A program to solve mechanics problems stated in English

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

UNIVERSITY OF EDINBURGH

DAI Working Paper : 8

May 1975

Subject : A program to solve mechanics problems stated in English

Authors : Alan Bundy, George Luger and Martha Stone

1. Introduction

The research question we are proposing to address is how it is possible to get a mathematical representation of a problem from its natural language statement : or more widely, how it is possible to form a mathematical model of a real world situation. These questions are obviously of vital importance if we are to fully understand mathematical problem solving activity.

We believe that this is a neglected area of research, now ripe for development. The only previous work in this area, by Bobrow 1964 and Charniak 1968, is intellectually unsatisfying and can only deal with simple problems.

For example, Bobrow's program, STUDENT, works by translating the English sentences directly into equations:

"The distance between Boston and New York is 250 miles" becomes

"The-distance-between-Boston-and-New York = 250 x miles"

It does this by a rudimentary parsing and the replacement of keywords, e.g., "is" becomes "=", etc.

These equations, together with a set of prestored equations such as "distance = speed x time",are solved to get the solution to the problem.

The result is that STUDENT has a very limited application.

(i) It can only handle a limited subset of English (e.g.,no dependent clauses),

(ii) It cannot use sentences which do not translate directly into one or more equations (e.g.,"A ship is travelling east"),

(iii) It has difficulty with para-phrase, and

(iv) It can only solve linear equations.

These limitations would prevent STUDENT from solving the Mechanics problems we are considering.

Charniak's program, CARPS, began to correct these defects by using a better parser and an intermediate meaning representation. But both are crude in comparison with later work in this area (cf, Winograd 1972; Schank 1973; Rumelhart and Norman 1973; Simmons 1973; Charniak 1972). Charniak(1969) gives examples of sentences whose grammar had to be altered in order for them to be accepted by the program, and of problems which CARPS could not solve because of its lack of real world knowledge:

"A barge whose deck is 10 ft below the level of a dock is being drawn in by means of a cable attached to the deck and passing through a ring on the dock. When the barge is 24 ft from and approaching the dock at 34 ft/sec, how fast-is the cable being pulled in?"

The ship is moving horizontally towards the dock, but the problem does not mention this, and the program is unable to infer it. Our program could easily be given such information by defining "ship" as a particle in contact with a horizontal plane, and giving the program this definition as a prestored model.

It seems to us that STUDENT, and to a lesser extent, CARPS, show a superficial understanding of the problems they solve. Hence they are limited in application, and easily "fooled". There is no doubt that they have some psychological validity as models of poor students (see Paige and Simon 1973), but they are intellectually unsatisfactory as models of experienced mathematicians.

Research on solving problems, given their mathematical representation, has been progressing for several years and has recorded some notable successes (see Gelernter 1963, Bledsoe 1973, Boyer and Moore 1973). The issue of the effect of different mathematical representations on the problem solving process has recently been discussed. (Amarel 1968, Kowalski 1974 P73). We believe that these discussions cannot be sensibly conducted in isolation from the problem of what mathematical representations it is possible (or feasible) to extract from a natural language representation. Indeed, getting a mathematical (problem-solving) representation is generally acknowledged to be the most difficult part of solving the missionaries and cannibals problem (the problem considered by Amarel). There has also been discussion about the relationship between the meaning representations of natural language and visual information (Minsky 1974, Simmons 1974). We see our proposed work as contributing to this discussion and extending it to mathematical representations.

We proppse to study the above questions in the domain of mechanics problems that deal with idealized objects such as smooth planes, light inextensible strings and frictionless pulleys (see Humphrey (1957) p 1-90). The choice of this domain was governed by the following factors :

(a) We wanted a set of problems which could not be solved using the keyword matching of Bobrow or the simple meaning representations of Charniak.

(b) Because a large amount of knowledge about the objects mentioned in a problem is necessary in order to solve the problem, we wanted a large set of problems about a small universe of discourse. In fact there are only a handful of idealized objects in the set of dynamics problems we are considering (add wedges, discs, containers, rods and particles to the above list). New objects can be handled by considering them as made up from ideal objects (e.g.,a ship as a particle on a horizontal plane). Our program will have a few such prestored definitions, but will not be able to make up new ones or choose between several alternatives.

(c) For our first attempt at natural language/mathematics translation, we wanted a domain which was well understood in that people with plenty of experience of the problems found them straightforward. This is not true, for instance, of "Brainteaser" puzzles, like the missionaries and cannibals.

(d) We wanted a domain in which the natural language input was relatively easy to translate into a meaning representation, and in which the mathematical problem-solving was fairly straightforward, so that we could concentrate on the transition from initial meaning representation to mathematical representation. The language of the dynamics problems is stylized and easily parsed. The mathematical problem-solving consists of equation solving, a domain in which we were already working (see Bundy 1974, Bundy 1975). In fact the equations we have looked at to date are simultaneous linear equations which can be solved by symbolic Gaussian elimination.

In addition to the above advantages, dynamics problems raise the following interesting issues:

(a) The equation forming and equation solving processes can interact in interesting ways. For instance, the order in which the equations are formed can suggest the order in which the variables in them should be eliminated and solved for. Failures of the equation solving process can suggest which equations to form next. Tying the equation-solving process to a particular domain will provide it with (so-called) semantic information to "guide" its solution (see Bundy 1975 section 4.2).

(b) It will be necessary to formalize common-sense physical and geometric notions; for instance: qualitative physical laws like "objects suspended from a string hang vertically down". We will also have to solve "ontological" problems in order to represent relationships between objects such as types of contact and spatial constraints. We will have to deal with the problem of causality (i.e., the frame problem), but in a limited, well-understood situation. We feel confident that solutions to these problems in our limited domain will generalize to more complex domains. The reason for our optimism is that the "ideal" objects we are considering were not chosen at random, but have been developed over many years by engineers and physicists as representatives of real world objects.

(c) Both the mechanics and equation-solving studies are aimed at formalising areas of mathematical activity which are not yet well understood, such as searching and model-making. We hope that techniques developed and answers discovered in one area will be applicable to the other. In particular, we expect to use the same protocol-analysis philosophy, equipment and personnel in researching each domain.

Our objective then, is to write a program which can solve Mechanics problems, of the kind given as examples in Humphrey (1957) p 1-90. We will concentrate on writing a program which can extract equations, given a surface level meaning representation obtained from the statement of the problem in English. If we make suitable progress on this, we would like to extend the program so that it would accept problem statements in English.

2. Methodology

The Descriptive Theory. Since the area we are proposing to study has not been formalized, our first task must be to build up a descriptive theory of it, i.e., we must discover what kind of reasoning goes on.

We plan to build this descriptive theory by analysing solutions to dynamics problems using any help we can get from mathematical textbook, introspection and our knowledge of physics, applied mathematics and geometry. We will make selective use of protocol analysis to attack difficult issues. As with most non-trivial AI research, solving mechanics problems is too big a task to tackle without dividing it up. However, it will not segment into non-interacting parts. Our solution to this dilemma is the normal one of dividing it into parts initially in order to build a descriptive theory, but studying the interactions and allowing a more flexible organisation within the program itself. In our descriptive theory we envisage the problem representation going through seven stages :

(1) The original natural language input.

(2) The parse tree of (1).

(3) A surface level meaning representation (verb and noun phrase definitions are still 'packed').

(4) A deep level meaning representation (mainly spatial information, including notions of contact, attachment region, etc. This is vital for later calculation of internal forces).

(5) A deep representation augmented with information about the physical quantities needed to calculate the accelerations, forces, energies,etc.

(6) The equations (i.e.,the mathematical representation).

(7) The answer.

Each stage will be obtained from previous ones by the application of laws (of English, Geometry, Physics, Mathematics, etc.) and by the making and confirming of plausible hypotheses. We would like to emphasise again that this division into stages is a first approximation to enable us to understand the kind of reasoning which takes place. We do not expect that it will be possible to solve problems by deriving these stages in linear sequence.

We intend to focus our attention on the transition from stages (3) to (6), because this is the area which is least well understood. The transition from (1) to (3) and/or (4) has been modelled by many natural language understanding programs (e.g.,Winograd 1972). So, initially, we will consult our colleagues working in natural language understanding as to the form of surface level representation it would be reasonable to assume, and input this. However, if the project makes satisfactory progress, we would like to extend the program to accept input in English. We choose to input the surface level representation, rather than the deep structure representation, because there is no concensus on an appropriate formalisation of the latter and because the transition between them is a vital part of the problem solving process. Since, initially anyway, we will not be doing any natural language processing per se, our project should be regarded as a study in the representation of knowledge rather than as a study in natural language understanding.

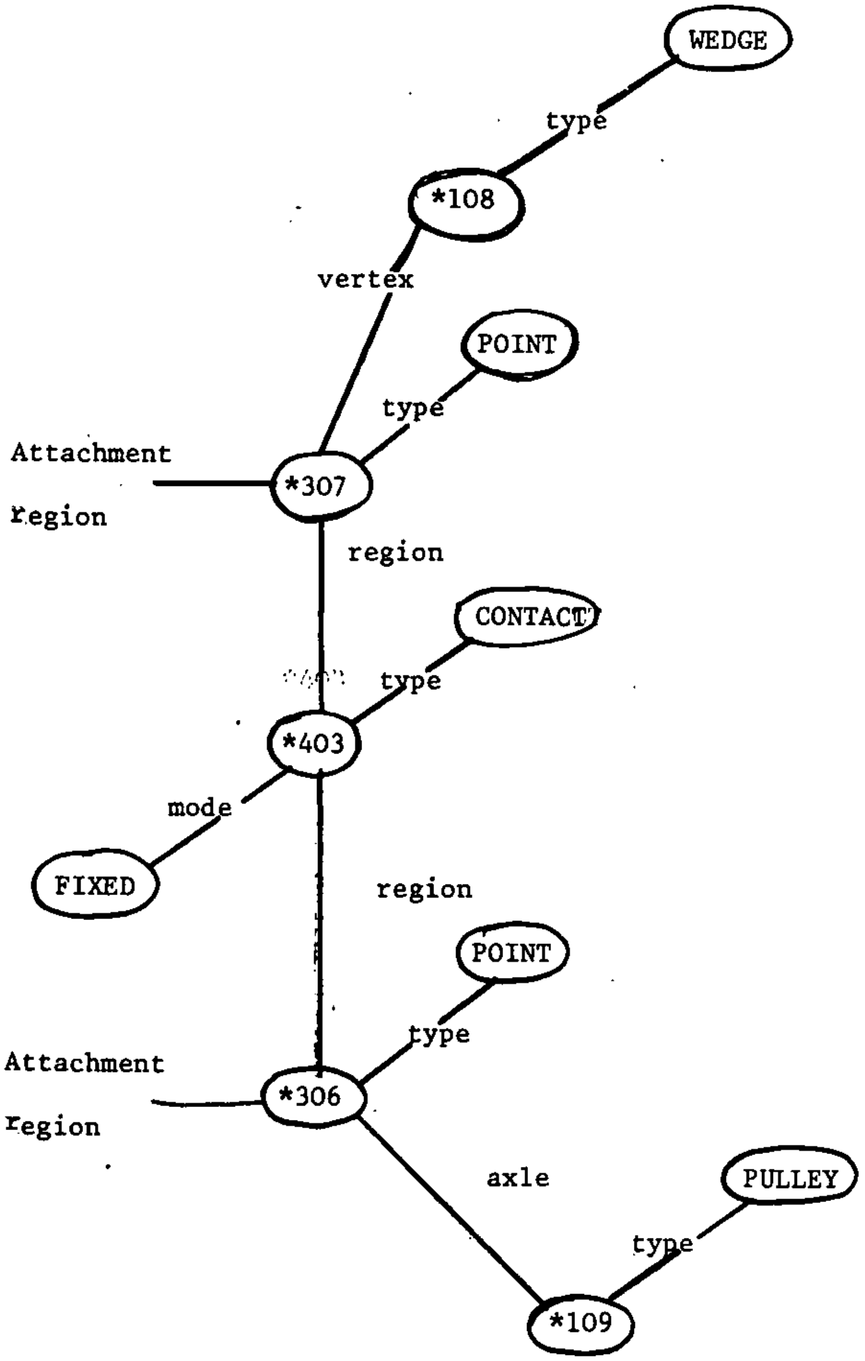
The transition from the equations to their solutions (7) is the business of the equation solver, being worked on in parallel by Bundy and Welham (see Bundy 1975). We expect this part to be relatively self-contained for simple problems, but expect some interesting interactions in more complex problems.

So far, we have only considered natural language input. We also hope that it will be possible to modify our program to use the visual input of a diagram. As a first approximation for doing this we would take the advice of our colleagues in visual perception as to the form of the symbolic description of the visual scene it would be reasonable to expect. In this case, instead of the deep level representations (4) being formed from representation (3) alone, it would be formed from (3) and a meaning representation of the visual scene. This would involve correlating objects named in the written input with objects seen in the visual input: there is obviously great scope for disambiguating one lot of data with the help of the other. However, we do not propose research in this area as an immediate goal.

We are building our descriptive theory from the outside in, i.e., given the equations (6) for a particular problem, we are developing the representation of the physical system (5). Given the initial surface level representation (3), we are developing the deep level representation (4). Finally, we are trying to bridge the gap between (4) and (5). The intermediate goal of wanting to visually display representations (4) and (5) has been of some help by suggesting spatial and physical information that is vital for forming equations.

We divide the task of designing these representations into two parts: discovering the ontology of each representation and discovering the laws of reasoning used to build them.

(i) Ontology We have to decide what type of entities we are going to reason about. For instance, in the deep structure representations (4) and (5), we have to classify the various kinds of contacts that can occur between objects. Our current classification is that contacts can be fixed or movable, and if movable they can be slipping or non-slipping. Touching objects are not in contact everywhere, so we have to introduce the notion of attachment regions which are the parts of the objects which are in contact. We illustrate this with a network representing the deep structure of the phrase "The pulley is attached to the vertex of the wedge".



Note that the information that the pulley is attached at the axle is not given in the original natural language statement, but is a reasonable assumption to make. This is an example of the kind of plausible inference our program will have to make as it develops a representation.

We use semantic nets to represent such information because we find them more readable than, say, Predicate Calculus assertions of PLANNER theorems. However we have no particular commitment to semantic nets as a programming technique. The networks we use, are not ad hoc. They have a clear (predicate calculus) semantics, i.e., nodes labels like "*108" or "CONTACT" are constants; link labels like "type" or "attachment region" are predicates. Starred numbers like *108 are objects; other node labels are descriptions of objects. Of course each description should occur only once in the diagram, but for readability we have repeated the node for "POINT". Each node has a "type", and two nodes of the same type have the same kind of links attached to them. Some of these links are compulsory, e.g., "type", some are optional, e.g. "vertex". Precisely what these links are for each type of node will change as we build our descriptive theory. So the networks are still tentative.

(ii) Laws We will also have to represent the way these entities are related. Some of this knowledge will be represented implicitly. The fact that the axle is an attachment region of the pulley will probably be represented in a prestored model of the "ideal" pulley. The fact that there is only friction between objects in slipping and movable contacts could be represented by only allowing a friction link to be drawn in this case. Some of the knowledge will be represented explicitly in the form of laws.

That objects in fixed contact have the same acceleration (also the same location and velocity) could be represented as below.

Type (x, contact) & Fixed (x) & Region (x, y1) & Region (x, y2) & acceleration (y1, z) → acceleration (y2, z).

The various representations (3), (4), (5) and (6) (p.6 above) will be stored in an assertional database. Such databases are provided by programming languages like CONNIVER (Sussman 1973), POPCORN (Hardy 1973) and PROLOG (Warren 1974). Each offers a different indexing scheme for the database and different methods of controlling search. Although, we have not yet committed ourselves to a programming language, our eventual choice will be determined by:

(a) The indexing scheme and search control needed.

(b) The range of languages available.

(c) The ease with which our descriptive theory can be expressed in the language.

(d) The amount of storage occupied by the program and the data structures it creates.

(e) The time our program takes to run.

(f) Compatibility with other programs such as the equation solver.

Having all the representations stored in a single database will allow the program the flexibility it needs in its development through the various stages. It will be possible to start one stage of the representation before completing previous ones and it will be possible to call the laws in a top-down (backchaining) manner. This is necessary since, for instance, it is sometimes not possible to decide the precise physical configuration before we have explored some of the forces that are acting on the system. We do not want to augment the representation with information about all the forces, velocities, energies, etc. that are acting, but only that needed to answer the questions posed. Thus, the various representations will be allowed to interact with each other in the way that our observations of this problem domain and the experience of workers in other domains indicate will be necessary. These issues of controlling inference will be considered in depth after completion of the first stage of building a descriptive theory.

3. The Partial Analysis of a Simple Mechanics Example

In our work so far we have concentrated mainly on problems with fairly simple solutions. This has resulted in the emphasis being placed on the preliminary analysis of the problem, the most poorly defined area in mechanics problem solving. Consider the example (taken from Humphrey, 1957):

"Two particles, of masses 5 and 7lb., are connected by a light string passing over a smooth pulley. Find their common acceleration and the tension in the string".

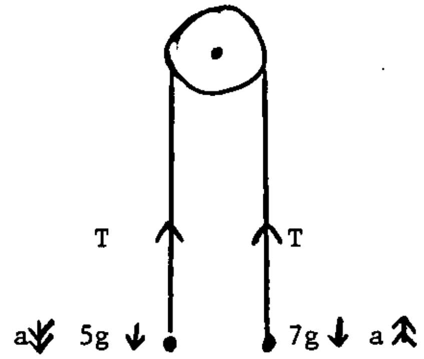


FIGURE 1

where "T" is "the tension in the string", "a" is the "common acceleration" and "g" is the "acceleration due to gravity"

As seen in this example, the problem statements use a surprisingly simple and stereotyped grammatical form. There are several existing grammars that would be more than adequate to parse them, e.g., Simmons (1973). It seems reasonable to assume that the progression from natural language to (at least) a phrase structure parse tree (our second stage) would be fairly standard. Either during this progression or at some later stage, simple semantic checking can take place. Most of the phrases in the problem refer to objects, e.g. 'particles', or simple r systems of objects, e.g. "a light string passing over a smooth pulley". The translations of these noun phrases are checked against the semantic

constraints on the verbs which connect them.

constraints on the verbs which connect them. If these checks are successful the translations are incorporated into the surface level network. Assigning modifiers to the appropriate nouns is not difficult since particles are objects and hence have masses, and thus "masses 5 and 7 lb." confirms the syntactic relationship implied by the preposition "of". Figure 2 (representing our third stage) illustrates the detail of information we have at this point. This could be the output from a natural language processor, and would be the input for our program.

Note that two unknowns have been created to represent "their common acceleration" and "the tension in the string". The presence of these will guide the future development of the representation, especially at stages (5) and (6).

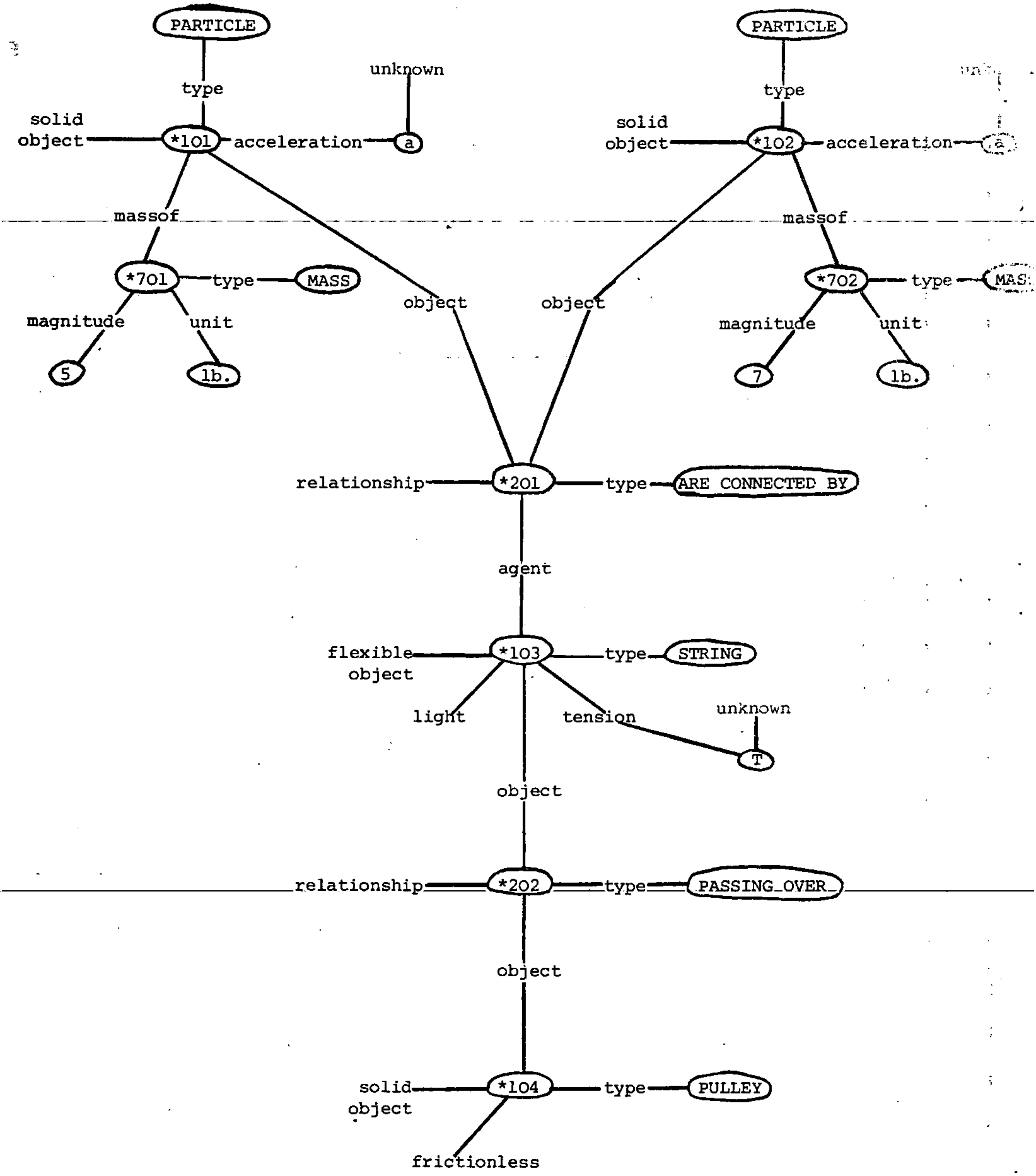
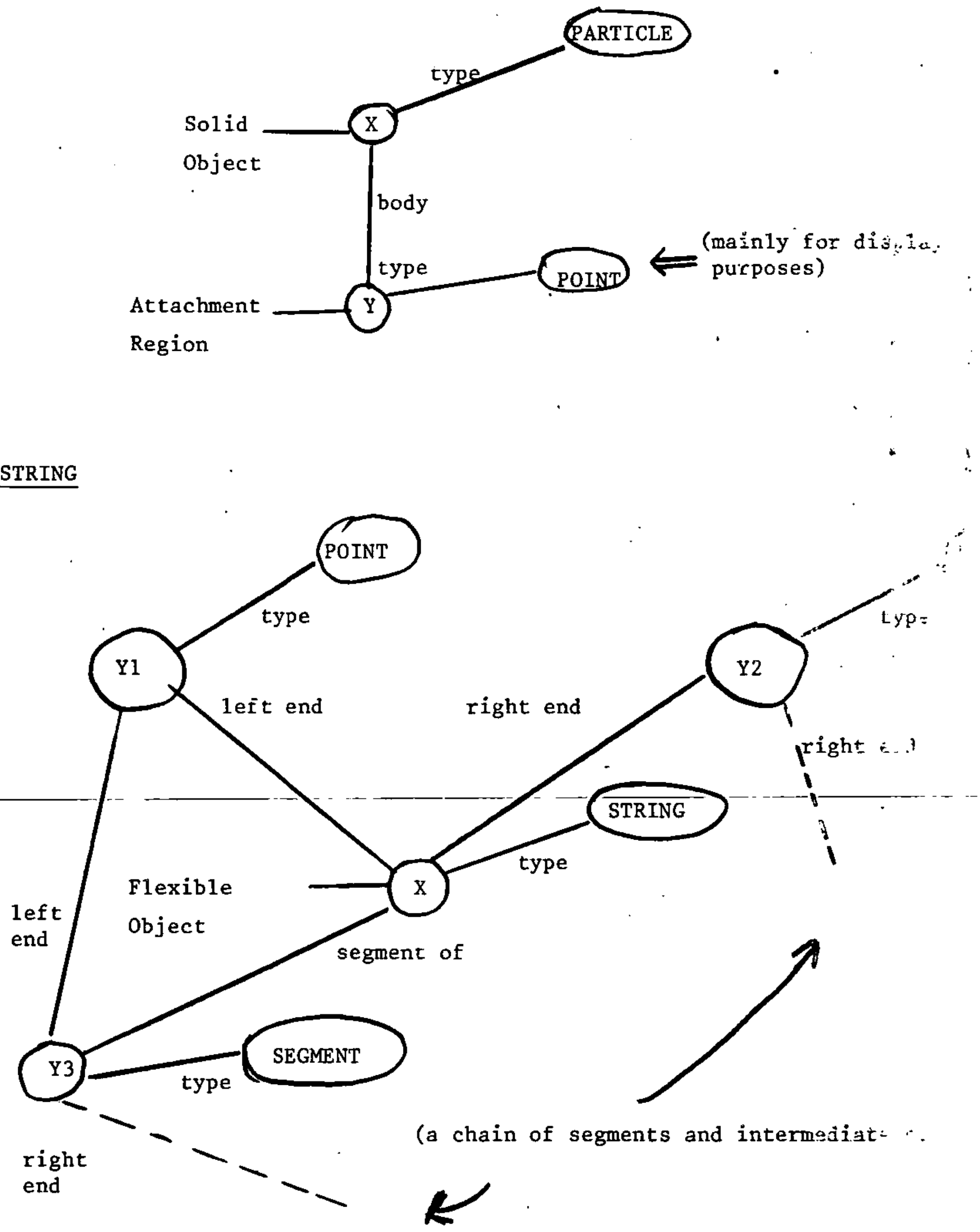


FIGURE 2 Semantic network of natural language statement.

The fourth stage is the deep structure representation: In order to examine the semantic relationships more closely, and so proceed to the deep structure representation, it is necessary to bring in a detailed definition of each verb. These definitions are supplied by our descriptive theory and are in terms of primitives like "contact" and "attachment region". At present our definition of CONNECT is divided into two parts, CONNECTED TO and CONNECTED BY. The appropriate definition in this case is CONNECTED BY, which is defined as "a fixed contact between one attachment region of the agent and an object, and another fixed contact between another attachment region of the same agent and another object". Prestored models of the objects are used in conjunction with these definitions to see if there is indeed an available attachment region or regions on the object in question. FIGURE 3 shows some examples of such models.

FIGURE 3 Models of Ideal Objects



a) PARTICLE

b) STRING

The models are matched against the verb definitions in order to build up a representation of the physical configuration of the objects in the problem. The process of building up this representation may involve further inference. For instance, the support relationship must be inferred:

(i) The pulley "supports" the string because the string is in contact with the top portion of the pulley.

(ii) The string "supports" the particles because they are in contact with the string and nothing else.

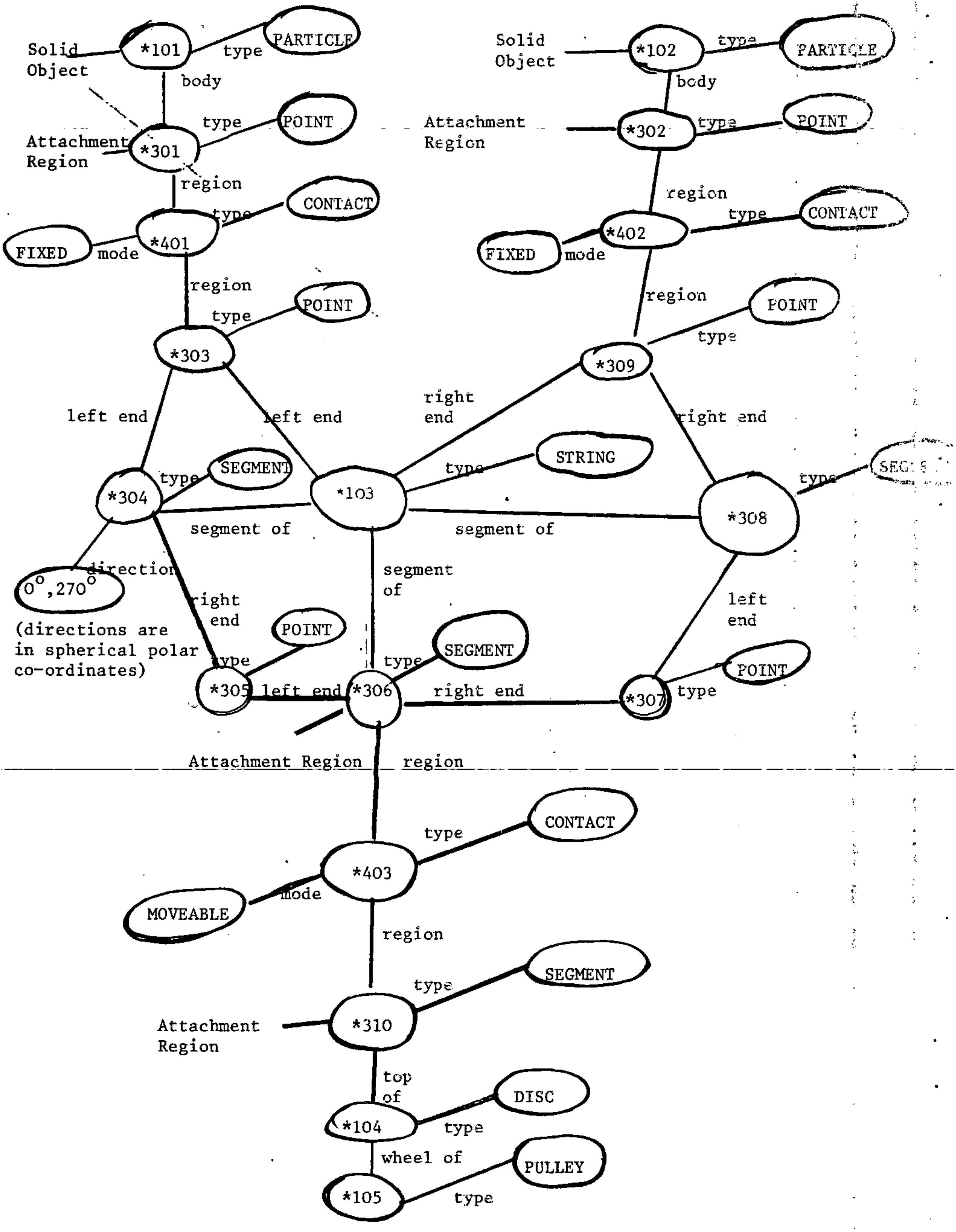
(iii) At least one object in every system is supported from outside the system. The only candidate for this in the problem above is the pulley.

(iv) The parts of strings which support objects, not supported by anything else, hang vertically down (270°) unless we have information to the contrary (i.e. unless we are told the system is a pendulum or that the object is displaced, we assume the <u>default case</u>, that it is not displaced c.f. Minsky 1974).

This process leads us to the representation shown in FIGURE 4.

FIGURE 4 Deep Structure of Problem Statement

(some inessential detail missing)



This representation provides most of the information we need to display a diagram on a graphics display. We plan to do this in order to get a readable description of the current state of the database for debugging purposes. The information necessary for a diagram, e.g., directions of strings, relationships between objects, etc., is also necessary for our next step of calculating the forces and accelerations.

The fifth stage is the deep structure representation, augmented with information about the physical quantities needed to calculate the accelerations of the particles and the tension of the string.

There are two different types of force that can be exerted, external and internal. The external forces are gravity, which acts on any object with a mass, and those applied by an external agency. The magnitude of the gravitational force is the mass times "g", while the direction is vertically down, (270°). In our problem the only relevant external forces are the effect of gravity on the particles and the notional force which supports the fixed pulley.

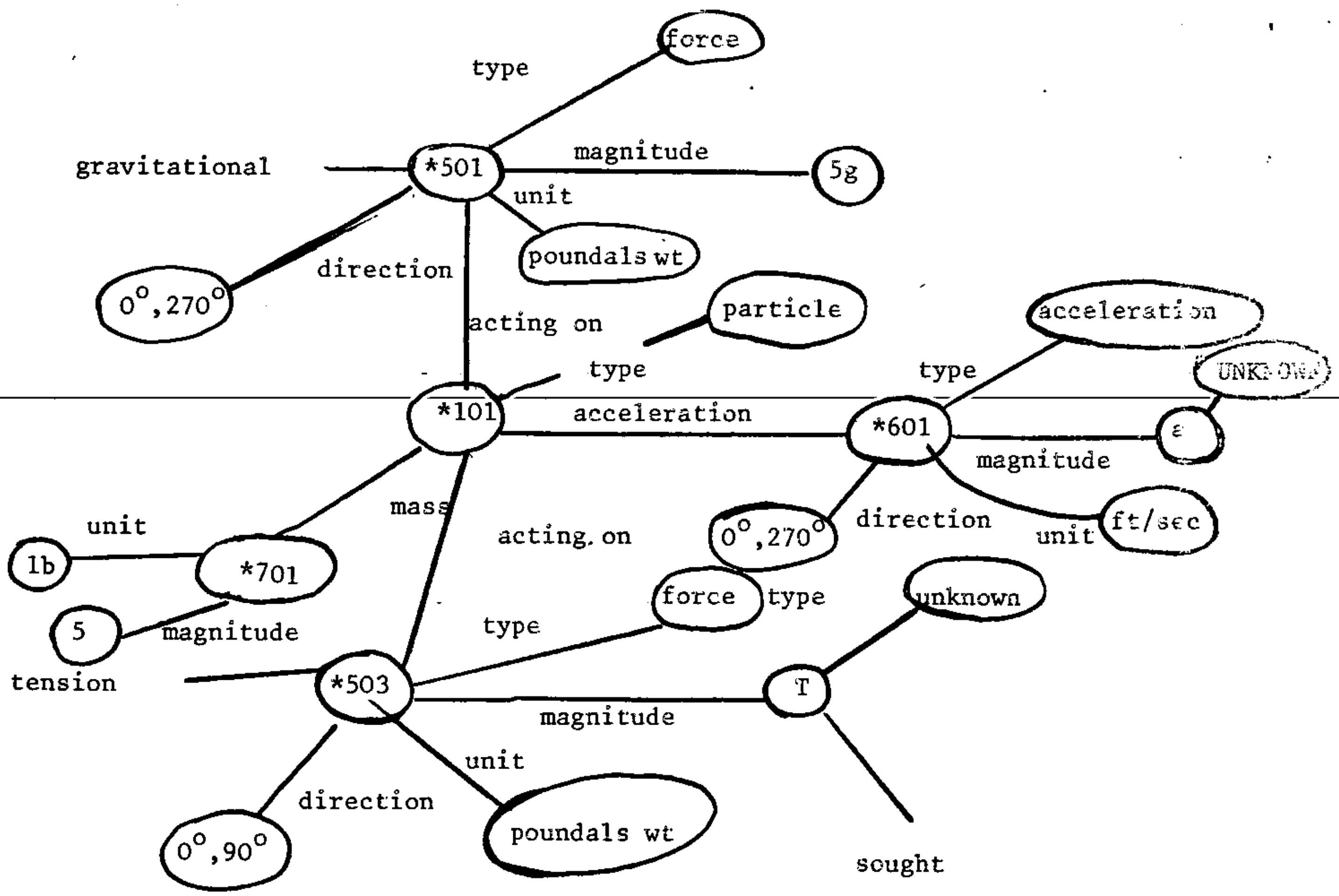
The internal forces are caused by pressure between objects in contact. The particles suspended by the string cause the tension, T, in the string. In fact the wording of the question suggests that the magnitude of the tension is uniform throughout the string. We would normally expect to deduce this. In this case we could adopt it as a plausible hypothesis and then confirm it. The law which enables us to do this in either case is that the tensions in two adjacent segments of the string have the same magnitude provided no force is applied in the direction of the string at their intermediate point. In our case the only candidate would be friction at the contact with the pulley, but this is zero since the pulley is smooth. So we could postulate a tension in the leftmost segment of the string, applying a force T vertically upwards on the left particle.

This tension would be inherited, unaltered in magnitude, by successive segments of the string, until it exerts a force T vertically upwards on the other particle.

In a similar way the wording of the questions suggests that the magnitudes of the accelerations of the particles are identical. We can confirm this by assigning an acceleration, a, vertically upwards on the leftmost particle and having it inherited, unaltered in magnitude, by successive segments of the string, until it is assigned to the other particle.

FIGURE 5 indicates the additions that this process would make to the network.

FIGURE 5 Forces acting on, and acceleration of, a typical object



The sixth and seventh stages are the generation and solution of the equations.

Our overall strategy is to get two simultaneous equations relating a and T, and to solve them. So we need equations connecting acceleration and force. This suggests Newton's equations and since all the forces and accelerations are acting vertically, it is obvious that we should resolve the forces in this direction. This might consist of the procedure:

```
Set sum to nil
For each force acting on the object (see Figure 5)
   Multiply the magnitude of the force by the cosine of
   the angle between its direction and the direction of
   resolution (trivial in our case). Add this quantity
   to sum.
Put sum equal to the mass of the object times the magnitude of
its acceleration in the direction of resolution.
Simplify the resulting equation.
```

Carrying out this process on both particles yields the equations:

$$5.g - T = 5.a \qquad (1)$$

$$T - 7.g = 7.a \qquad (2)$$

which can be solved to give values of 5 5/6ths poundals for T and $-1/6$. g ft/sec^2 for a.

4. Wider Justifications

Although there are many areas in which our research could be fruitfully applied we will expand only three: the teaching of mechanics, the study of education in general, and the provision of an interactive aid for engineers.

In order to write a computer program to solve mechanics problems it will be necessary to spell out carefully the meanings of many intuitive physical and geometrical notions such as those implicit in the statements "a body is accelerating in space" or "the velocity of a wheel is a constant". We believe, and human protocols seem to confirm, that misconceptions of notions such as these are a major source of mistakes for humans solving problems in mechanics. The human might think, for example, that a body has a single acceleration : that is that all points of a rotating accelerating body will necessarily accelerate at the same rate. This "bug" in a problem solving system - whether human or mechanical - could produce interesting, even if contradictory, results. Hopefully, our program, if given the same misconceptions as some human subjects, will be flexible enough to effect the same incorrect results, and to serve as a vehicle for investigating similar misconceptions in students. The need for correct intuitions about basic physical concepts and the way these interact to produce a solution are the main concerns of our research. These are also important topics in the teaching of mechanics and show why some of our strongest encouragement and offers of assistance have come from such places as the Department of Engineering at Cambridge (Marples 1974, and Simpson, 1975).

There are several educational issues, especially in the area of teaching mathematics, that our research will address. Among these are the study of the possible decompositions of a problem into subproblems and the interweaving of these subproblems into an eventual solution of the problem. Polya (1962) often discusses this in terms of general problem-solving "strategies" . While not taking on the study of problem solving strategies in general, we will give concrete conditions under which a specific strategy is useful. Another aspect of education we address is the study of "transfer". In creating a system capable of solving large classes of problems we will need to outline the general structural features of problems that remain invariant across problem-solving situations; for if our solver is not to succumb to the combinatorial explosion the strategies used must be powerful enough for there to be "transfer" within the problem-solving domain. The attempt to make explicit exactly what a problem-solver "takes" from one problem solving situation to the next, as well as being a crucial question for us, has long been a concern of educational researchers.

The third application of our work could be as an automatic problem-solver for Engineers. Ideally the Engineer would type his problem into a computer in English and have the answer typed back. For example, to find the stress bearing on some point in a bridge he would type in the specifications for the materials and forces involved. While the program we envisage writing would not be capable of doing this, it would indicate the directions for writing a program which could. Some of the improvements to our program necessary for such a system are :

(i) The program would have to deal with far more difficult problems involving more sophisticated engineering knowledge (e.g. vibration), more complexity, and the analysis of real world objects (e.g., bridges).

(ii) The program would have to be more flexible in its interactions with the user e.g., accept diagrams and ungrammatical sentences, and perhaps be able to engage in a dialogue with the user.

(iii) The time/space efficiency of the program would be a major factor and require further study.

Acknowledgements

We would especially like to thank our fellow workers at D.A.I. who helped us think through and formulate this research. (Bernard Meltzer, Gordon Plotkin, Tim O'Shea, Mark Eisenstadt). Also our thanks to the "outside readers" whose criticisms helped us "tighten up" the presentation of our own work (Pat Hayes, Aaron Sloman). This paper is a revised version of a Grant Application to the S.R.C.

References

1. Amarel, S. 1968

   "On representations of problems of reasoning about actions"

   pp 131-172, Machine Intelligence 3, ed Michie D, Edinburgh University Press, 1968.

2. Bledsoe, W. W. 1973

   "A man-machine theorem-proving system"

   pp 56-66 Proceedings of IJCAI3, ed Nilsson N, Stanford, California, 1973.

3. Bobrow, D.G. 1974

   "Natural language input for a computer problem solving system"

   Report MAC-TR-1, Project MAC, MIT, Cambridge, Mass., 1974.

4. Boyer, R. and More J.S. 1973

   "Proving theorems about LISP functions"

   pp 486-493, Proceedings of IJCAI3, ed Nilsson N., Stanford, California, 1973. (Also DCL Memo No. 60).

5. Bundy, A. 1974

   "A treatise on elementary equation solving"

   Internal Memo, Department of Artificial Intelligence, Edinburgh, April, 1974.

6. Bundy, A. 1975

   "Analysing mathematical proofs (or reading between the lines)"

   DAI Research Report No. 2, Department of Artificial Intelligence, Edinburgh, March, 1975 (Shortened version submitted to IJCAI4).

7. Charniak, E. 1968

   "CARPS, a program which solves calculus word problems"

   Report MAC-TR-51, Project MAC, MIT, Cambridge, Mass., 1968.

8. Charniak, E. 1969

   "Computer solution of calculus word problems"

   pp 303-316, Proceedings of IJCAI2, eds Walker,D.E. and Norton L.M.1969

9. Charniak, E. 1972 .

"Toward a model of children's story comprehension"

Report AI-TR-266, AI Laboratories, MIT, Cambridge, Mass., 1972

10. Gelernter, H. 1963

"Realization of a geometry theorem-proving machine"

pp 134-152, Computers and Thought, eds Feigenbaum E.A.

and Feldman J., McGraw-Hill, New York, 1963.

11. Hardy, S. 1973

"The POPCORN Reference Manual"

SCM-1 Essex University, 1973.

12. Humphrey, D. 1957

" Intermediate Mechanics (Dynamics)"

Longmans, Green and Co., London, 1957.

13. Kowalski, R. 1974

"Logic for problem-solving"

DCL Memo No. 75, Department of Artificial Intelligence,

Edinburgh, 1974.

14. Marples, D.L. 1974

"Argument and Technique in the Solution of Problems in

Mechanics and Electricity"

CUED/C - Educ/TR1, Dept. of Engineering Memo, University

of Cambridge, 1974.

15. Minsky, M. 1974

"A framework for representing knowledge"

MIT AI, Memo No. 306, 1974.

16. Paige, J.M. and Simon, H.A. 1966

"Cognitive processes in solving algebra word problems"

Problem Solving, ed Kleininuntz B., Wiley, New York, 1966.

17. Polya,G. 1962

"Mathematical Discovery" Vol. I and II,

New York : Wiley, 1969.

18. Rumelhart, D.E. and Norman, D.A. 1973
"Active semantic networks as a model of human memory"
pp 450-457, Proceedings of IJCAI3, ed Nilsson N., Stanford, California, 1973.

19. Schank, R.C., Goldman, N, Reiger, C.J. and Riesbeck, C. 1973
"MARGIE: Memory, Analysis, Response Generation, and Inference on English" pp 255-261, Proceedings of IJCAI3, ed Nilsson N, Stanford, California, 1973.

20. Simmons, R. 1973
"Semantic networks: their computation and use for understanding English Sentences"
Computer Models of Thought and Language, eds Schank and Colby, Freeman, 1973.

21. Simmons, R. 1974
Grant Proposal to NSF, 1974.

22. Simpson, P. 1975
Untitled Manuscript describing work of the Educational Research Unit in the Dept. of Engineering, University of Cambridge, 1975.

23. Sussman, J.F. and McDermott, D.V. 1972
"The Conniver Reference Manual"
MIT AI Memo No. 259, 1972.

24. Warren, E. 1974
"Epilog (400,400) - A Users Guide to the DEC 10 PROLOG System"
Internal Memo, Department of Aftificial Intelligence, Edinburgh, 1974.

25. Winograd, T. 1972
"Understanding Natural Language"
Edinburgh University Press, 1972.