3. This short scenario has resulted in the generation of a tree:



Since oscillation is a possibility, this tree is in fact infinite. Fortunately, oscillation is easily detectable. Oscillation occurs when a previous state of relations, velocity direction and segment repeats.

3.0 THE UTILITY OF QUALITATIVE KNOWLEDGE

3.1 The Necessity for More Knowledge

The qualitative knowledge presented in the previous chapter was predictive in nature; in a given situation, the possible subsequent actions can be determined. This ability plays a major role in qualitative reasoning but more knowledge than just the set of productions of the previous chapter is needed. A second ability, the ability to communicate with the external world and other forms of knowledge, is required.

The models are able to use the productions to simulate the action over time. But how does that help in answering questions? Or, in other circumstances how can and should a specific production be used (other than in simulation).

Communication with the external world is the responsibility of the qualitative knowledge. The diagram of the situation, the description of the problem, the identification of the variables and the values of variables all must be understood by the qualitative knowledge.

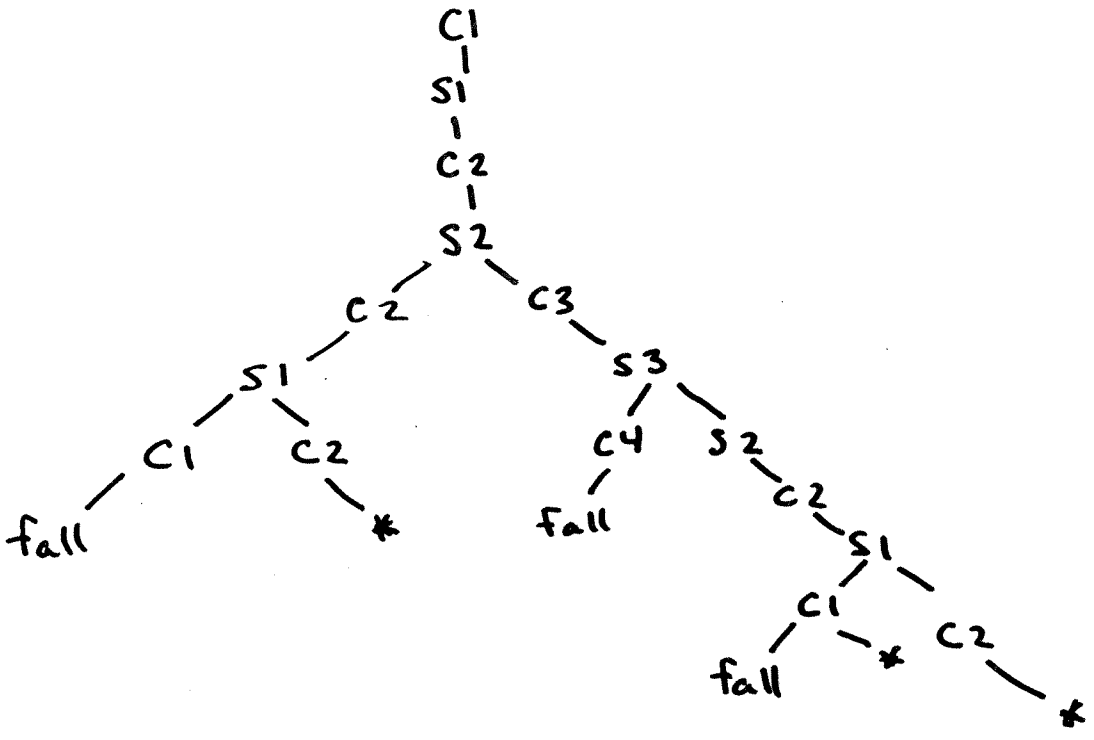
The qualitative knowledge must also be able to understand the question it is being posed. And as the problem will often require recourse to quantitative reasoning, it must be able to communicate specific problems to the quantitative knowledge. This also requires communication of certain variables and values given in the original problem which the qualitative knowledge knows nothing about.

In the remainder of this chapter we will discuss these problems in greater depth and present specific solutions to each of them.

3.2 Answering Qualitative Questions

We must be able to answer questions. This section describes how the productions of the previous chapter can be used to solve simple problems in the roller coaster world.

The possibility tree of the second problem of 2.1 looks like:



We can use this tree to answer simple questions.

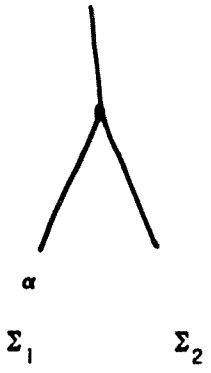
The simplest kind of request is simply for a description of the action. "What happens?" A top down scan of the tree gives such a description. The segments marked with an asterisk indicate that oscillation occurs there as a possibility - the tree is in fact infinite. Each node with multiple branches indicates that the qualitative knowledge could not determine which of the possibilities occurs, such information has to be included in the reply. Many "What Happens?" type questions are very simple or are slight modifications of this basic type: "What happens next?" Both can often be answered by qualitative reasoning alone.

Some questions require a search of the possibility tree, looking for specific conditions being true. "Is x reachable?" is such a question. Answering this question has subtle difficulties. Since the qualitative knowledge is incomplete there is a choice between three answers: *yes*, *no* and *maybe*. A response of *maybe* indicates that only the quantitative knowledge can decide whether the answer is *yes* or *no*. The exact reason for the *maybe* reply must be obtained and used to communicate the precise problem to the quantitative knowledge.

An initial approach might be to search down the tree and check if the first node with

multiple branches is encountered before the desired node (the point in question) is discovered. (The surface must be a path, not a closed curve.) This would succeed in distinguishing the *yes* answers from the *no* and *maybe* answers.

Any branch whose sub-branches are non-trivial (not fall or collide) must be a decision between velocity directions. Once a velocity in a certain direction along a path is established it continues as a possibility until the end of the surface is reached or it falls off This means that one branch can never reach the first point of the other branch.



Assume that the desired point is in Σ_1, then every path to it must go through α. (Again circular paths are ruled out.) Since not every possibility in Σ_2 reaches α, we see that the reach question, upon discovering a branch is perfectly correct in answering not-*yes*. This procedure identifies all the not-*yes* possibilities.

The procedure fails in distinguishing *maybe* and *no* answers. *No* answers can be identified since all possible paths must not reach the questioned point. If these procedures do not result *yes* or *no* answer, the answer must be *maybe*. The difficulty is in identifying the exact cause for the *maybe* answer. The more careful the identification of the cause is, the less calculation the quantitative knowledge must do. For example, in the previous velocity branch example, a *maybe* might be generated, but within Σ_2 all but one possible path might reach α, and a calculation about the original velocity branch could be futile.

The following is an outline for a procedure which meets the above problems. It answers *yes*

no or *maybe* correctly, and for a *maybe* answer it returns a specific point about which will be useful to do the quantitative analysis.

If the current point is the desired point, return *yes*. If the point has no branches, return *no*. Now the branches can be examined recursively. Examine the first one. If the recursive call returns *maybe*, return *maybe*. If the returned value is *yes*, continue examining the other branches recursively; if a non-*yes* is found, return *maybe* and mark the current point. Similarily, if the first recursive call returned *no*, continue, looking for a non-*no*. The marked point is where quantitative knowledge is required.

When the quantitative knowledge identifies the correct possibility, this procedure is reapplied. In fact, it does not really matter who prunes incorrect branches from the tree, external suggestions and givens could also do it. Also note that the entire tree need not be generated, only those of interest to the procedure.

Many other kinds of questions can be constructed from these two primitive types. "Will it reach x before ... ?" "What happens when ... ?" The methods developed in this section are applicable to these kinds of questions.

The detailed analysis required to be able to answer such simple questions demonstrates that specific knowledge about questions is necessary. Currently, we are only concerned with these kinds of questions and simple modifications of them.

3.3 The Necessity of Quantitative Knowledge

The analysis of questions indicated the exact role for the quantitative knowledge played in their solution. Quantitative knowledge is necessary to resolve between possibilities - quantitative reasoning is necessary to resolve qualitative ambiguity. The possibilities are specific and small in number so that the quantitative knowledge knows exactly what it has to distinguish between.

Some questions inherently require quantitative knowledge. "At what time does it reach S1?"

"What happens when the velocity reaches 10?" These questions require qualitative knowledge to be understood, but only quantitative reasoning can answer the question. The qualitative knowledge can of course identify nonsense questions. "At what time does it reach 5I?" Where 5I can be seen to be unreachable by qualitative reasoning alone.

3.4 Communication Between Qualitative and Quantitative Knowledge

The roles of qualitative and quantitative knowledge in problem solving have been explained. The qualitative knowledge can understand the overall problem, but it can never understand it completely. The quantitative knowledge can completely understand a specific local problem, but it can never understand the overall problem. These two kinds of knowledge complement each other and together they can completely solve entire problems. This requires communication.

Communication, required to handle the incompleteness of the different kinds of knowledge, requires special representations to take into account communication and specific interfacing knowledge (knowledge about communication).

The qualitative knowledge decomposes the problem into specific sub-problems, it must be able to pass these to the quantitative knowledge. In return, the quantitative knowledge must be able to give a result which is comprehensible to the qualitative knowledge. Or, if the problem cannot be solved, it must return an explanation about why the problem could not be solved. The qualitative knowledge can then try to resolve some of these impediments to the solution.

In one direction communication consists of variables (with their values) and purposes. These purposes are requests for the values of specific variables. The response consists of results -- variables with assigned values, and complaints -- explanations for the failure in terms of variables.

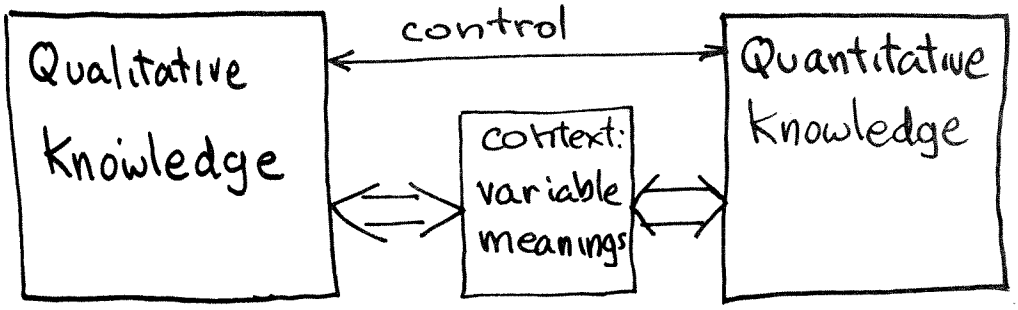
One solution which meets these criterion is allowing the qualitative and quantitative knowledge to know the names of the internal variables and procedures of the other. Such a solution, however, lacks generality -- each representation must know about the internal workings of the other.

The critical concept is that of a variable. In the above solution a variable only has meaning in the sense of how both forms of knowledge reference it. A much better idea is to put the meaning explicitly with the variable and then forcing the representations to look at the meanings of variables. Thus only meanings of variables and not variable names are communicated. Qualitative and quantitative knowledge can then be independent of each other, each faithful only to the meaning of the variable, not to where it came from or who references it.

These meanings can be stored in a CONNIVER-like data base. The qualitative knowledge communicates the problem to the quantitative knowledge by providing it a context of meaning - value pairs and a purpose expressed as the meaning of the variable whose value is required. Results are easily returned in the context itself and complaints can be expressed in terms of variable meanings, without having to be concerned about whether they can be understood.

The qualitative knowledge can use these contexts to analyze the problem over time, changing contexts for different sections of the surface. This allows previous results to be remembered and reconsidered.

By giving variables meaning, quantitative problems can be handled in more general terms. Knowing the nature of the givens, and the meaning of the unknown makes it easy to determine what kind of computation is required. Only the general domain of the question and the different possibilities which need to be distinguished between are required to solve the problem: "This is a question about the movement of X on surface Y; will the mass change direction?" Since the number of possible question domains and the number of different kinds of qualitative ambiguity are so small, the different combinations can be accounted for by assigning them specially named requests points in the quantitative knowledge.



There remains the problem of actually developing the meaning descriptions and determining the nature of the requests points which are needed. Both of these issues are discussed in more detail in the next chapter where the specific representation of quantitative knowledge is considered.

4.0 REPRESENTATION OF QUANTITATIVE KNOWLEDGE

4.1 General Overview

As we have seen in the previous chapter many questions cannot be answered by qualitative knowledge alone and require the solving of equations. The choice of the relevant equations and their subsequent solution is the task of the quantitative knowledge. This chapter presents a theory about how such knowledge can be represented and how it interacts with the qualitative knowledge.

The fact that a symbiosis of qualitative and quantitative knowledge exists puts specific constraints on the representation of the quantitative knowledge. The major constraint is that there must be a facility for communication: (1) the other kinds of knowledge must be able to ask questions of the quantitative knowledge, (2) the quantitative knowledge must be able to return solutions that are understandable and (3) if for some reason the quantitative knowledge fails to be able to answer the question, it should be able to give an explanation for its failure. This communication allows the quantitative knowledge not to be concerned about (1) discovering the problem in the first place and (2) issues of searching large spaces of strategies, since the communication will have restricted the space considerably.

As with the qualitative knowledge, we would like to have the representation have some semblance to what is seen in people. It may be argued that people always think qualitatively about equations. However, from observation and introspection, the dichotomy of qualitative and quantitative gives a natural description for what we see. Qualitative reasoning involves studying gross behavior in time and space while quantitative reasoning is usually characterized by assigning variables to the observations made by qualitative reasoning, ignoring much of the meaning of the variables, artificially manipulating equations consisting of these variables and reassigning meaning to the results. This loss of meaning of variables leads to abstraction in the manipulation process and gives mathematics its generality and power. Unfortunately, this loss of meaning makes the transferences and selection of relevant equations very susceptible to error. Quantitative knowledge handles these transferences and the selection of equations.

For some domains it is possible to develop a completely uniform quantitative reasoning

strategy such as a circuit simulator for an electronics domain <Brown *et.al.*, 74>. Few domains have this property and solutions to specific questions are often derived inefficiently and have unnatural derivations. Rather, the quantitative knowledge for a domain is better represented as a collection of inter-dependent strategies. As opposed to a uniform quantitative strategy, each of these simpler strategies specializes in certain aspects of the general problem being fixed, thus allowing for elegant and more natural solutions. Each of these strategies has conditions for which it is applicable, and therefore a method to express these conditions must be possible in the representation.

We propose a frame organization to meet the above criteria. The descriptions are organized such that collections of knowledge about a certain situation are put together into a frame. In the proposed system one such frame will be knowledge about the trigonometric relationships within a triangle. A frame is not a procedure in the conventional sense, instead it describes dependencies and assignments between variables. These frame descriptions are examined by a general reasoning strategy (REASON) in order to answer questions. Frames can only be invoked for a definite purpose which is to discover the value of a particular variable. This purpose must be made explicit when the frame is invoked. So in an abstract way invoking a frame means that the invoker knows that it requires the value for a certain variable and that the frame it is attempting to invoke knows something about that particular variable.

When a frame is invoked, REASON looks at the purpose for the invocation and then examines the frame to see where in its description it makes a possible assignment to that variable. This possible assignment can either be an explicit assignment or a reference to another frame. This assignment may have a number of prerequisites which have to be met. Many possible assignments may be found, each with different prerequisites. One of these must be chosen. Much of the syntax for frames involves how such preconditions can be structured.

Frames act at various levels of the problem solving process: one frame might be concerned about using an energy method versus a Newton's Law method while another frame might be concerned about the base length of a sliding block. In a usual LISP-like representation we would

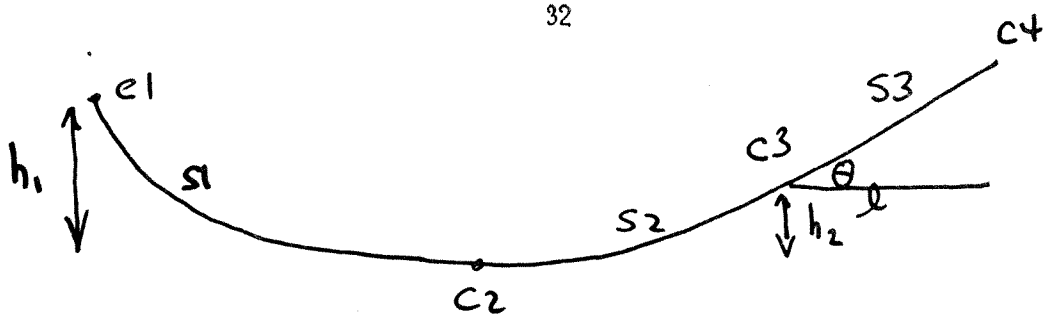
either have to pass down all relevant variables from the top-level or use free variables. Passing variables down from higher levels has the problem that the top-level frames have to be concerned about many details which are irrelevant to them. Free variables have the problem that more than one frame must know their names. The frames we propose here are not invoked with argument lists, neither do they explicitly refer to free variables. Instead, the name of a variable is only local to frame that defines it. Communication is achieved by assigning meanings to every variable and constructing bindings between variables with matching meanings. These meanings are designed to be understood by the qualitative knowledge. This solves the problem of variable names and makes it possible for the qualitative knowledge to examine the state of a quantitative computation.

4.2 A Scenario using Quantitative Knowledge

This section presents a scenario of how the quantitative knowledge can be used to solve a simple problem. We will solve a "Will it reach?" problem using the results of the discussion about the qualitative knowledge for this question in section 3.2 as a starting point (such as the possibility tree). The quantitative knowledge necessary to solve this problem is given in the representation we are trying to develop. 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.

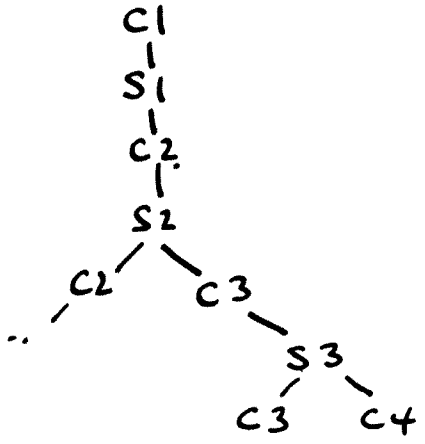
```
(DEFINE-FRAME  MASS-MOVEMENT  (?OBJECT ?SURFACE ?T1 ?T2)
                              ((A (ACCELERATION ?OBJECT)))

        (FCOND ((FEQ (TYPE ?SURFACE) 'STRAIGHT)
                ;if the surface is flat, try simple kinematics
                (FPROG (THETA (ANGLE1 ?SURFACE))
                       (FRAME RTRI (?SURFACE))
                       (FRAME KIN (?OBJECT ?SURFACE ?T1 ?T2))
                       (VSETQ A (TIMES %G (SIN THETA))) )))
        (FRAME ENERGY (?OBJECT ?SURFACE ?T1 ?T2)))
        ;energy will work for arbitrary shapes

(DEFINE-FRAME  ENERGY (?OBJECT ?SURFACE ?T1 ?T2)
                      ((VI (VELOCITY ?OBJECT ?T1))
                       (VF (VELOCITY ?OBJECT ?T2))
                       (H (HEIGHT ?SURFACE)))

        (EQUATION (PLUS (TIMES -1 VF VF) (TIMES VI VI)
                        (TIMES 2 G H))))
        ;vf^2 - vi^2 = 2 g h

(DEFINE-FRAME  RTRI (?TRIANGLE)
                    ((H (HEIGHT ?TRIANGLE))
                     (L (BASE ?TRIANGLE))
                     (HYP (DISTANCE ?TRIANGLE))
                     (T1 (ANGLE1 ?TRIANGLE))
                     (T2 (ANGLE2 ?TRIANGLE)))

  (EQUATION (PLUS (TIMES -1 HYP) (SQRT (PLUS (TIMES H H)
                                             (TIMES L L)))))
  (EQUATION (PLUS (SIN T1) (TIMES -1 (QUOTIENT H HYP))))
  (EQUATION (PLUS (SIN T2) (TIMES -1 (QUOTIENT L HYP)))) )

(DEFINE-FRAME  KIN (?OBJECT ?SURFACE ?T1 ?T2)
                   ((VF (VELOCITY ?OBJECT ?T1))
                    (VI (VELOCITY ?OBJECT ?T2))
                    (D (DISTANCE ?SURFACE))
                    (T (TIME ?T1 ?T2))
                    (A (ACCELERATION ?OBJECT)))

  (EQUATION (PLUS VI (TIMES A T) (TIMES -1 VF)))
  ;vf = vi + a t
  (EQUATION (PLUS (TIMES VI VI) (TIMES 2 A D) (TIMES -1 VF VF)))
  ;vf^2 = vi^2 + 2 a d
  (EQUATION (PLUS (TIMES VI T) (TIMES .5 A T T) (TIMES -1 D))))
  ;d = vi t + .5 a t^2
```

The problem is:



Will the object reach C4?

The qualitative knowledge determines that the object must slide down S1 without falling off, then slide up S2 where it might slide back or reach S3 on which it also might slide back.



The qualitative knowledge determines that there is never a possibility that the object will fly off. Also, it isolates the problem into the two necessary subproblems: first it must be determined whether the object reaches S3 and then whether it slides back on S3.

First the velocity at the end of S1 must be found. To do this MASS-MOVEMENT must be invoked:

(MASS-MOVEMENT (B1 S1 TIME1 TIME2))

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

(VELOCITY B1 TIME1) - known

(VELOCITY B1 TIME2) - unknown

(ACCELERATION B1) - known

'Unknown' indicates that this is the value MASS-MOVEMENT is required to find. When

MASS-MOVEMENT is invoked REASON attempts to find a value for this variable. By searching through the FRAME's it is found that there are two places in MASS-MOVEMENT in which a possible assignment to VF take place. REASON quickly discovers that the assignment in the FCOND is impossible to reach as (FEQ (TYPE ?SURFACE) 'STRAIGHT) is unsatisfiable. The only alternative is the ENERGY frame. When ENERGY is invoked it complains back that it was given insufficient information, namely it was not provided with the HEIGHT. Every possible attempt to achieve a value for VF has now failed, the alternative is to examine the reasons for these failures and attempt to eliminate them. The complaint blocking the only path to VF is that HEIGHT is unknown. There are no accessible references to HEIGHT in the frame so it fails back to the qualitative knowledge complaining that it was not provided with height of the surface. The qualitative knowledge must now find the height or find the problem impossible to solve. Assume the height is computable by some other means. It can set HEIGHT and continue. REASON can now complete its examination of ENERGY and return a value for the unknown VF which MASS-MOVEMENT subsequently returns to its caller. This value is remembered and the segment S2 is examined in much the same way using the VF on S1 as VI on S2:

(MASS-MOVEMENT (B1 S1 TIME2 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 REASON has two possible paths to a solution. If ENERGY is tried it fails because HEIGHT is unknown. Since (FEQ (TYPE ?SURFACE) 'STRAIGHT) is satisfied KIN can be tried immediately for a solution. KIN complains back that it must know the value for either D or T before it can proceed. Again every path to VF is blocked and these complaints must be examined to see if they can be satisfied. The two alternative complaints that block the path for finding VF are (1) find HEIGHT and (2) find D or T. There are no other references to T in the frame 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, of course there is not enough information to solve for the variables so complaints are generated back up to the qualitative

knowledge. The qualitative knowledge sets values for TI and L in the instantiation of RTRI on S3 and REASON proceeds. Now RTRI returns with values for both D and HEIGHT. REASON has a choice whether to restart 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 C4 is reachable has been answered.

4.3 Frames as a Representation

Any frame under consideration by REASON has an *environment*. This environment contains the current variable bindings (A-LIST) and the current state of the computation. The same frame may be under consideration with a number of different environments. A frame combined with a particular environment is an *instantiation* of that frame. So far this paper has often used frame to mean an instantiated frame, the context of the word frame will indicate which meaning is intended.

An instantiated frame can be in a number of states. It can be in a SUCCESS state which means it has fulfilled its purpose. Frames in SUCCESS states are interesting since they might still be invoked later on for a new variable and they give a derivation of the result. Frames can also be in the ACTIVE state which means that even though it has not yet fulfilled its purpose, it has not yet tried all the possibilities open to it. A frame reaches a PASSIVE state when it has tried every option possible to achieve its purpose and has discovered it cannot proceed unless it is given more information. A frame which could not meet its purpose and has no possibility to fulfill its purpose is in the FAILED state. This means that that particular instantiation can shed no more light on its purpose at all. A frame can get into a FAILED state for two reasons; one is that the values provided by the caller make the problem impossible to solve, the other is that every possible attempt to resolve the failures of the PASSIVE state has completely failed.

Whenever a frame is instantiated it must be given an instantiation pattern, a short description of the scene about which the frame is solving problems. For example the KIN frame is given a four-tuple indicating the object about which it is solving kinematic problems, the

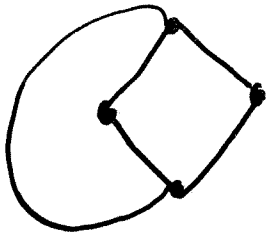
surface this object is moving on and the times of this movement. This pattern serves two purposes. First it gives a name to every instantiation so that all references to a frame with the same scene will be to the same instantiation. So, the same instantiation can have multiple invokers. Secondly, this name is used to give meaning to the secondary variables. If a secondary variable is to have a non-local binding it must be given a meaning which can match with other meanings. These meaning descriptions contain primary variables which must be assigned the names of the specific objects under consideration. In the description of the secondary variable VF in KIN, the primary variable ?SURFACE must be assigned the value S1. The values for the variables are derived from matching the pattern at the beginning of the frame with the instantiation pattern.

Since the meanings of these secondary variables cannot be elaborated unless all the primary variables have assigned values, a frame cannot be instantiated unless all if its primary variables are given values.

The format for instantiation patterns and meanings is very simple, and there is no possibility for fuzzy descriptions. Descriptions must match exactly, or they don't match at all. The format of instantiation patterns is arbitrary, but the invoker and invokee both must know this format. The descriptions of variables have fixed formats and a variable can only be of certain types. A meaning description consists of a type such as VELOCITY or ACCELERATION and an ordered list of modifiers which indicate such information as the name of the moving object or the time span of the movement.

In order to keep track of variable meanings and instantiation patterns a simple data base is required. Since all the descriptions must match exactly, just the ability to put property lists on arbitrary lists is sufficient. Whenever a new secondary variable reference is made (either through FRAME or FPROG) the meaning description is elaborated by substituting for the primary variables and then looked up in the data base to find its property list. So the A-LIST for an instantiation contains local variable name - global property list pairs, not local variable - value pairs. It is the data base that really describes the current environment and not any A-LIST or stack structure. Thus, unbinding in the conventional LISP sense becomes a vacuous concept.

At any point in the problem solving the reference links between instantiations form a graph with a distinguished node.



Such a reference link can only be created by a FRAME. Every time a reference link is created the variable meanings for the referenced frame are matched into the data base. (In the remainder of this chapter variable will be used to refer to secondary variable unless explicitly indicated otherwise.) Added to the property list of a variable is the fact that there is a new frame which knows something about that variable. Associated with that fact is a reference path or plan for how to reach that instantiation. So whenever a value for a particular variable is required its property list can be examined to find where it is referenced. If all these references fail, other FRAME's can be looked for in other referencable instantiations and new references created.

Whenever an individual attempt to find a value for a variable fails there is the choice of trying to deal with the failure or trying another alternative. We will take the general position that computation should remain local for as long as possible. This means that alternatives outside (above) an instantiation should be considered only after all attempts to resolve the failures have also failed.

This control structure will be clarified with some examples. Consider the case where REASON is considering a frame A and a purpose and A references another frame B which might fulfill this purpose. So B is attempted and returns in a FAILED or PASSIVE state. If B reaches a PASSIVE state, it returns a complaint list back to A. This list consists of predicate - environment pairs. The predicate indicates the condition that must be met before B will be able to proceed in achieving its purpose. Processing can resume with the environment provided in the second half of the pair. Each such predicate - environment pair suggests an independent alternate

way for B to achieve its purpose.

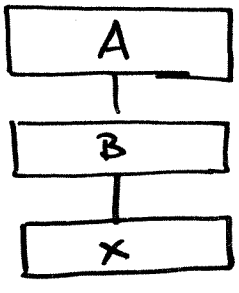
As a consequence of this mechanism the same instantiation can have many sub-instantiations. The sub-instantiations have the same names, superiors, and A-LIST's as the original instantiation but the state of REASON's consideration of them can be different. For example they can have different PASSIVE frames suspended from them. Note that only one of the sub-instantiations of an instantiation can ever succeed.

Assume for the moment that B has no inferior frames in PASSIVE states. In that case each predicate is simply a conjunct of the form:

$$KNOWN(\alpha_1) \wedge KNOWN(\alpha_2) \ldots \wedge KNOWN(\alpha_n)$$

$\alpha_1, \alpha_2, \ldots \alpha_n$ are variables in A. Each such conjunct indicates that if B was provided with the values of $\alpha_1, \alpha_2, \ldots \alpha_n$ it would be able to proceed toward the purpose A set for it. After REASON has tried all possible paths to achieve its purpose and every path is blocked with a PASSIVE frame, it begins to consider the complaint list. This involves looking at the predicates and choosing to try to achieve one of them. This is done by REASON generating an additional purpose for A. REASON then pushes the old purpose onto a stack and tries to fulfill this new purpose.

If B has inferior frames in the PASSIVE state the predicate it returns is far more complicated.



Let X be one of B's inferiors in the PASSIVE state. It would have returned a complaint list to B, and each conjunct in that predicate could contribute to the reason B returns to A.

4.4 Frame Syntax and Semantics

A detailed description of the syntax of frames will be presented. It will be shown how the semantics meet the conditions of the previous discussion. Some example frames about physics will be shown.

A frame definition consists of a list of specifications and a body.

(DEFINE-FRAME <name> <pattern> <variable list> <body>)

<pattern> is a list of primary variables, upon instantiation these will be bound to values which will be used in the expansion of variable meanings. The <variable list> consists of all the variables that can possibly be bound into the data base.

<variable list>: $((\alpha_1 \pi_1) \ldots (\alpha_n \pi_n))$

The α_i are the names of these variables and the π_i their corresponding descriptions. A variable description consists of a list whose first element is a type and the remaining elements, primary variables. The current types with their modifiers are:

(ACCELERATION object)

(VELOCITY object time)

(DISTANCE surface)

(TIME time1 time2)

(HEIGHT surface)

(BASE surface)

(ANGLE1 surface)

(ANGLE2 surface)

The <body> is formed from a sequence of forms. Abstractly these forms are a unordered set of non-conflicting pieces of knowledge about the variables. The forms can be considered in any order or in parallel. REASON decides which of the forms should be considered and in what order.

The most basic form relating variables is the EQUATION:

(EQUATION <expression>)

<expression> is an arithmetic expression in conventional LISP notation giving an arithmetic relation between the variables. The convention is that <expression>=0. Usually symbol manipulation must be performed in order to solve for the unknown variable. If this form fails it returns a complaint list just as a conventional frame would. A simple kinematics frame could be given by:

```
(DEFINE-FRAME  KIN (?OBJECT ?SURFACE ?T1 ?T2)
                        ((VF (VELOCITY ?OBJECT ?T1))
                         (VI (VELOCITY ?OBJECT ?T2))
                         (D (DISTANCE ?SURFACE))
                         (T (TIME ?T1 ?T2))
                         (A (ACCELERATION ?OBJECT)))

      (EQUATION (PLUS VI (TIMES A T) (TIMES -1 VF)))
      (EQUATION (PLUS (TIMES VI VI) (TIMES 2 A D) (TIMES -1 VF VF)))
      (EQUATION (PLUS (TIMES VI T) (TIMES .5 A T T) (TIMES -1 D))))
```

Suppose KIN was invoked with VF and VI known and A unknown with D and T given don't care but bound values. KIN might return a complaint list consisting of three pairs:

KNOWN(D)

KNOWN(T)

KNOWN(D)∧KNOWN(T)

It is possible to explicitly give a variable a value:

```
(VSETQ <variable> <expression>)
```

The VSETQ will fail back with a complaint list if any of the variables <expression> references are not known. Knowledge about an object moving along a straight line might be given by:

```
(DEFINE-FRAME  STRAIGHT (?OBJECT ?SURFACE ?T1 ?T2)
                            ((VF (VELOCITY ?OBJECT ?T1))
                            (VI (VELOCITY ?OBJECT ?T2))
                            (D (DISTANCE ?SURFACE))
                            (T (TIME ?T1 ?T2))
                            (A (ACCELERATION ?OBJECT))
                            (THETA (ANGLE1 ?SURFACE)))
          (VSETQ A (TIMES %G (SIN THETA)))
       ... )
```

So if either the angle or the acceleration were given this frame could handle it.

To reference another frame the FRAME form is used:

(FRAME <name> <instantiation-pattern>)

<name> is the name of the referenced frame and <instantiation-pattern> a list of primary variables describing the scene which will become the name of that instantiation.

Local variables can be created by the FPROG form. The syntax of the FPROG is similar to that of a DEFINE-FRAME definition.

(FPROG <variable list> <body>)

In order to express dependency relations between assignments the FCOND form is used.

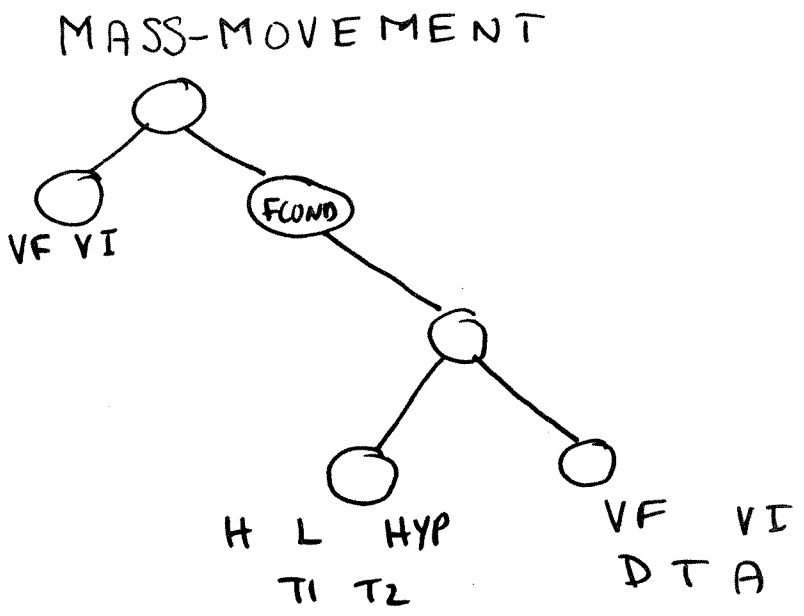
```
(FCOND (<condition> <body>)
       ...
       (<condition> <body>))
```

When the <condition> is met the <body> can be considered. If the <body> fails to achieve its purpose and another pair is tried the effects of the previous pair must be undone. (i.e. in particular VSETQ's must be undone.)

4.5 Frame Control Structure

So far in this chapter the problems of control structure have been glossed over. Many of the details will be presented here. Much of the control structure is intimately related to the organization of REASON.

Every frame description gives a syntactic description of the dependencies between variables, these dependencies give an initial plan to achieve the purpose of the instantiation. These dependencies can be conveniently represented as a tree:



The variables at the nodes indicate that possible assignments to those variables can occur there. Let us assume for the moment that the purpose of the instantiation is a conjunct on these variables (using KNOWN). The tree can be scanned for various ways to achieve this conjunct. Since the tree may contain FCOND's with multiple pairs some conjuncts may be unachievable even though their component variables might be individually achievable. The plan suggested by this scan of the tree is then used. When a particular node fails to achieve a purpose it can be marked with the purpose it failed on and the reason why it failed. Also the property list of the desired unknown variables should be marked, so that whenever the value for this variable is discovered for any reason, this failure can be reconsidered.

At some point these reasons for failures can be examined to see if any of them can be satisfied. The collection of all these failures for the original purpose gives the reason why this purpose could not be satisfied. What we would like to do here is to make this collection our new purpose, but the only kind of purposes we can handle are conjunctions. This collection forms a predicate containing conjunctions and disjunctions, so it can be transformed into disjunctive normal form and each individual conjunct will be a valid purpose. REASON decides the order in which these conjuncts should be tried. If these new purposes generate other purposes these can be understood as substituting these other purposes back into the original failure collection,

recomputing the disjunctive normal form and continuing. Every time this happens we have decomposed that conjunct into simpler conjuncts. To prevent infinite looping a record must be kept of each purpose that was tried and failed so that it will not be reattempted.

The only two forms which can assign values to variables are the EQUATION (which does an implicit VSETQ) and the VSETQ. The act of giving a variable a value has many ramifications. It has the immediate effect of meeting the current purpose REASON is considering, but many other purposes may also be blocked because this variable was unknown. All those purposes which were blocked by this variable can now be reconsidered. Clearly the topmost level purposes which are affected by this assignment should be tried first.

The fact that VSETQ's have such global effects for both the current frame, its superiors and its inferiors makes backing up over VSETQ's very difficult. To undo a branch of a FCOND not only the values of the VSETQ's have to be undone but also the reference links and complaint requests on variables have to be removed. This could be handled with a CONNIVER-like PUSH-CONTEXT mechanism. Each pair of the FCOND requires an independent PUSH-CONTEXT of the original environment. This keeps the pairs of the FCOND completely disjoint, yet allows for the conventional complaint mechanisms to work. Unfortunately discoveries made in one context which were independent of the VSETQ have to be rediscovered in the next context. The real difficulty is in the identification of those effects which depended on the VSETQ. This is the McCarthy frame problem.

Fortunately our control structure lends itself to a convenient partial solution to the problem. The basic idea is this. Whenever a VSETQ is made we want to find out in which contexts (currently active) this is true. This does not mean examining the reason why we reached this particular VSETQ. Instead we must examine what conditions made it possible to reach this VSETQ and find the most superior context in which these conditions are true and add the VSETQ to that context (and thus to all its inferiors). Of course if we POP-CONTEXT this context, we will lose the VSETQ. This partial solution solves the frame problem in that a discovery made in a context which was not conditional on that context would not be entirely lost.

Since any currently inactive context must be an inferior of some active context the VSETQ would also be communicated to the inactive context. This is only a partial solution in that the most superior context might be the current context and then although the necessary conditions to the VSETQ might be present elsewhere it would have to be recomputed again there.

Although it is easy to identify what conditions made it possible to reach the VSETQ when it was first reached, later computations may reveal less restrictive conditions for the VSETQ. This means that the VSETQ could hold true in a superior context to that it was in originally. To find the most superior context would mean that in the creation of every reference link a check would have to be made if it established a new path to a VSETQ (not necessarily the same as the original) and then its context computed and the most superior of them chosen. Probably only using the original superior context will be sufficient for most cases, and this is what will be originally implemented.

4.6 A Critique of this Theory

This representation was designed to meet certain criteria, we must examine whether those criteria are still valid in the light of what we have discovered and how this representation meets those criteria.

(1) Can we represent the quantitative knowledge of physics?

(2) Does the organization lend itself to a convenient and intelligent communication scheme with the qualitative and interfacing knowledge?

(3) Can it solve quantitative problems at all?

(4) Are its derivations easy to access and are they reasonably elegant?

(5) What are the features of this representation that make it better than others, than lets say just LISP code, or CONNIVER programs, etc?

The last question is the hardest of all to answer. Why go to such an extent to design a new representation when other already existing representations might work just as well and would require no design? This question has a number of answers: (1) it makes communication between

the different kinds of knowledge very easy and (2) it makes it very simple to codify quantitative knowledge. Quantitative knowledge could be represented as LISP or CONNIVER programs but then we would still have to develop a very extensive communication mechanism to interface conveniently with the other forms of knowledge and it would be necessary always to represent exactly how to solve the problem in every case. In general using LISP or CONNIVER would lead to a very bulky and inconvenient representation. The frame representation has the great advantage that once something is expressed in the representation communication happens automatically and in well defined ways.

In many ways our proposed representation is much simpler than PLANNER or CONNIVER. This simplicity makes it easy to see this representation from a search perspective. The goal is to discover the value for a variable in a space of instantiated frames. Search is not however the dominant paradigm. As with PLANNER this search is so the structured that the structures themselves are the dominant feature, not the fact that a search is taking place. It is this structure that frames represent. In this sense we have a simplified cross between PLANNER and CONNIVER; the automatic goal searching of PLANNER and the context mechanisms of CONNIVER.

There remain a number of details that have not been worked out. The most important of these is specification of initial and final conditions and how this should be represented. In the scenario it was implicitly assumed that the qualitative knowledge made the correct assignments. For example HEIGHT should have been assigned a positive or negative value depending on the direction of travel.

A rather elaborate explanation of REASON was given, some parts of it are very easy to implement, but others are extremely difficult. A decision must be made about how REASON should store the state of an instantiation, and whether this requires a special kind of language such as CONNIVER.

The MASS-MOVEMENT frame knew only about flat surfaces, how would we represent other kinds of surfaces? Every different kind of surface has its parameterized representation,

clearly we don't want a different top-level frame for each different kind of surface.

5.0 CONCLUDING REMARKS

5.1 Discussion of the Theory

In this proposal we have used the dichotomy of qualitative and quantitative knowledge to develop a representation of knowledge which can solve problems in classical mechanics. The idea of this dichotomy has wider breadth than just the domain of mechanics.

The concept has an immediate application to studying the knowledge needed to understand electrostatics or electronics (in fact, to many of the engineering disciplines). As for mechanics, we claim that the understanding of these domains requires two fundamentally different kinds of knowledge. These require radically different representations. Furthermore, we claim that one of the representations must deal with the qualitative aspects of the domain, and the other with the quantitative. The nature of their interactions can also be precisely described. The qualitative knowledge understand the problem from an overall point of view and fails only on details. The quantitative knowledge can handle only those details.

From another perspective we are solving the problem of interfacing two fundamentally different representations. The ideas of limited and specific communication through meanings rather than names has applicability to other areas of artificial intelligence such as natural language and vision.

5.2 Future Work

Aside from generalizability, another test of the theory is whether it can be realized as a computer program. This is exactly what we intend to do. We will not be concerned with natural language understanding at all. The entire problem will be communicated to the program in the method best suited to it. This will probably consist of a simple language of primitives and the ability to initialize contexts (representations' data bases).

Certain theoretical details remain to be resolved. These lie mainly in the communication mechanisms. There is also the more general question of whether the qualitative representation can capture enough of the qualitative knowledge necessary. Section 4.6 discussed some of the problems

which remain with the quantitative representation. All these issues are best resolved by actually attempting to construct the program.

There always remains the temptation to extend the domain. Unfortunately, this usually results in much more work than is originally intended. If it does turn out, however, that the roller coaster world is indeed to simple, it can easily be extended in a number of interestingly different ways. Friction could be added. The point mass could be given a volume, thus adding the difficulty of having to be concerned simultaneously with movement and collision possibilities of the mass (such as in caves). Another movable object could be added, adding the necessity to understand interactions. The ability to handle trajectories could be added. Questions about the surface itself could be allowed. Clearly, there are many interesting extensions, and if the necessity arises the appropriate one can be chosen.

REFERENCES:

<Bobrow, 68>
Bobrow,G.D.,"Natural Language Input for a Computer Problem Solving System", in Minsky (ed.),*Semantic Information Processing*, M.I.T. Press, Cambridge, 1968.

<Brown, 74>
Brown,A.L.,"Qualitative Knowledge, Causal Reasoning, and the Localization of Failures -- a Proposal for Research",M.I.T. A.I. Lab,Working Paper 61,1974.

<Brown *et.al.*, 74> Brown, John Seely, Richard R. Burton and Alan G. Bell, *SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (An example of AI in CAI)*, Final Report, B.B.N. Report 2790, A.I. Report 12, March,1974.

<Charniak, 68>
Charniak, E.,*CARPS, a Program which solves Calculus Word Problems*, M.I.T. Project MAC TR-51,1968.

<Charniak, 71>
Charniak, E.,"A Note on an Electrostatics Problem Solver", M.I.T. A.I. Lab,(unpublished),1971.

<Galileo, 00>
Galileo, G.,*On Motion & On Mechanics*, 1600, Drabkin & Drake (translators), The University of Wisconsin Press, Madison, 1960.

<den Hartog, 48>
den Hartog,J.P.,*Mechanics*, McGraw-Hill, New York, 1948.

<Hewitt, 71>
Hewitt,Carl,"Description and Theoretical Analysis (using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot", M.I.T. A.I. Lab,Memo 251,1971.

<Kleppner & Kolenkow, 73>
Kleppner, Daniel, and Robert J. Kolenkow,*An Introduction to Mechanics*, McGraw-Hill, New York, 1973.

<Levinson, 61>
Levinson,I.J.,*Introduction to Mechanics*,Prentice-Hall,Englewood Cliffs,1961.

<Mach, 60>
Mach,E.,*The Science of Mechanics*,Open Court,Lasalle,1960.

<McDermott & Sussman, 74>
McDermott,Drew, and Gerald Jay Sussman,"The Conniver Reference Manual", M.I.T. A.I. Lab,Memo 259a,1974.

<Minsky M., 73>
Minsky, Marvin L.,"Frame-Systems: A Framework for Representation of Knowledge", M.I.T. A.I. Lab,(unpublished),1973.

<Purcell, 65>
Purcell, M.E.,*Electricity and Magnetism*, McGraw-Hill, New York, 1965.

<Sussman & Brown>
Brown,A.I.,"Localization of Failures in Radio Circuits a Study in Causal and Teleological Reasoning", M.I.T. A.I. Lab, Memo 319,1974.