Are Argument Representation Schemes Useful? *

D. Charles Hair
Clayton Lewis

CU-CS-475-90

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

* The work described in this article was supported by NSF grant number IRI-8617383.

Are Argument Representation Schemes Useful?

D. Charles Hair and Clayton Lewis

CU-CS-475-90 June 1990

Department of Computer Science
Campus Box 430
University of Colorado at Boulder
Boulder, Colorado 80309-430
Telephone: 303-492-1592
FAX: 303-492-2844

The work described in this article was supported by NSF grant number IRI-8617383.

Are Argument Representation Schemes Useful?

D. Charles Hair and Clayton Lewis
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309
Telephone: 303-492-1592
FAX: 303-492-2844

INTRODUCTION.

As part of the EUCLID research project (Smolensky et al 88), we have looked into a variety of possibilities for representing arguments. One of the general goals of the EUCLID work is to develop a representation scheme that will enhance people's ability to analyze arguments. It has been suggested that schematic representation methods can be used to capture that goal. The desired benefits are in the following areas: (1) a person constructing such a representation for an existing argument may achieve a deeper understanding of the argument, (2) a person using an existing schematic representation may achieve a deeper level of understanding of the argument than they would with a standard prose representation, and (3) a person constructing a new argument may create stronger arguments as a result of using a schematic representation.

While our results have bearing on all three of these areas, we focussed specifically on the second one. In exploring this idea, we used the point of view of supposing that a good schematic representation of an argument would enable someone looking at the argument to pick out weaknesses in the argument more easily than could be done with a typical prose representation. We proceeded to apply three specific representation methods to two different arguments that we selected from articles in scientific journals. The articles were chosen on the basis that both presented somewhat involved arguments for given positions, both of the positions argued for left some open questions, the articles were from different academic areas, and there was a body of followup developments in the relevant literature for both articles.

The rationale for pursuing this approach rests in part on the fact that a representation method that does enhance people's analytic abilities would be of obvious practical use in fields such as academia, debate, and law. Also, the finding that a given representation helps in a cognitive act like argument analysis would have implications for psychological questions about how people perform such acts, and might well lead to more generalizable conclusions about ways to construct user interfaces to computer systems.

ARGUMENTATION LITERATURE.

There is a well developed academic interest in the area of argumentation, as evidenced by the writings of scholars from a variety of backgrounds (Anderson and Dovre 68) (Barth and Martens 82) (Cox and Willard 82) (Fogelin 87) (Golden and Pilotta 86) (vanEemeren et al 84). Much of the interest stems from those who are primarily interested in the study of rhetoric, and who trace their interest back to Aristotle's analysis of rhetoric. There are also people whose main academic interest is in philosophy, communications, law, and speech.

Two of the most influential modern scholars in this area have been Toulmin (Toulmin 58) and Perelman (Perelman and Olbrechts-Tyteca 69). Although they take different approaches in their respective analyses of argument, both are known for having mounted an attack on what can be viewed as the dominance of deductive logic as the paramount model for good argument. It should be noted that although both of them have been widely cited, both have been subjected to as much criticism as praise. The general consensus seems to be that while it is desirable to understand the workings of nondeductive arguments, Toulmin and Perelman both went too far in the direction of attacking the role of deductive logic. Toulmin is particularly noted for having set forth a graphical model of arguments, which is one of the three schemes we have explored.

There have been three basic approaches to modelling arguments. The most common approach has been to base a given model on some form of formal deductive logic. Another approach has been to propose a variety of different kinds of arguments, each of which has its own peculiar characteristics. This approach is used by Perelman. Toulmin takes the third approach, which is to propose a graphic representation scheme which he asserts can be used to represent all argument types.

In Toulmin's model an argument is graphically represented through use of a schematic in which there are three necessary parts: (1) data, which support (2) a claim, where the support is justified by (3) a warrant. In addition to those basic parts, there can be qualifiers of the claim like "probably", there can be backings which further justify warrants, and there can be rebuttals which act to restrict the applicability of the claim. Each individual argument scheme can be linked to other arguments as a datum, warrant or the like for that argument.

Toulmin's model has been much discussed by people in the argumentation field (Brockriede and Ehninger 60) (Burleson 79) (Cooley 59) (Manicas 66) (McCroskey 65). The discussions have ranged from those who have tried to demonstrate the value of the model by showing how it might be used with real arguments, to those who are critical of the model's ability to be of any real general use. The critics point to ambiguities in trying to discriminate for real arguments between the data and the warrants, as well as to the fact that Toulmin himself seems to have viewed the model as only a tentative proposal that he did not return to in his later work(Cowan 64). However, it can be noted that Toulmin did use the model at least once more in (Toulmin et al 1979). It has also been suggested that this scheme can be of only limited value for analysis because the model does not lead to the production of unique representations for given arguments (Cooley 59). The Toulmin structure is currently being used in research work by (Marshall 87,89).

One of Toulmin's critics suggests that a problem with Toulmin's approach is that it has too many argument parts (Cowan 64). Cowan suggests that the only essential parts of an argument are premises and conclusions.

There seems to have been little work done in the area of obtaining empirical evidence as to what kinds of argument models might help or hinder people's understanding of arguments. The work we have discovered, has suggested that there is little observable difference in people's ability to recall an argument based on changing sentence order (Bettinghaus 86). Although one can find occasional intuitively based claims to the effect that graphical argument representations would aid comprehension (Kneupper 78) (Brockriede and Ehninger 60), we have found no actual references to experimental data to support that claim.

There has been one report, however, that has suggested the generation of insights into arguments directly related to the use of a graphical, hypertext based representation system. (VanLehn 85) indicates that in the course of using NoteCards to represent a lengthy argument he had two incidents in which he believes the use of the tool led to significant new insights. In one case, he was led to observe a major flaw in his argument analysis. In the other case he was led to develop a new kind of representation of the knowledge in argument structures.

As a related matter, (Hample 82) discussed his view that there should be increased efforts to develop mathematical models of argumentation. He distinguishes what he means by "model" from "diagrams." In his analysis, diagrams are the kind of schematic representations that we have investigated and that we sometimes refer to as "models." He suggests that diagrams have their uses, but do not get at the nature of arguments in the deeper way he envisions for models.

METHOD.

The two principal arguments we have used have been about the verifiability of computer programs (Fetzer 88), and about the density of tree species in tropical forests (Janzen 70). Author Lewis worked with the Janzen article, while author Hair worked with the Fetzer article.

One of the representation schemes is based on a hypertextlike model of arguments being developed under the EUCLID project (Smolensky et al 88). This representation method relies on breaking an argument up into a series of claims. The claims are related to each other by "support" or "refute" links.

The second scheme uses the Toulmin structures referred to above. We only made use of the three basic parts of that structure, namely the data, claim, and warrant.

Our third scheme consisted of representing the arguments as Prolog programs. Unlike the other schemes, this one requires a full formalization of the argument, and in principle permits some checks on the correctness of an argument to be made automatically. Also, this scheme is the only one that is not graphical.

In developing the various representations, the two investigators used a different order. Lewis began with the Prolog model, then went to the Toulmin model, and concluded with the EUCLID model. Hair began with the EUCLID model, then worked on the Toulmin model, and concluded with the Prolog model. In both cases, the investigators relied heavily on the first representation they developed to guide work on the following representations.

While doing the original work both investigators made efforts to use the EUCLID prototype that had been developed on a Symbolics machine. That prototype was version 2 of EUCLID. However, both ended up abandoning that effort and using other means to simulate the EUCLID output. Lewis' simulation was not patterned strictly after the way EUCLID version 2 represented arguments. Hair's simulation was quite close to the actual EUCLID version 2 layout. A subsequent version of EUCLID proved to be more usable. That prototype is version 4 of EUCLID. A representation of the Fetzer argument was successfully created using EUCLID version 4.

THE JANZEN ARGUMENT.

Summary of Article.

Janzen notes that in tropical forests individuals of any given tree species are widely scattered rather than clumped together. He proposes that that diversity of tree species in tropical forests is due to the action of species-specific predators in killing seeds and juvenile trees near adult trees of the same species. Juveniles can escape predation in two basic ways. They will escape if their predators are few at appropriate times, and Janzen hypothesizes that is what happens in temperate forests. They can also escape predation if their species invades an area in which its predators are not present.

Prolog Representation.

The Prolog statements for the Janzen article are shown in Figure 1. These statements permit the conclusion to be drawn that there is more likelihood of low density of adult trees in tropical than in temperate forests. That is, the statements are logically sufficient to explain the pattern of distribution Janzen seeks to explain. In the course of preparing this representation the following observations were made.

The order of presentation makes the argument difficult to follow. One difficulty is that Janzen presents analyses of various cases of predation and seed dispersal, but these analyses are not linked to the main argument. Another problem is that the hypothesized difference between tropical and temperate forests, which is crucial to the argument, appears only near the end of the paper. Lewis assumed through more than one reading that the crucial difference between these habitats was the prevalence of host-specific predation. The discipline of composing a logically adequate representation of the argument forced the discovery of the structure without regard to order of presentation.

Janzen presents in diagram form part of the argument in which quantitative relationships among seed dispersal, predation, and survival are developed. The reader became suspicious that certain relationships which were supposed to be multiplicative might actually be shown as additive in the diagrams, an easy mistake to make in dealing with graphs, and that consequently some of the relationships inferred might be wrong. Figures 2 and 3 show two representations of the relationship between seed numbers and distance from the parent. In Figure 2 the lower curves are derived from the upper ones by subtracting a fixed number at each distance, while in Figure 3 the lower curves represent constant fractions of the upper curve. Figure 2 resembles Janzen's hand-drawn curves, but the multiplicative relation in Figure 3 seems the appropriate one: reducing the seed crop should cause approximately a proportional reduction at each distance from the parent. The Prolog representation did not help clarify this matter; instead it simply provided a framework in which to state the (putative) conclusions from the diagrams. On the other hand, though Janzen does draw some apparently erroneous conclusions from these curves they do not figure in the main line of the argument.

A key step in the argument links low survival of juveniles near adults to wide spacing of adults. This conclusion seems doubtful, since it could happen that a number of juveniles could mature as a cluster far from any adult and then become a cluster of adults. The Prolog representation forces a clear statement of this step in the argument, but does nothing to flag it as questionable or to suggest (as done here) how it could fail to be sound.

Another weak point in the argument occurs in developing the assertion that juvenile trees in temperate forests can escape predation because of fluctuations in the numbers of predators due to seasonal or climatic variation. It is necessary to assume that trees can reproduce at times of few predators, which Janzen notes as a premise but does not mark as questionable. The form of the required Prolog representation highlights the strength of the required assumption. Thus, the existential assertion of times with few predators naturally involves a constant denoting such a time, which must then figure in an assertion about possible times for reproduction.

The argument here seems weak on other grounds as well, since presumably it is not sufficient simply for a tree to reproduce at a low-predator time. The juveniles must mature enough to be invulnerable to the predators when they return. Janzen does not develop this issue. The Prolog representation did not assist in noticing this point in any direct way.

The Prolog representation was inconvenient or clumsy in a number of respects. It appeared necessary to represent a number of general rules of inference to connect otherwise isolated assertions. An example is the rule: if X produces Y and Y produces Z, then X produces Z. However, many of the general rules of inference are clearly unsound in general. For example, a rule was formulated that if factor A produces factor B, and if habitat H1 has more of factor A than habitat H2, then H1 has more of B than H2. Each rule seemed reasonable in some cases, including those in which it would be invoked in this argument, but not unconditionally correct. This problem poses a dilemma for the argument critic examining the Prolog representation. Throwing out any rule that is not unexceptionable would throw out the argument, even though it might in fact be sound. But leaving questionable rules in is to trust that all applications of them in the argument are sound, which cannot be easily confirmed.

Checking the logical sufficiency of the representation turned out to be problematic for reasons not anticipated. Just showing that one can infer that low density is more likely in the tropics than in temperate forest is not enough: it could (and did) happen that this conclusion is reached in error by inferring that low density is true of the tropics and failing, because of a mistake in the representation, to show the same for temperate forest. Since Prolog uses the negation-as-failure principle, in which failure to be able to prove something is taken as establishing its negation, errors can easily produce spurious negatives, which in turn lead to spurious, though not necessarily incorrect, conclusions. In short, the Prolog representation can easily produce the correct overall conclusion on false grounds, and testing of various subsidiary inferences was necessary to gain confidence that the representation was in fact capturing what was intended.

As an example of these difficulties, in an early version of the representation it was asserted that tropical forests had large numbers of seeds close to the parent tree. Failure to assert this same fact for temperate forests meant that spurious contrasts were inferred. In particular, it was possible to prove both that low density was more likely in the tropics and that it was less likely. Thus, the omission of information about seed numbers permitted the inference that low seed numbers near the parent, which would produce low density, was more likely in temperate forests.

These difficulties, and a general difficulty in seeing what the Prolog inference process is doing, reflect the fact the the Prolog representation is not really a representation of the argument but rather a representation from which the argument can be automatically recreated. The recreated argument, the proof which Prolog constructs internally, is not shown explicitly and is hard to check. A formal representation in which a trace of the argument is explicitly created as part of building the representation might be an advance.

Another limitation of the Prolog representation, which was not crucial for this argument in the end, though it appeared relevant in earlier stages of comprehending the argument, is in representing multiple arguments with the same conclusion. In logic multiple proofs of an assertion are redundant, and do not increase confidence in the conclusion, while in practical discourse multiple lines of inference often increase the force of an argument. It would presumably be possible to construct an inference system, even within Prolog, that would count the number of distinct arguments leading to a given conclusion.

A final difficulty with the Prolog representation arose in linking representations of separate parts of the argument. In particular, part of the argument deals with predation in general and how it can be escaped temporarily, and part deals with the effects of host-specific predation in particular on survival near the parent. The Prolog representation created here did not explicitly model the interplay of these effects but simply asserted that high predation, in the absence of escape, leads to low survival. This way of joining the pieces of the argument works but leaves a possibly important part of the argument incompletely examined.

<u>Toulmin Representation.</u>
The representation of the argument in Toulmin structures shown in Figure 4 was constructed from the Prolog representation just described. Even though a fresh analysis was not made, some new points emerged. First, it became apparent that the Prolog representation had failed to capture the highest level structure of the argument, leading to the conclusion that predation explains tree species distribution in the tropics. The Prolog representation demonstrated a crucial part of this argument, namely that appropriate assertions about distribution could be derived from assertions about predation, but the step from that demonstration to a conclusion about explanation was not represented. This point is of some importance in view of later discussion of the Janzen paper in the literature, where the identity of the top-level claim of the paper is an issue.

The exercise of building the Toulmin representation led directly to this revision of the representation as built in Prolog. In the Toulmin form the top-level claim must be represented explicitly, whereas in the Prolog representation the top-level claim is not part of the representation but is presented as something to be proven, which can be (and was, in this case) specified later.

Once the top-level claim and immediately supporting argumentation are identified they can be seen to be weak. To show that host-specific predation causes sparse distribution one has to show that in all cases in which sparse distribution does not occur host-specific predation is absent or overridden in its effects. The paper argues this result for temperate forests, for mangrove forests, and for some particular species that occur without their predators, but obviously it cannot deal with all such cases. Further, even if all cases were dealt with it could be that some other factor, with a similar distribution, is instead the cause.

Since the Toulmin representation is not formal, some steps in the argument could be expressed more naturally in this form than in the Prolog form. In particular, it was not necessary to state general inference rules about predicates like "more" and "produces", which were seen as problematic in the Prolog representation. Rather, specific applications of such rules were used only when needed.

The Toulmin representation shows the structure of the argument explicitly, unlike the Prolog representation, so it is easier to check. On the other hand, it is not easy to explore what could be, but is not, concluded about temperate forests. As was noted above, errors in the Prolog representation were detected by showing that unintended inferences were possible; no comparable check seems possible in the Toulmin form.

EUCLID Representation.

Figure 5 shows a graphical presentation of the Toulmin analysis. This can be seen as a simple application of the EUCLID representation, where the only relationship between claims is support. The Toulmin distinction between datum and warrant is discarded, and the graphical presentation makes it easy to display overlapping Toulmin structures without redundancy, but otherwise this representation is equivalent to the textual presentation in Figure 4. The graphical presentation seems quite readable, at least for an argument of this scale. The following issues arose in preparing it.

Packing the diagram into a small space is important. The facilities in the EUCLID version 2 were clumsy to work with (entry and connection of claims was awkward) and the graphical layout constraints it implemented were too rigid (for example support relations were always mapped to indentation, making it difficult for a claim to support more than one other claim). The figure shown was produced not using EUCLID but rather Claris MacDraw running on a Macintosh computer. Facilities to tie connections between claims, and to provide automatically-sized borders for claims, and to rearrange the resulting tree, would have eased the task. However, even without these facilities it was not difficult to prepare the new representation from the Toulmin representation. On the other hand preparing the representation from the text of the article without this support would have been tedious because of the need to rearrange and re-edit claims repeatedly.

EUCLID version 4 no longer uses built in constraints to layout the claims, nor does it use indentation to show support. In the new version the user has complete control over positioning of the claims. Relationships like support are indicated by labeled lines connecting the relevant claims.

As with the parent Toulmin representation, this representation is passive. There is no system in the background capable of displaying selected portions of the argument, for example unsupported claims. In the EUCLID design a graphical representation like that in Figure 5 would be backed by an internal representation called ARL which would be used to format the display and to support selective viewing.

It is not clear that such facilities would contribute much in this case study. The graphical layout makes spotting unsupported premises fairly easy, at this scale. The value of at least manual override of formatting in packing the representation into a small space may outweigh the convenience of automatic layout for complicated argument structures. Further, no complicated structures, for which ARL might provide useful discipline, were used in this example.

As noted, this graphical representation was produced from the Toulmin representation, which in turn was produced after the Prolog formalization was constructed. We have already noted that this probably artificially reduced the value of higher-level tools for editing the representation. It may also have masked another problem with less-structured representations: the problem of choosing an appropriate level of granularity for the analysis. What is to prevent the analyst from producing a trivial representation with just a few claims connected together, like that in Figure 6? It might appear that the more structured Prolog and Toulmin representations would force a deeper analysis.

On reflection, however, this seems not to be the case. Figures 7 and 8 show Prolog and Toulmin versions of the reduced representation in Figure 6. The missing warrants in the Toulmin diagram, and the questionable use of a claim as a warrant in the last structure, signal that something has been left out of the unconstrained diagram. However, it is easy to see that these problems could be dealt with without making the representation much more complete or detailed, simply by adding trivial warrants like "There is high predation near adults unless there is escape."

There is nothing to stop the analyst from producing such reduced representations in any of these schemes. The check on the analyst seems to be an internal one: at what level is the analyst satisfied that the interesting issues in the argument have been captured?

THE FETZER ARGUMENT.

Summary of Article.

Fetzer's argument is two-pronged. He argues that there are fatal flaws in an earlier argument by (DeMillo et al 79) to the effect that computer programs are not susceptible to absolute verification. The idea of absolute verification relates to the idea of being able to absolutely guarantee that a program will operate correctly, as opposed to an idea of relative verifiability in which program correctness is only expressed as a probability of correctness. Fetzer does agree with their conclusion and he goes on to present his own argument as to why that conclusion is correct. While presenting these arguments, Fetzer also presents arguments as to why it is important to argue about absolute verifiability.

EUCLID Representation.

The EUCLID version 2 representation of the Fetzer article appears in Figure 9. In constructing this model Hair used the Claris MacPaint program running on a Macintosh computer. The EUCLID version 4 representation is given in Figure 10.

The EUCLID model of argument seemed useful in focusing the user's attention on the underlying structure of the argument. The resulting argument schema did not appear to be particularly useful as a means of presenting the argument to a third party, however. What results from representing a lengthy argument in EUCLID tends to be simply a lengthy series of text boxes which do not seem particularly better than a prose representation. Moreover, here again there is the possible problem arising from the fact that there is no one correct way to represent an argument with the model, so it seems unlikely that a given person's model will necessarily be useful to other people.

It was while working on the EUCLID representation that Hair perceived a previously unnoticed problem with Fetzer's argument. The problem lies in the fact that at two points in the paper Fetzer seems about to assert that not even relative verifiability can be achieved for programs. However, he does not actually reach that conclusion. The problem lies in letting a possibly unwary reader finish the article thinking that a conclusion had been reached that had not been reached. In fact, much of the later correspondence in CACM about this article center on people in part seeming to think that Fetzer had argued that even relative verifiability was not possible (CACM, March, April, July, August 89) (Dobson and Randell 89).

Toulmin Representation.

The Toulmin representation of Fetzer's article is shown in Figure 11. The use of the Toulmin structures was similar to using the EUCLID model to represent arguments. A further observation about the Toulmin structures is that they do not readily lend themselves to showing possible interrelationships between arguments. Thus, the structure suggests the need to break down large arguments into what might be thought of as microarguments, or individual pieces of self-contained argument. Although the pieces can be used as data, warrants or the like in other argument structures, it did not seem natural to link up all parts of the overall Fetzer argument in this way. Moreover, there simply does not seem to be any way to link together discussions of opposing arguments.

It became necessary to set forth a series of unconnected argument pieces. Thus, it was possible to connect together the separate arguments Fetzer gives about why (DeMillo et al 79) had incorrect arguments, but there was no direct way of using Toulmin structures to connect together arguments for and against a given position. Also, there was no clear way to connect the argument about why the main argument was important to the other arguments.

In using the Toulmin structures, it seemed clumsy to force there to always be the three necessary parts: data, claim, and warrant. It appears that there are arguments constructed such that it would be natural to leave out one or another of those parts. Also, decisions as to which part was which tended to be somewhat arbitrary. This observation agrees with those critics of the Toulmin structure who argue that the distinction between data and warrant is ambiguous. In addition, no use at all was made of the backing, rebuttal, or qualifier parts of the structure.

Prolog representation.

The Prolog representation of the Fetzer article is in Figure 12. Prolog representations seem inherently difficult for an outside person to comprehend. In fact, a problem with all of the three representations produced may lie in the fact that decisions as to how to represent the parts of an argument tend to be subjective, so that even in using the same model it is unlikely that two people would represent an argument the same way. Thus, a question arises as to whether another person's comprehension would be aided by a representation that is different from what their own independently produced representation would look like. The specific problem with a Prolog representation is that the argument parts are necessarily made cryptic, which eliminates a lot of semantic information and is generally not easy for people to follow.

On the other hand, Prolog does offer a very concise method of representing the underlying structure of an argument. Also, the Prolog representation can be used as an actual program which can be executed to test the validity of various assumptions.

As with the Toulmin structures, it appeared quite difficult to link together different parts of the argument. Thus, the program wound up as basically three separate parts: one for the argument about the arguments of (DeMillo et al 79), one for the main argument, and one for the argument about why the overall argument was important.

It was while working on this model that an insight into the overall structure of Fetzer's argument was realized. In reanalyzing Fetzer's attack on the arguments of (DeMillo et al 79), it occurred to Hair for the first time that Fetzer actually uses part of their argument to lead into his own discussion of what he thinks the proper argument should be. What he does is to make a somewhat subtle argumentation move in which he points to their refusal to equate algorithms with programs, gives what he views as the correct reasoning for making such a distinction, and then uses that distinction as a central ground for concluding that absolute program verification is impossible.

Another insight realized while using the Prolog model was that Fetzer seemed to play a little loosely with ideas about complexity. At one point in arguing that (DeMillo et al 79) had a weak argument, he notes that they had argued in part that programs were too complex for formal verification to work. He suggests at that point that one can picture two kinds of complexity in programs, one relating to a patch-work quality resulting from the incorporation of things like error messages and user interfaces and one relating to programs that may simply be complex because they are large, but straightforward, aggregates of smaller programs. At the point where he criticizes them, he says they are only looking at patch-work complexity but that such complexity can be hypothesized away for purposes of discussion. The implication is that the more straightforward kind of complexity can be dealt with to achieve absolute verification.

He goes on to argue that even where there is no patch-work complexity the conclusion remains that absolute verifiability is an impossibility. However, he makes his argument by effectually positing what can only be viewed as another kind of complexity, namely the complexity of the environment. His argument is to the effect that for causal models like programs there is simply too complex a real world environment to permit the possibility of absolute verifiability, although he does not phrase the argument in quite those terms. Thus, he refers to factors in the environment concerning the actual hardware being used, operating systems, compilers, and so forth.

A possible criticism, then, is that he might have strengthened his discussion if he had focused more on what he himself was saying about complexity. On the one hand he argues that complexity is not the real problem, but then it appears that complexity really is the problem.

Further observations.

As noted at page 10, Fetzer's argument has two goals. It is in part this split approach that led to some of the problems Hair had in using the three schemes for modelling the argument. It seemed preferable to set forth the argument parts in a single structure in order to capture the fact that there is a relationship between all of the parts. However, it only seemed possible to achieve a single structure by using the EUCLID model. The result was achieved simply by stating the main claim as a claim of two things being true. The EUCLID structure is then elastic enough to permit fitting all of the other argument parts together. Thus, in EUCLID it is only necessary to imagine all claims as either supporting or refuting some other claim, where one can allow the notions of "support" and "refute" to take on rather imprecise meanings.

This flexibility solved a particular problem that Hair only realized was a representation problem when he proceeded to try other models. The problem lies in the fact that a part of the Fetzer argument is an argument about the argument. Thus, he argues that the argument about the possibility of absolute verifiability is important both because there are computer scientists who believe that it is possible, and because the consequences of believing that it is possible are important. While this argument seemed to fit in more or less naturally as an argument in support of the main claim in the EUCLID model, as discussed above it did not fit so neatly in other models.

One of the main observations Hair reached while doing these representations, and particularly while working on the EUCLID model, is that Fetzer's argument seemed to skip around somewhat. Thus, in developing the EUCLID representation Hair worked up a simulation of the EUCLID structure using MacPaint and appended page numbers to the various claims (see Figure 9). Although for the most part the claim order seems to match the page order, there are a few points where it is quite clear that the page order diverges a lot from Hair's view of how the claims are interconnected.

Three specific problems with the argument arose because of this failure to order the text so as to have related claims discussed together. One initial problem with the argument was in seeing how the concluding part of the article really fit into the overall argument. That is the section of the article in which Fetzer discusses the possibly disastrous results that could come about in real life applications if people erroneously believed that absolute verification was achievable. It was only in reflecting on the fact that the first part of the article seemed to be a justification of the importance of the topic that it appeared that the last part is aimed at the same thing. The other part that caused trouble was in trying to see if Fetzer was basically equating relative verifiability with inductive reasoning. The conclusion is that he does equate them, but he never really says that and the parts of the discussion about inductive reasoning and relative verifiability are separated. The third problem relates to the complexity problems discussed above. Again, since the relevant parts of the discussion were not in proximity it took longer to see the connection than might otherwise have been the case.

FOLLOWUP LITERATURE ABOUT THE TWO ARTICLES.

Subsequent developments concerning Janzen's article.

A number of subsequent papers have commented on Janzen's work, permitting a comparison between weaknesses detected by argument analysis with weaknesses noted in the literature. None of the weaknesses found in our analysis, or for that matter any other, is explicitly referred to in follow-on papers. The commentary does suggest revisions to Janzen's analysis which relate implicitly to the points turned up here.

Papers by (Hubbell 79) and (Connell 78) argue that disturbance of forests, not predation, is the dominant determinant of species distribution. This approach is essentially the alternate cause argument anticipated above. These authors do not criticize Janzen's argument but simply argue that the effects Janzen discussed are swamped by more powerful forces. The paper by (Hart et al 89) discusses a tropical forest in which a single species dominates, and suggests that it differs from neighboring diverse forest in the recency of major disturbance. This approach picks on the incompleteness of the causal argument in not dealing with all cases in which sparse distribution fails to occur.

Papers by (Connell 84) and (Clark&Clark 84) present data suggesting that density of juvenile trees, not distance from the parent, affects juvenile survival, despite Janzen's emphasis on the latter even when considering density-specific predation. Here again the commentators are not criticizing Janzen's argument but rather are suggesting that other influences are at work.

Hubbell attacks Janzen's top-level argument by criticizing a premise rather than the argument itself. Hubbell suggests that tropical forests in fact do not show the uniform scattering of species Janzen describes, but are instead clumped to some degree. Since this approach is an attack on a premise it is clearly inaccessible to argument analysis. Interestingly, it is possible that Janzen's model would actually produce the distribution Hubbell reports, and not that Janzen expects, because of the possibility of clumped juveniles noted above.

The Clarks reconcile Janzen's argument with later, apparently contrary data in an interesting way. They write as if Janzen was not trying to explain the distribution of forest trees, which he clearly was, but was simply describing a process by which predation could influence survival of juveniles. This more limited claim is clearly much easier to support than the actual top-level causal claim. The Clarks' interpretation is thus a generous one. It seems quite appropriate for scholars to seek to salvage valid, if more limited, claims from papers rather than focussing on refuting assertions that are too general. To the extent that argument analysis emphasizes criticism of claims interpreted literally it could be damaging to inquiry.

Data gathered after, and in some cases because of, Janzen's paper show that the actual determinants of species distribution are very variable. Different tree species behave differently. Indeed, one can see it as hopeless to attempt generalizations, as Janzen did, about the "tropical forest". Argument analysis does not seem to help in identifying this kind of difficulty, other than making the scope of generalizations used in the argument obvious.

Further Correspondence About Fetzer.

The Fetzer article led to a large volume of responses in later volumes of the Communications of the ACM (CACM, March, April, July, August 89) (Dobson and Randell 89). Interestingly, a number of the responses strongly attacked Fetzer without seeming to really address his argument. There seemed to be a feeling that Fetzer had suggested that program verification was of no value whatever. In answering these criticisms, Fetzer notes that he made no such claim and that he specifically allowed for program verification to have the ability to provide relative verifiability. There seem to be three likely reasons for the critics to have apparently missed this point. First, as some of the people more sympathetic to Fetzer have suggested, people deeply engaged in a given field may tend to overreact to any kind of criticism. Second, as discussed above, there are points in the original article where it seems that Fetzer is going to suggest that even relative verifiability is not possible even though he does stop short of actually saying that. Third, the tone of the article is highly critical of program verification and there is no place in the article itself where Fetzer directly concedes that there is any value to such research.

Another dominant theme of Fetzer's critics is to dismiss his main argument as obvious and unimportant. Thus, most of them apparently concede that absolute verification is not possible. They assert that no one seriously believes that it is possible so that there is no point in arguing about it. Therefore, they are really arguing against Fetzer's argument about why his argument is important.

CONCLUSIONS.

The Representation Schemes.

(1) Prolog.

Our interest in using Prolog to represent arguments stemmed from the facts that Prolog itself is based on formal logic, and that such representations would be subject to automatic consistency checking. In practice, the use of Prolog proved disappointing.

Even though Prolog is based on formal logic, it was necessary to separately program various rules that do not directly flow from Prolog's internal logic. While this effort led to increasing our awareness that the arguments relied on such rules, it was cumbersome to have to make the rules explicit. Moreover, the need for these rules added to the general problem of the representation requiring too much effort in formalizing uncontroversial parts of arguments.

Prolog representations are concise, but probably too cryptic for third parties to use them with any facility. The automatic checking is not adequate. It would be desirable for Prolog to be able to display proof structures. Prolog is the only scheme of the three we used that does not render a graphical representation.

(2) Toulmin structures.

Unlike the Prolog schemes, Toulmin structures led to the need for stating a top level claim, and in the case of the Janzen argument the statement of that claim led to revising the Prolog representation. Also unlike the case with Prolog, the Toulmin structures establish an explicit graphical representation. Because this representation method is not formal, it is easier to express a variety of ideas.

On the minus side, Toulmin structures have no inherent automated checking facility. There were problems noted in connecting different parts of a large argument into a unified structure. It was somewhat clumsy and arbitrary trying to break up arguments into the data/claim/warrant constituents. Finally, it turned out that neither of us used any of the parts of the Toulmin structure other than the data, claim, and warrant.

(3) EUCLID.

Many of the observations about the Toulmin structures apply to the EUCLID scheme. Here too a top level claim was needed, there is a graphical representation, and it is easy to express ideas. Also, there is no automated checking.

However, the EUCLID scheme was simpler than the Toulmin structures. That simplicity made it relatively easy to link together the entire body of large arguments. There was also less arbitrariness in dividing the arguments up into pieces, since here the only pieces are claims and relations between claims. These observations about the power or representation resulting from simplicity can be seen as reflecting the argument of (Cowan 64) that only two parts are needed to represent arguments.

Benefits to Users.

(1) Person building representation of existing argument.
In using these models to represent arguments we did develop increasingly deeper personal analyses of the arguments. However, we cannot directly attribute specific new insights to specific models. It may simply be the case that any kind of in-depth analysis effort would lead to similar insights. On the other hand, it may be worthwhile to conduct empirical work designed to test whether particular schemes lead to particular ways of understanding arguments.

(2) Person using an existing representation.
We find it doubtful that the representations we constructed would assist a third party's analysis of the arguments. As noted above, the Prolog representations are extremely cryptic, but it is also true that all three representations are somewhat cryptic. Moreover, in all three cases the representations were subjective, which suggests that others might not find that the representations capture their own analytic insights.

We do not conclude that it is hopeless to construct representations that could aid third party analysis. However, we do conclude that considerable effort would be necessary in order to overcome these problems.

(3) Person creating a new argument.
Since we did not create new arguments using these schemes, our evidence about this area is indirect. It appears that the use of schematic representations could have significantly improved the presentation of the arguments. Thus, the analysis required for using these schemes led to clarification of what the issues actually were, and of how the issues related to each other. However, we had no results of the type reported

by (VanLehn 85) where he attributed the acquisition of particular insights to the use of his particular representation scheme.

Follow-up Literature.

In general, the argument weaknesses we turned up while developing our representations are not reflected in the follow-up literature in that even direct critics of the arguments do not criticize them for the weaknesses we found. However, the insights gained in the representation process were useful in understanding how the subsequent commentary related to the arguments.

REFERENCES.

Anderson, J.M. and Dovre, P.J. (eds.), *Readings in Argumentation*, Allyn and Bacon, Inc., 1968.

Barth, E.M. and Martens, J.L. (eds.), *Argumentation Approaches to Theory Formation*, John Benjamins, 1982.

Bettinghaus, E.P., "Structure and argument," in Miller, G.R. and Nilsen, T.R. (eds.), *Perspectives on Argumentation*, pp. 130-155, Scott, Foresman and Company, 1986.

Brockriede, W. and Ehninger, D., "Toulmin on argument: An interpretation and application," *Quarterly Journal of Speech*, 46:44-53, February, 1960.

Burleson, B.R., "On the foundations of rationality: Toulmin, Habermas, and the 'a priori' of reason," *Journal of the American Forensic Association*, 16:112-127, Fall, 1979.

Clark, D.A. and Clark, D.B., "Spacing Dynamics of a Tropical Rain Forest Tree: Evaluation of the Janzen-Connell Model," *The American Naturalist*, 124:769-788, December, 1984.

Communications of the ACM, ACM Forum, 32:287-290, March, 1989.

Communications of the ACM, Technical Correspondence, 32:374-381, March, 1989.

Communications of the ACM, Technical Correspondence, 32:506-512, April, 1989.

Communications of the ACM, ACM Forum, 32:790-792, July, 1989.

Communications of the ACM, ACM Forum, 32:920-921, August, 1989.

Connell, J.H., "Diversity in Tropical Rain Forests and Coral Reefs," *Science*, 199:1302-1310, March, 1978.

Connell, J.H., Tracey, J.G., and Webb, L.J., "Compensatory Recruitment, Growth, and Mortality As Factors Maintaining Rain Forest Tree Diversity," *Ecological Monographs*, 54(2):141-164, 1984.

Cooley, J.C., "On Mr. Toulmin's revolution in logic," *Journal of Philosophy*, 56:297-319, 1959.

Cowan, J.L., "The uses of argument--an apology for logic," *Mind*, 73:27-45, January, 1964.

Cox, J.R. and Willard, C.A. (eds.), *Advances in Argumentation Theory and Research*, Southern Illinois University Press, 1982.

DeMillo, R., Lipton, R., and Perlis, A., "Social processes and proofs of theorems and programs," *Communications of the ACM*, 22:271-280, May, 1979.

Dobson, J. and Randell, B., "Program verification: Public image and private reality," *Communications of the ACM*, 32:420-422, April, 1989.

Fetzer, J.H., "Program verification: The very idea," *Communications of the ACM*, 31:1048-1063, September,1988.

Fogelin, R.J., *Understanding Arguments: An Introduction to Formal Logic*, 3rd edition, Harcourt Brace, 1987.

Golden, J.L. and Pilotta, J.J. (eds.), *Practical Reasoning in Human Affairs*, D. Reidel Publishing Company, 1986.

Hample, D., "Modeling argument," in Cox and Willard, 1982.

Hart, T.B., Hart, J.A., and Murphy, P.G., "Monodominant and species-rich forests of the humid tropics: Causes for their co-occurrence," *The American Naturalist*, 133:613-633, May, 1989.

Hubbell, S.P., "Tree Dispersion, Abundance, and Diversity in a Tropical Dry Forest, *Science*, 203:1299-1309, March, 1979.

Janzen, D.H., "Herbivores and the number of tree species in tropical forests," *American Naturalist*, 104:501-528, 1970.

Kneupper, C.W., "On argument and diagrams," *Journal of the American Forensic Association*, 16:181-186, Spring, 1978.

Manicas, P.T., "On Toulmin's contribution to logic and argumentation," *Journal of the American Forensic Association*, 3:83-94, 1966.

Marshall, C.C., "Exploring representation problems using hypertext," *Proceedings of Hypertext-87*, pp. 253-268, Chapel Hill, North Carolina, 1987.

Marshall, C.C., "Representing the structure of a legal argument," *Proceedings of the Second International Conference on Artificial Intelligence and Law*, pp. 121-127, 1989.

McCroskey, J.C., "Toulmin and the basic course," *Speech Teacher*, 14:91-100, 1965.

Perelman, C. and Olbrechts-Tyteca, L., *The New Rhetoric: A Treatise On Argumentation*, University of Notre Dame Press, 1969.

Smolensky, P., Fox, B., King, R., and Lewis, C., "Computer-aided reasoned discourse, or, how to argue with a computer," *Cognitive Science and Its Applications for Human-Computer Interaction*, Guindon, R. (ed.), pp. 109-162, Lawrence Erlbaum, 1988.

Toulmin, S., *The Uses of Argument*, Cambridge University Press, 1958.

Toulmin, S., Rieke, R., and Janik, A., *An Introduction to Reasoning*, Macmillan Publishing Co., Inc., 1979.

van Eemeren, F.H., Grootendorst, R., and Kruiger, T., *The Study of Argumentation*, Irvington Publishers, 1984.

VanLehn, K., "Theory reform caused by an argumentation tool," Xerox PARC Technical Report ISL-9, 1985.

```
/*formalization of Janzen on tree distribution*/
/*rules for comparisons between habitats*/
/*these are marked i for internal to allow unmarked name to be
used elsewhere for facts*/
imore(X,H1,H2):-more(X,H1,H2).
imore(X,H1,H2):-produces(Y,X),imore(Y,H1,H2).
imore(X,H1,H2):-ihas(H1,X), not ihas(H2,X).
imore(and(A,B),H1,H2):-imore(A,H1,H2), not imore(B,H2,H1).
imore(and(A,B),H1,H2):-imore(B,H1,H2), not imore(A,H2,H1).
ihas(H,X):-has(H,X).
ihas(H,X):-produces(Y,X),ihas(H,Y).
ihas(H,and(A,B)):-ihas(H,A),ihas(H,B).
ihas(H,not(X)):-not ihas(H,X).
ihas(H,X):-more(X,H,Hp).
/*use qmore, qhas to ask about conclusions*/
qmore(X,H1,H2):-imore(X,H1,H2).
qhas(H,X):-ihas(H,X).

/*facts about connections among aspects of habitats*/
produces(weather_change,predator_fluctuation).
produces(predator_fluctuation,times_with_few_predators).
produces(times_with_few_predators,few_predators(goodtime)).
produces(and(few_predators(T),trees_can_reproduce(T)),
      juvenile_trees_escape).

produces(and(high_predation(D),not(juvenile_trees_escape)),
      low_survival(D)).
produces(host_specific_predation,high_predation(close)).
produces(low_survival(D),low_adult(D)).
produces(low_seed(D),low_adult(D)).
produces(low_adult(close),low_density).

/*facts about the habitats*/

/*assume*/
has(H,trees_can_reproduce(T)).
has(tropics,host_specific_predation).
has(temperate,host_specific_predation).
has(escapees_on_tropical_islands,not(host_specific_predation)).

more(weather_change,temperate,tropics).
more(weather_change,temperate,escapees_on_tropical_islands).
```

Figure 1-1. Prolog Representation of Janzen Article

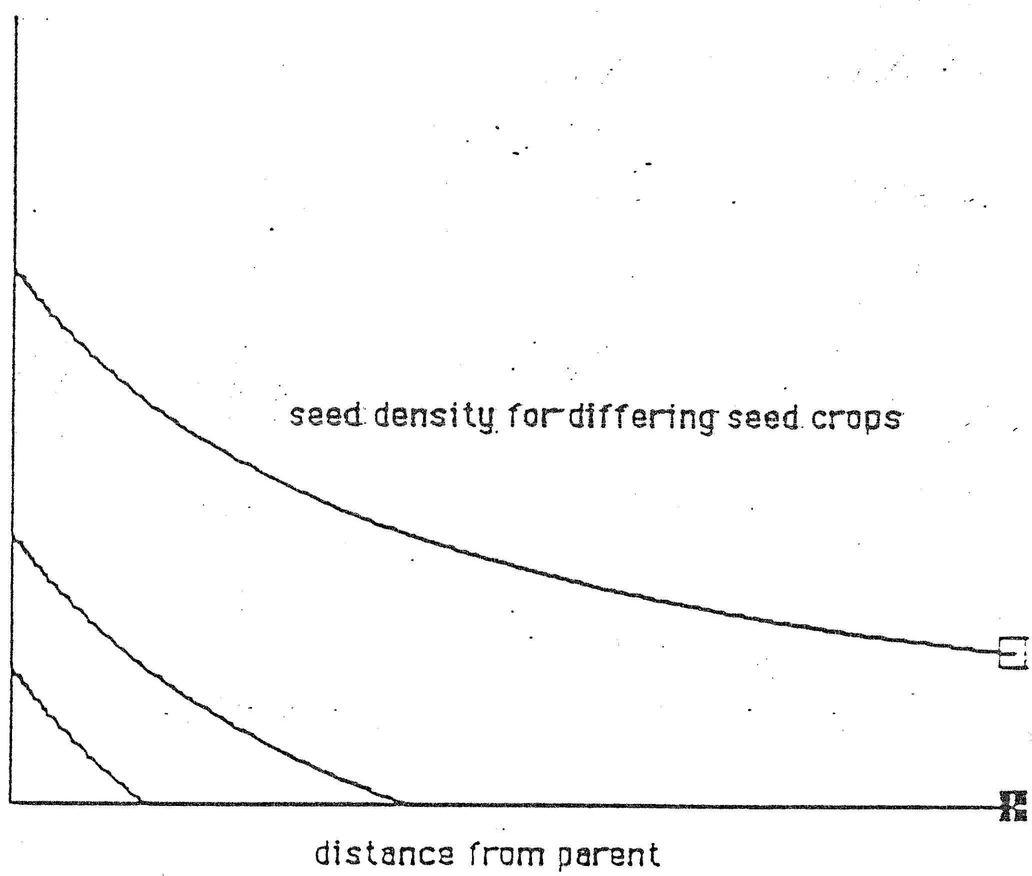


Figure 2-1. Janzen Related Graph Using Subtraction.

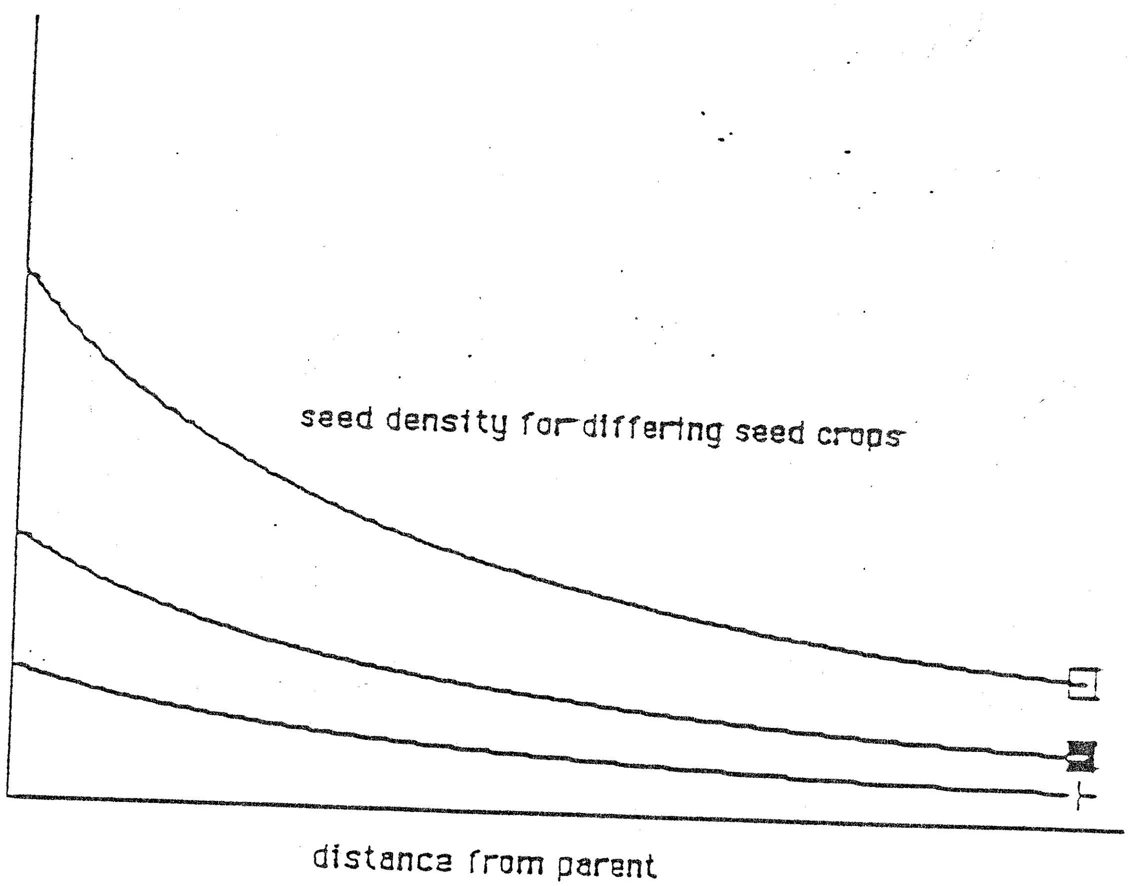


Figure 3-1. Janzen Related Graph Using Fractions.

Datum: There is weather change in temperate forest.
Warrant: Weather change produces predator fluctuation.
Claim: There is predator fluctuation in temperate forests.

Datum: There is predator fluctuation in temperate forests.
Warrant: Predator fluctuation produces times with few predators.
Claim: There are times with few predators in temperate forests.

Datum: There are times with few predators in temperate forests.
Warrant: None.
Claim: Trees in temperate forests can reproduce at times with few predators.

Datum: Trees in temperate forests can reproduce at times with few predators.
Warrant: Reproduction at times with few predators produces escape.
Claim: Trees in temperate forests can escape predation.

Datum: There is high seed near adults in forests.
Warrant: High seed produces high survival unless there is high predation.
Claim: There is high survival near adults in forests unless there is high predation.

Datum: There is host-specific predation in forests.
Warrant: Host-specific predation produces high predation near adults unless there is escape.
Claim: There is high predation near adults in forests unless there is escape.

Datum: Trees in tropical forests cannot escape predation.
Warrant: There is high predation near adults in forests unless there is escape.
Claim: There is high predation near adults in tropical forests.

Datum: There is high predation near adults in tropical forests.
Warrant: High predation produces low survival.
Claim: There is low survival near adults in tropical forests.

Datum: There is low survival near adults in tropical forests.
Warrant: Survival near adults determines density of adults.
Claim: There is low density near adults in tropical forests.

Datum: Argument that there is low density of adults in tropical forests.
Warrant: If A can be derived from B then B implies A.
Claim: Host-specific predation implies low density of adults in tropical forests.

Datum: Low density of adults occurs in tropical forests.
Warrant: Host-specific predation implies low density of adults in tropical forests.
Claim: Host specific predation implies low density of adults where it occurs.

Datum: There is high predation near adults in forests unless there is escape.
Warrant: There is high survival near adults in forests unless there is high predation.
Claim: There is high survival near adults in forests if there is escape.

Datum: Trees in temperate forests can escape predation.
Warrant: There is high survival near adults in forests if there is escape.
Claim: There is high survival near adults in temperate forests.

Datum: There is high survival near adults in temperate forests.
Warrant: Survival near adults determines density of adults.

Figure 4-1. Toulmin Representation of Janzen Article.

Claim: There is high density of adults in temperate forests.

Datum: Argument that there is high density of adults in temperate forests.
Warrant: If not A can be derived from B then B does not imply A.
Claim: Host-specific predation does not imply low-density of adults in temperate forests.

Datum: Low density of adults does not occur in temperate forests.
Warrant: Host-specific predation does not imply low-density of adults in temperate forests.
Claim: Host-specific predation does not imply low density of adults where it does not occur.

Datum: Host-specific predation does not imply low density of adults where it does not occur.
Warrant: if A implies B where B occurs and not where B does not occur then A may be a cause of B.
Claim: Host-specific predation may be a cause of low density of adults.

Figure 4-2. Toulmin Representation of Janzen Article.

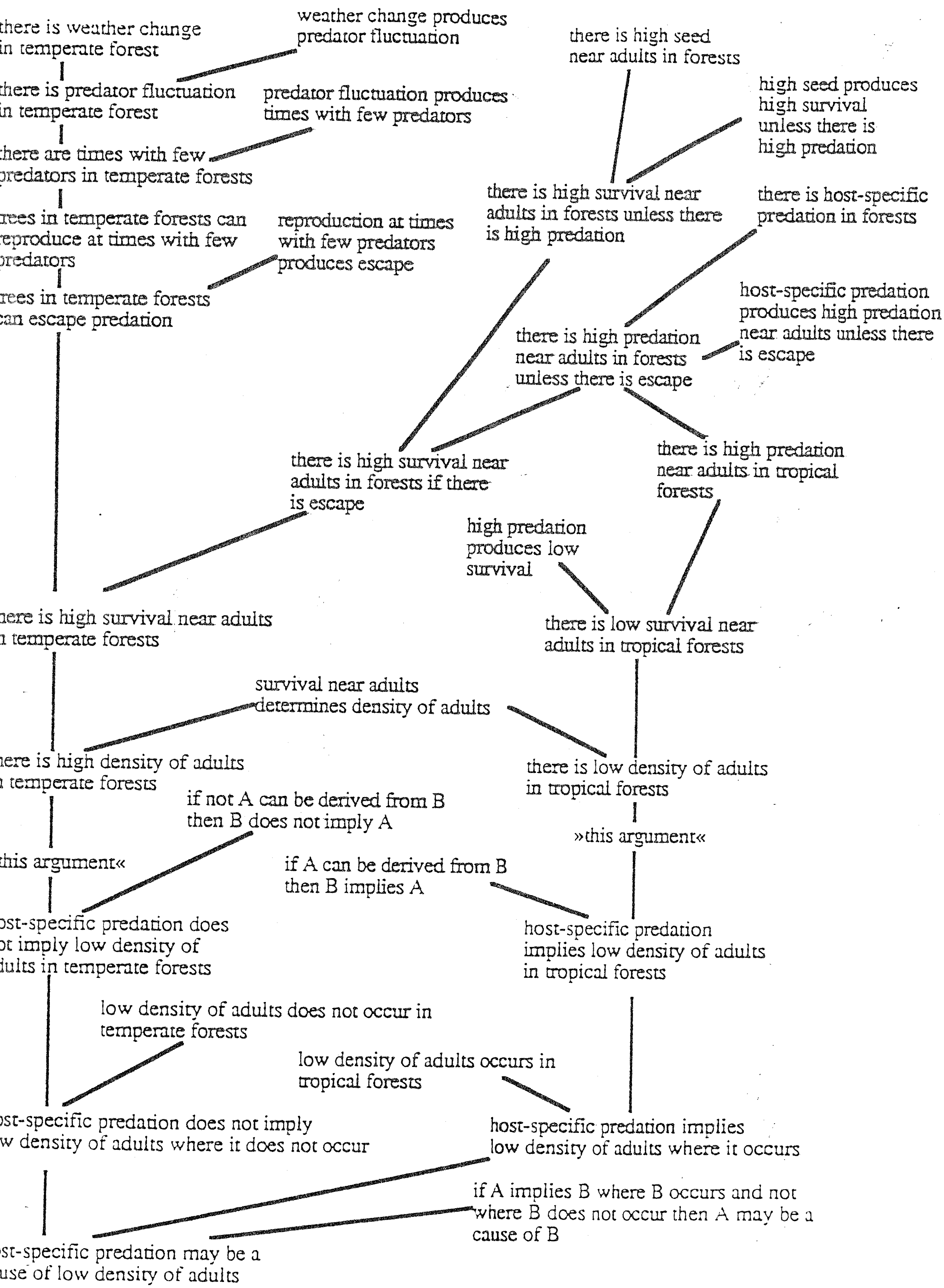


Figure 5-1. EUCLID Representation of Janzen Article

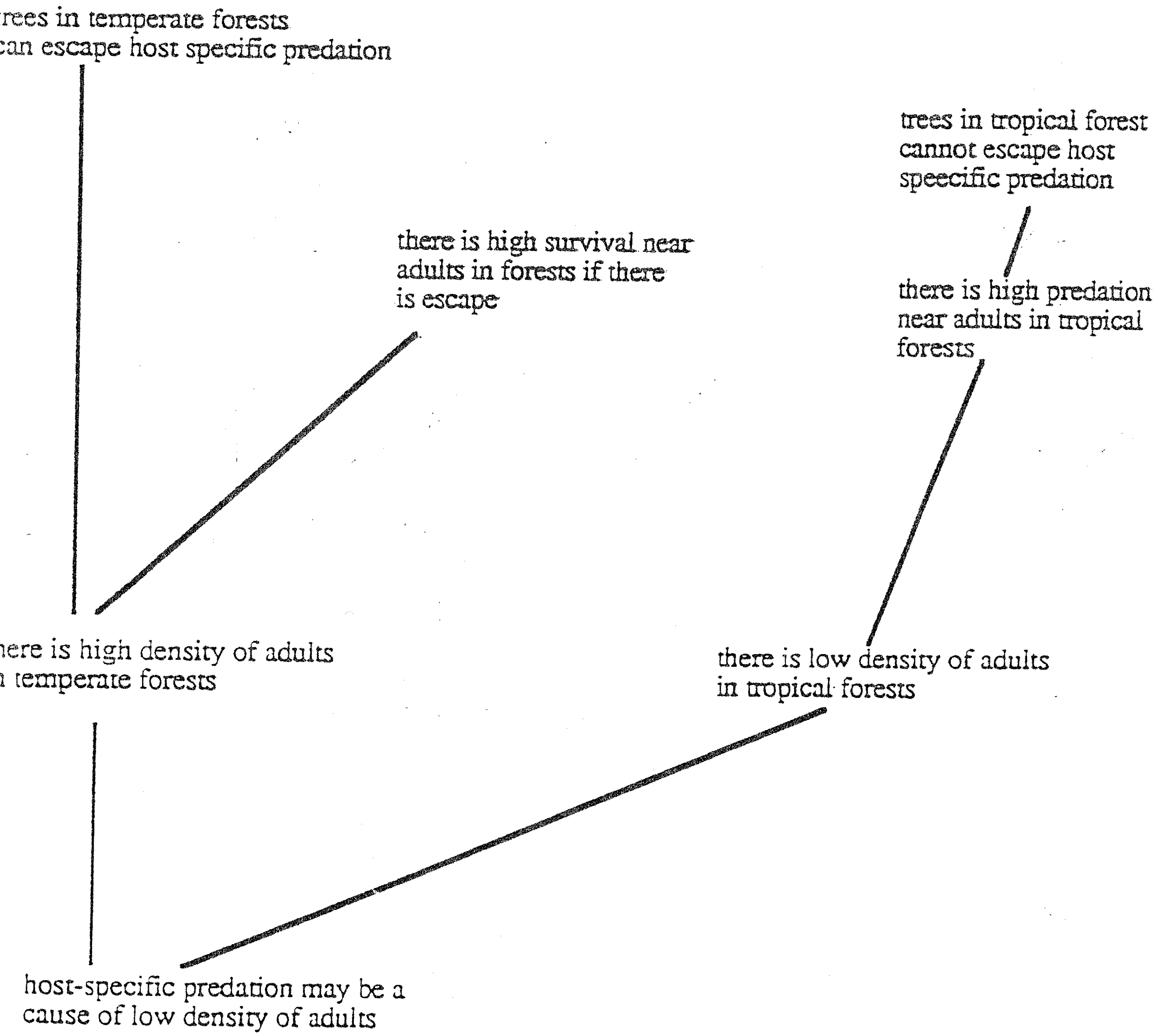


Figure 6-1. Reduced EUCLID Representation of Janzen Article.

```
/*formalization of Janzen on tree distribution*/
/*rules for comparisons between habitats*/
/*these are marked i for internal to allow unmarked name to be
used elsewhere for facts*/
imore(X,H1,H2):-more(X,H1,H2).
imore(X,H1,H2):-produces(Y,X),imore(Y,H1,H2).
imore(X,H1,H2):-ihas(H1,X), not ihas(H2,X).
imore(and(A,B),H1,H2):-imore(A,H1,H2), not imore(B,H2,H1).
imore(and(A,B),H1,H2):-imore(B,H1,H2), not imore(A,H2,H1).
ihas(H,X):-has(H,X).
ihas(H,X):-produces(Y,X),ihas(H,Y).
ihas(H,and(A,B)):-ihas(H,A),ihas(H,B).
ihas(H,not(X)):-not ihas(H,X).
ihas(H,X):-more(X,H,Hp).
/*use qmore, qhas to ask about conclusions*/
qmore(X,H1,H2):-imore(X,H1,H2).
qhas(H,X):-ihas(H,X).

/*facts about connections among aspects of habitats*/
produces(escape,high_survival(close)).
produces(not(escape),high_predation(close)).
produces(high_predation(close),low_density).

/*facts about the habitats*/

/*assume*/
has(temperate,escape).
has(tropics,not(escape)).

more(weather_change,temperate,tropics).
more(weather_change,temperate,escapees_on_tropical_islands).
```

Figure 7-1. Reduced Prolog Representation of Janzen Article

Datum: Trees in temperate forests can escape host specific predation.
Warrant: There is high survival near adults in forests if there is escape.
Claim: There is high density of adults in temperate forests.

Datum: Trees in tropical forests cannot escape host specific predation.
Warrant:
Claim: There is high predation near adults in tropical forests.

Datum: There is high predation near adults in tropical forects.
Warrant:
Claim: There is low density of adults in tropical forests.

Datum: There is high predation near adults in tropical forests.
Warrant: There is low density of adults in tropical forests.
Claim: Host-specific predation may be a cause of low density of adults.

Figure 8-1. Reduced Toulmin Representation of Janzen Article.

One cannot, in principle, obtain absolute program performance verification through program verification techniques. However, the arguments made in this regard by DeMillo, Lipton, and Perlis are not persuasive.

This is not a straw man argument.

Note: I really infer this argument from the later correspondence.

People like Hoare have suggested that guaranteed performance can be achieved with program verification.

1048, 1052

Furthermore, it is dangerous to let the idea go forward that absolute verification is possible, because the results can be disastrous in real world situations.

1062

DeMillo, Lipton, and Perlis argue for the same conclusion, but their reasoning is bad.

They argue that social processes like those in the mathematics field are not found in the program verification field. Therefore, program verification cannot succeed.

1048-49

But one could suppose that those processes could come into being, in principle.

1048

Also, while those processes do determine what is thought to be valid, they do not determine what actually is valid.

1050

They also argue that program verification is too tedious a process for real life programmers to undertake.

1052-53

But one could suppose an ideal programmer who would undertake it.

1052-53

Figure 9-1. EUCLID (v2) Representation of Fetzer Article

DeMillo, Lipton, and Perlis also argue that proofs can only express probabalistic proofs.

1053

That argument obviously does not hold if proofs are treated as deductive and hence absolute in nature. 1053

So they seem to advocate a view of mathematics as a domain of inductive procedure. 1054

This conclusion results from their view of the overriding complexity involved in trying to achieve deductive proofs. 1054

In this regard, they argue that the scaling up argument of program verification advocates will not hold up because real programs are not built up smoothly out of well defined pieces.

1054

But two kinds of complexity seem to be involved here. In fact, one can imagine programs having cumulative complexity being derived from simpler pieces. Demillo, Lipton, and Perlis seem to refer to what can be called patch-work complexity where their argument does hold.

1055

Cumulative complexity does seem amenable to scaling up, so there remains a theoretical interest in the possibility of absolute verification despite the DeMillo, Lipton, and Perlis argument.

1056

The real reason program verification cannot be used for absolute verifiability lies in the fact that programs are causal models suitable for execution, unlike algorithms which have no causal significance of physical counterparts. 1057

"Programs" are defined as particular implementations of algorithms suitable for machine execution. 1057

This distinction can be analogized to the distinction between applied and pure mathematics. One can achieve absolute deductively based proofs in pure mathematics, but only inductive proofs are possible in applied mathematics. 1060

Figure 9-2. EUCLID (v2) Representation of Fetzer Article

Reasoning about programs tends to be non-demonstrative, ampliative, and non-additive, whereas good deductive arguments tend to be demonstrative (true premises cannot yield false conclusions), non-ampliative (conclusions do not add information to the premises), and additive (the addition of more premises does not affect the argument's strength).

1051

Inductive arguments can be styled as knowledge-expanding while deductive arguments are truth-preserving.

1051

Therefore, programs are only susceptible to relative verifiability.

This distinction lies in the idea that in deductive systems it can be absolutely verified whether a consequence follows from the system's primitive axioms, while in inductive systems consequences must be determined based on premises whose truth is not verifiable.

1051

In the case of programs, there is simply too complex an overall system involving the interaction of software, firmware, hardware, and so forth for absolute verifiability to be possible.

1052

It could be argued that this could be accomplished by machine.

1061

But then, how would the verifying programs, themselves be verified?

1061

Programs appear to be syntactical entities, much like mathematical proofs. But programs differ in having a semantic significance relating to the performance of operations by machine.

1052

It is tempting to analogize between mathematical theorems and programs, but the analogy will not hold up.

1056

One can view the relation between theorems and proofs as the same as that between programs and verifications.

1056

Figure 9-3. EUCLID (v2) Representation of Fetzer Article

Extending that analogy, a program would be viewed as a function from inputs to outputs, corresponding to rules of inference that relate premises to conclusions where the analogy becomes less plausible. 1056

But the difference lies in the fact that while algorithms are functions from a domain to range, programs need not be. 1056 Note: this isn't really backed up.

The difference lies in the definition of programs as causal models of algorithms, where other possible definitions would result in different conclusions. 1058

This difference also relates to the difference between abstract and target machines, where target machines are physical things. As defined, programs are supposed to have physical target machine counterparts, while algorithms do not. 1058

The characteristics of abstract machines can be formalized, and are therefore susceptible to deductive proof techniques leading to absolute verifiability, while with programs there is relative verifiability at best owing to the physical relationship to target machines. 1058

A basic error in arguing for absolute program verification is to confuse the distinction between verifying what will happen in abstract machines with what will happen on actual physical machines. 1058

The difference is analogous to the difference between pure and applied mathematics, as in the following example. In pure mathematics 2 + 2 = 4 is always true, but in applied mathematics 2 units of water + 2 units of alcohol = 4 units of mixture may not be true.

1059

Figure 9-4. EUCLID (v2) Representation of Fetzer Article

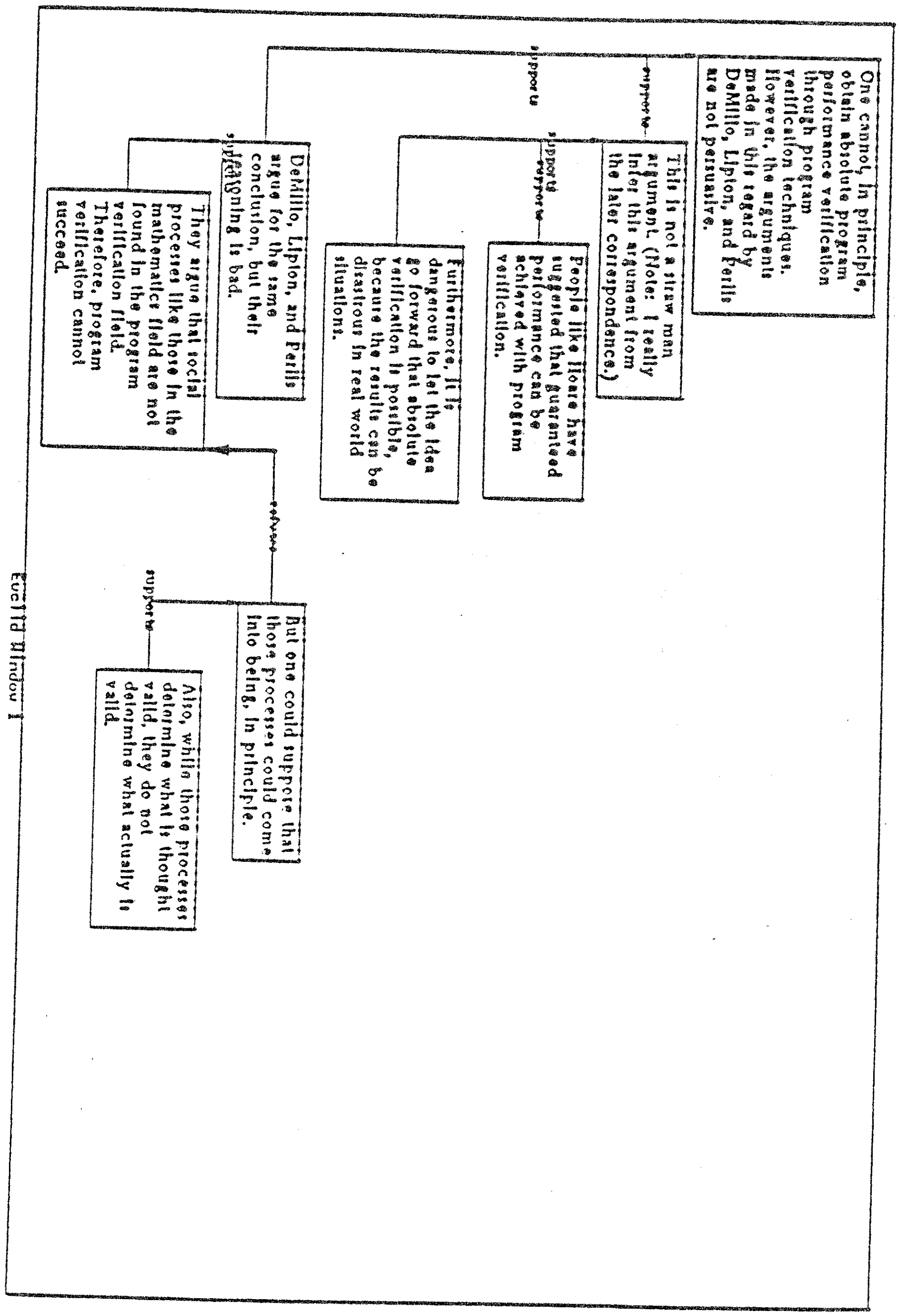


Figure 10-1. EUCLID (v4) Representation of Fetzer Article

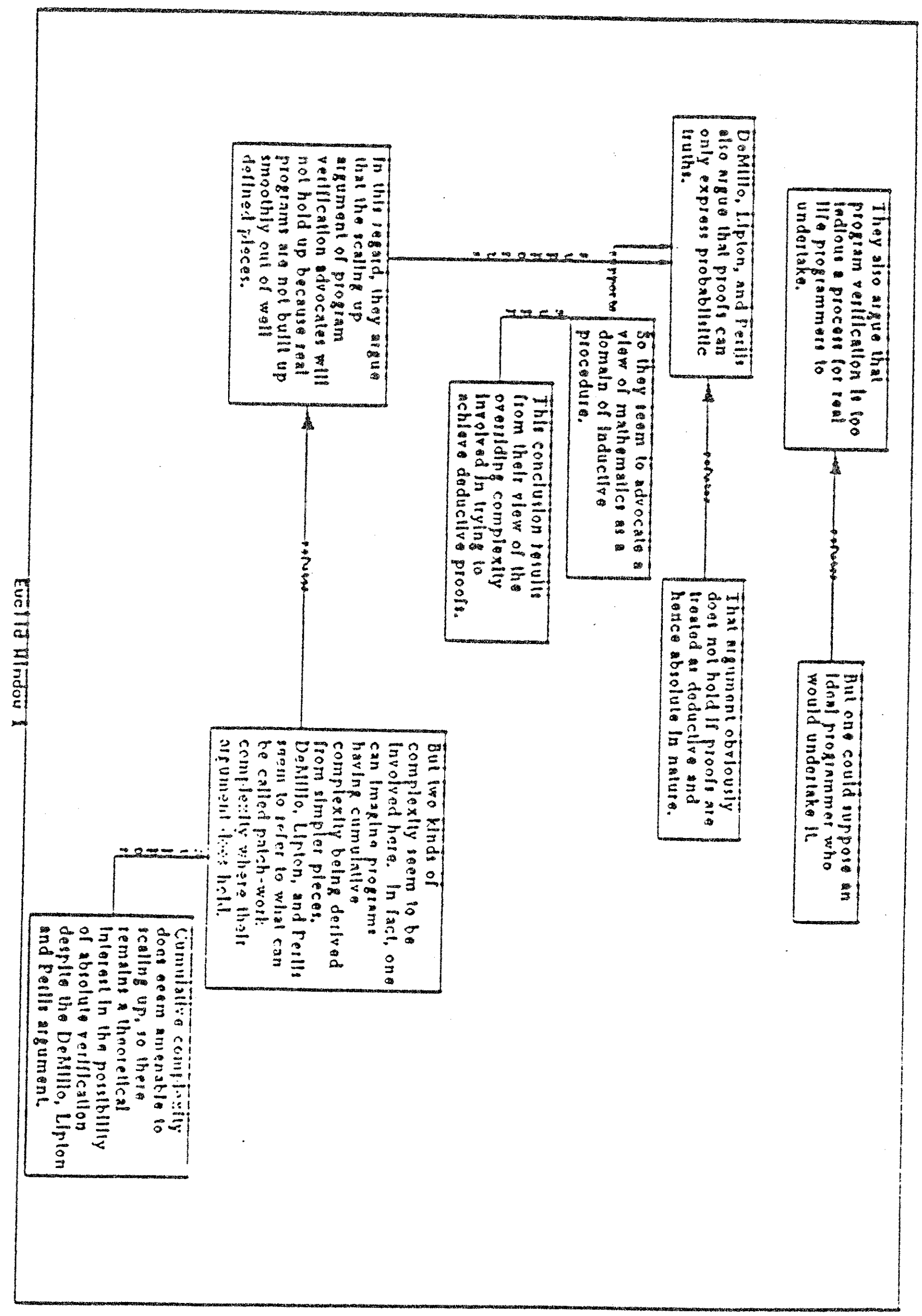


Figure 10-2. EUCLID (v4) Representation of Fetzer Article

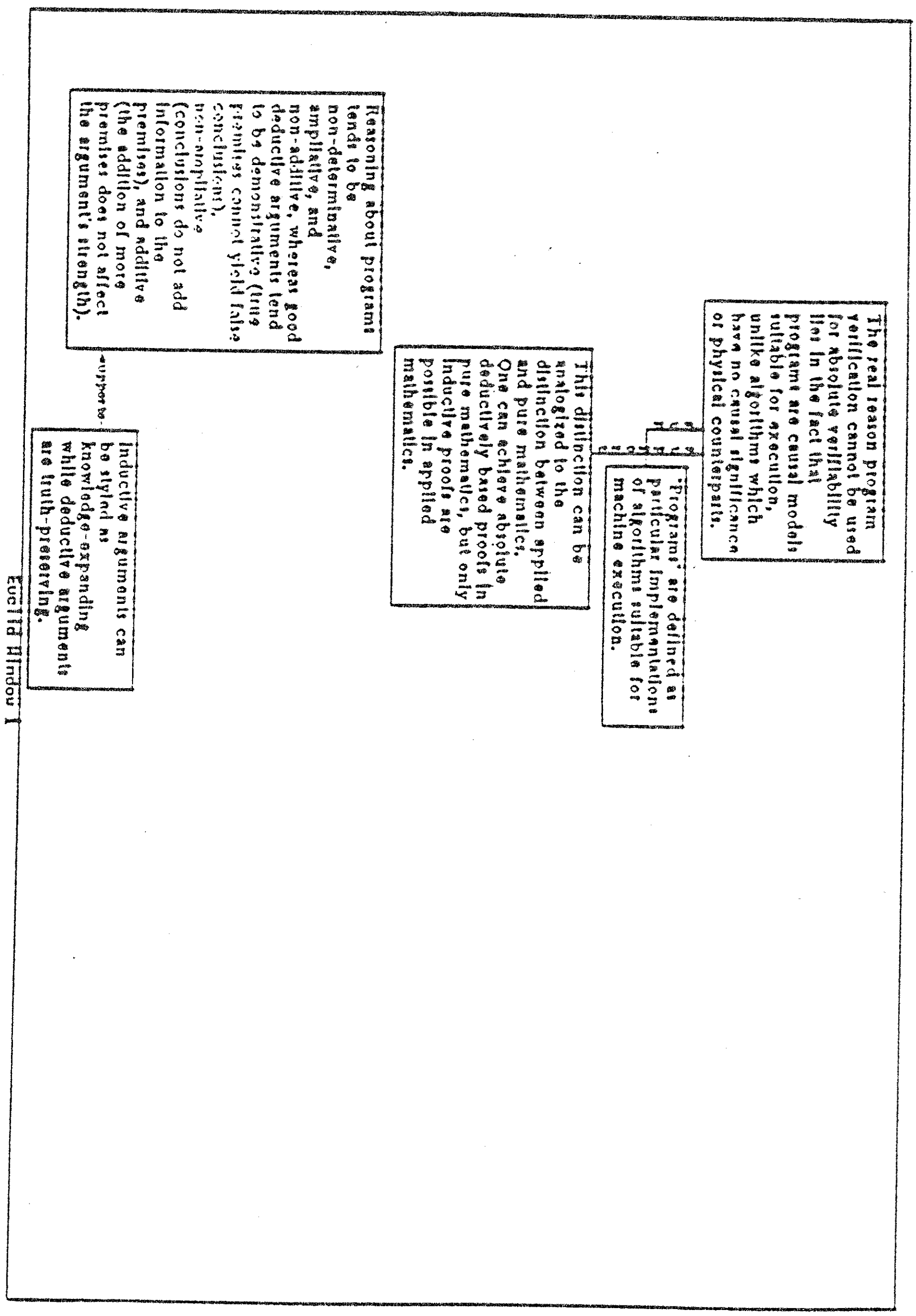


Figure 10-3. EUCLID (v4) Representation of Fetzer Article

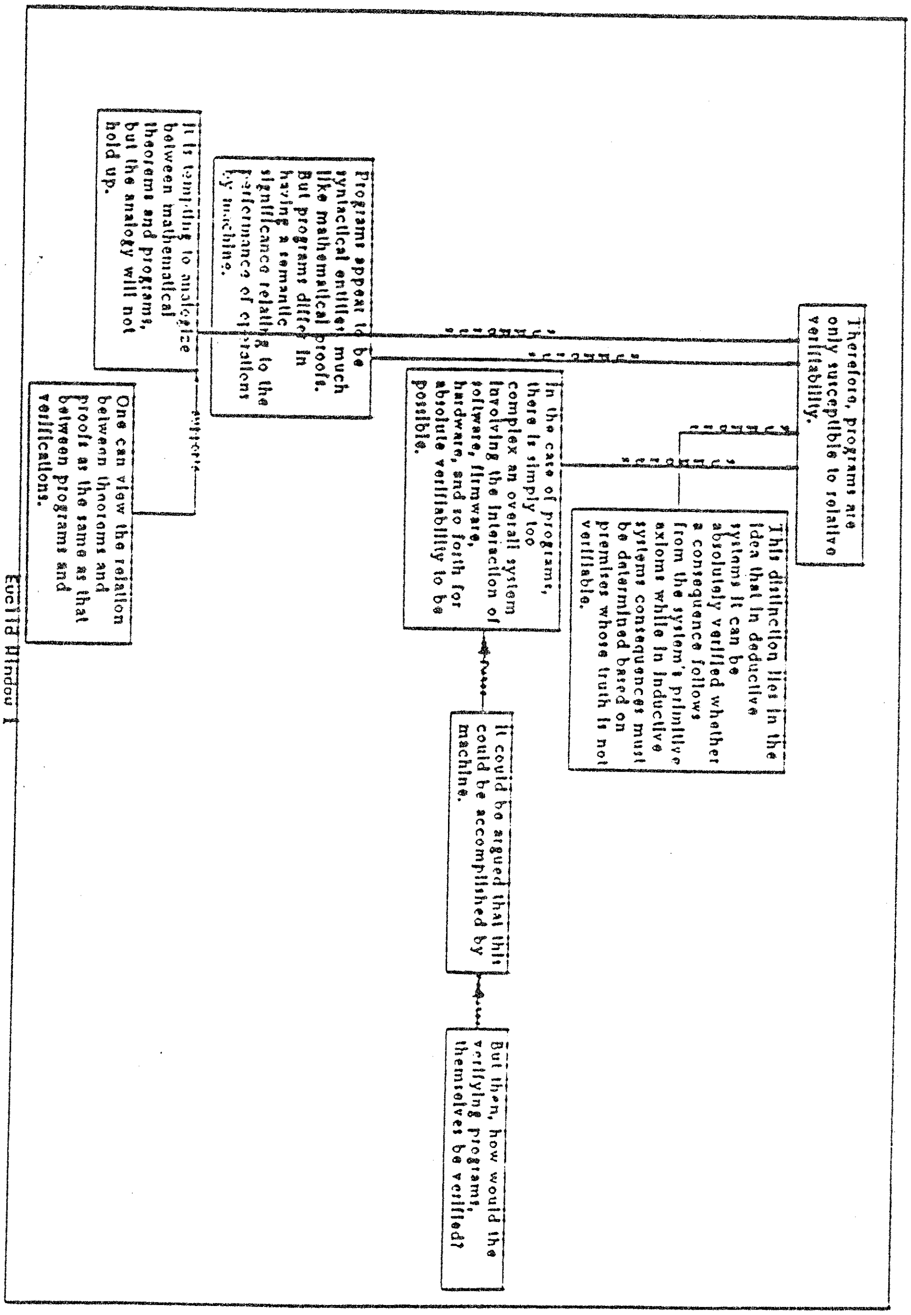


Figure 10-4. EUCLID (v4) Representation of Fetzer Article

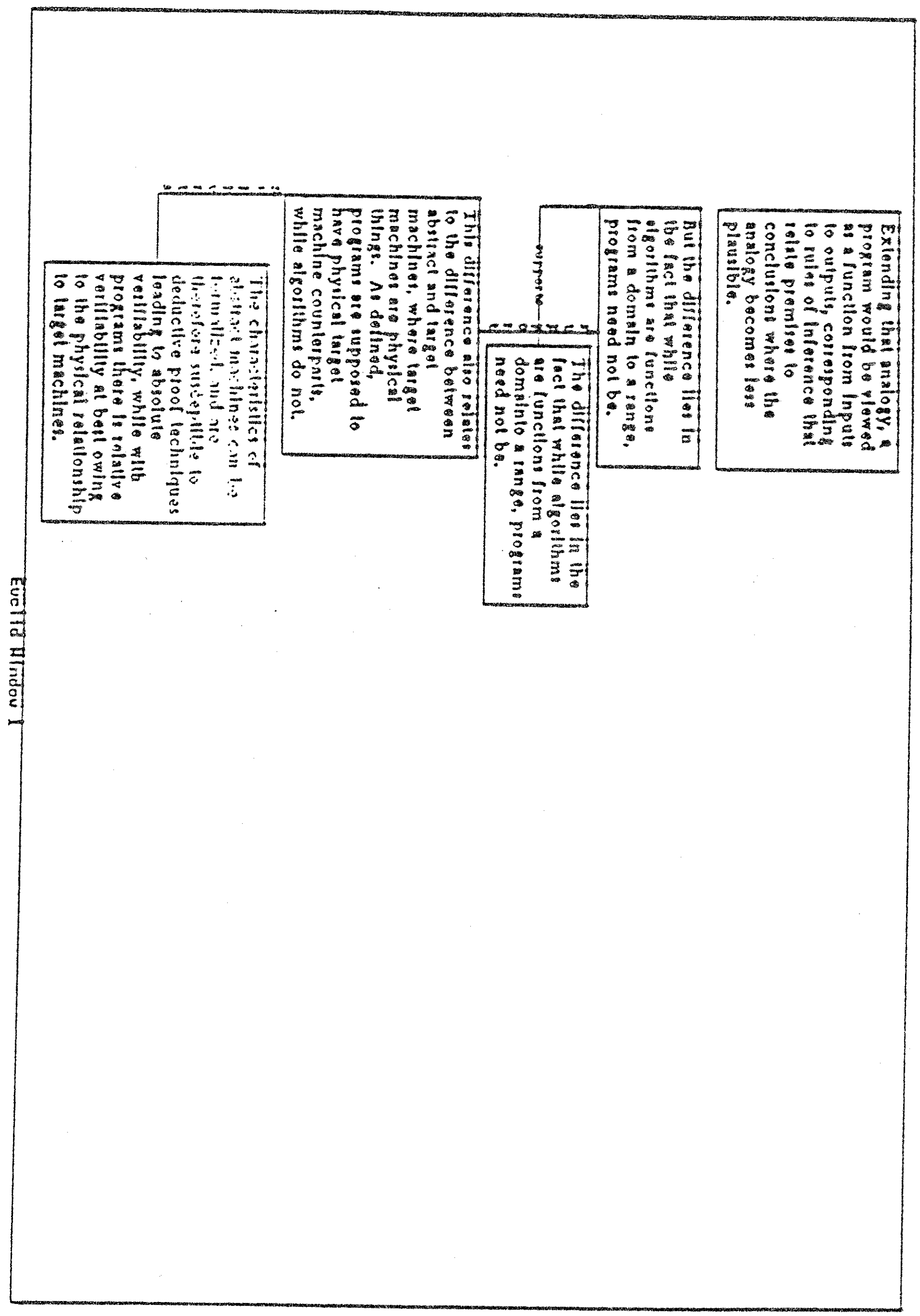


Figure 10-5. EUCLID (v4) Representation of Fetzer Article

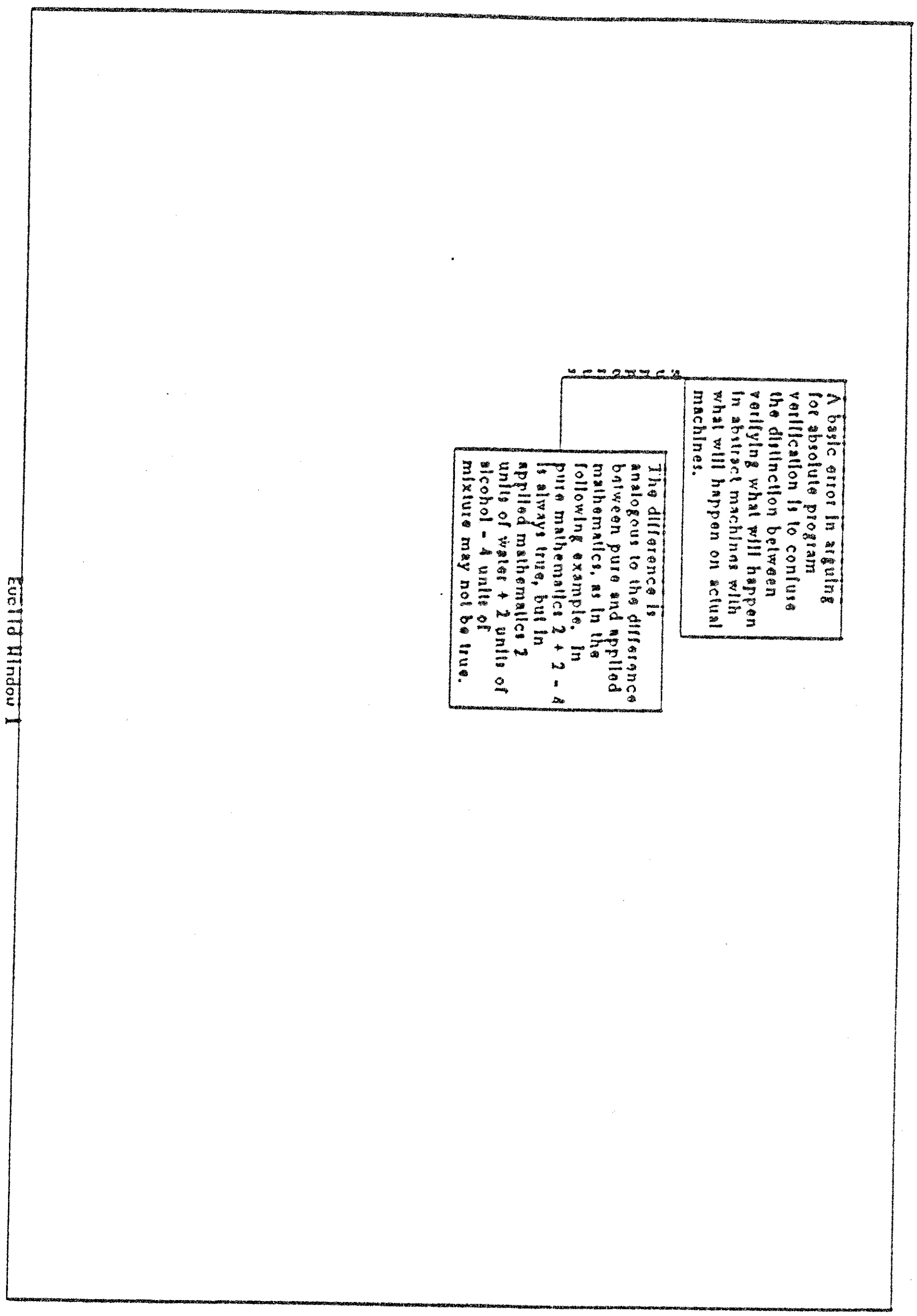


Figure 10-6. EUCLID (v4) Representation of Fetzer Article

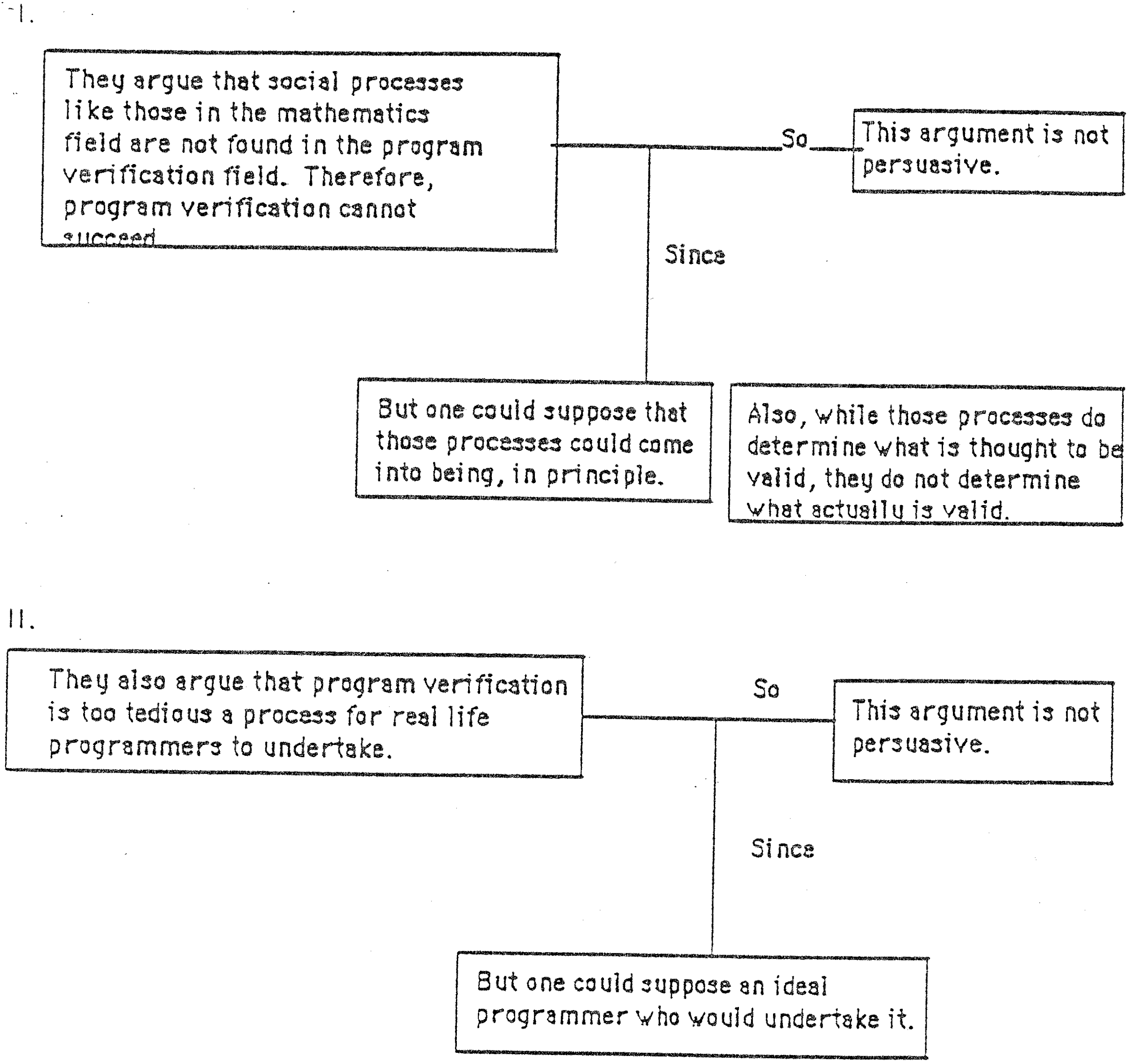


Figure 11-1. Toulmin Representation of Fetzer Article.

III.

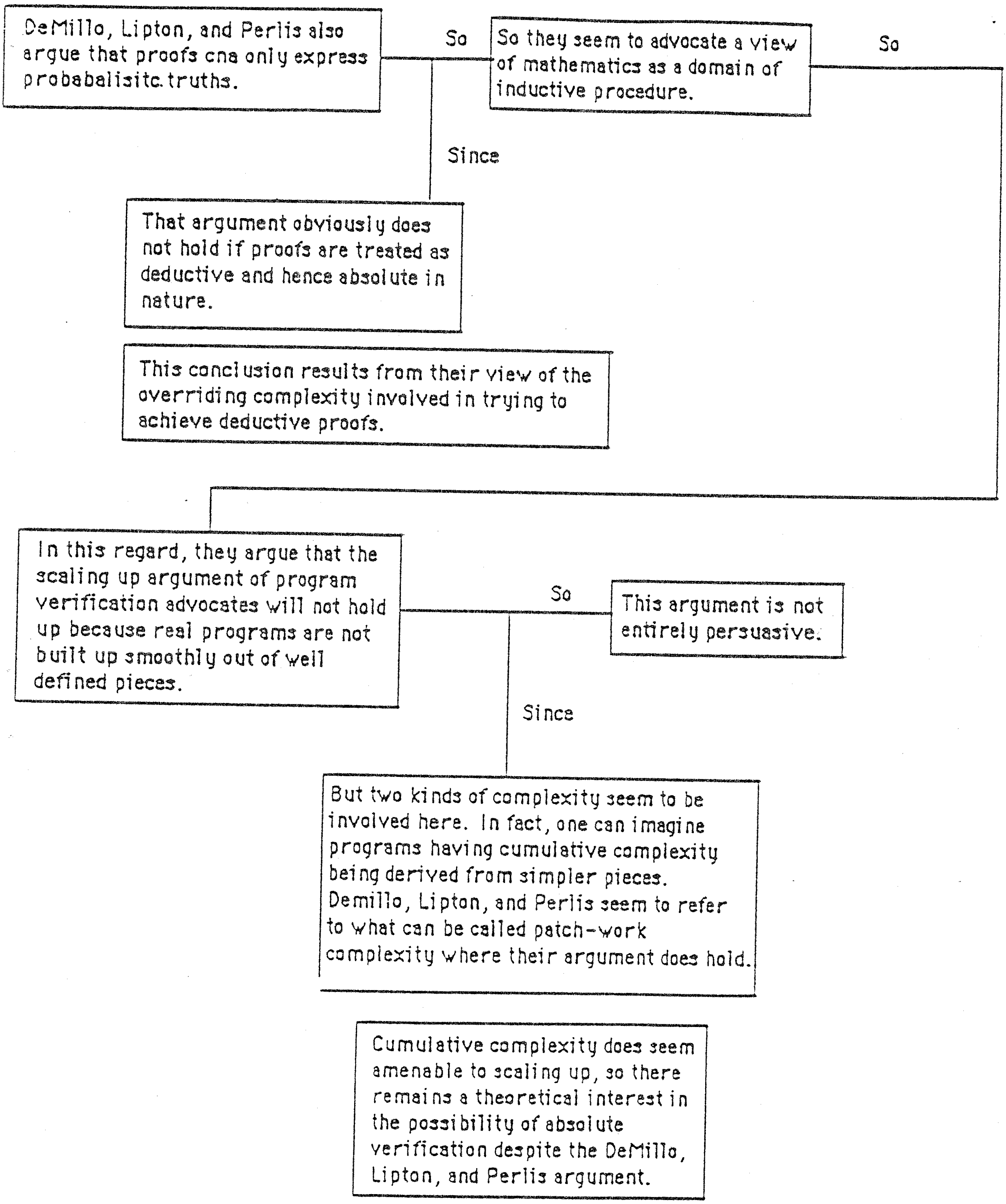


Figure 11-2. Toulmin Representation of Fetzer Article

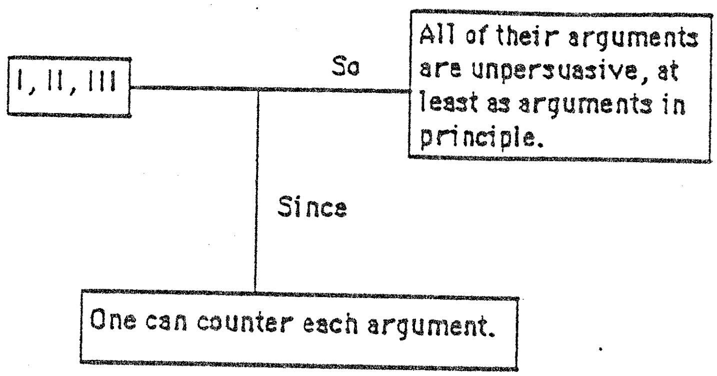


Figure 11-3. Toulmin Representation of Fetzer Article

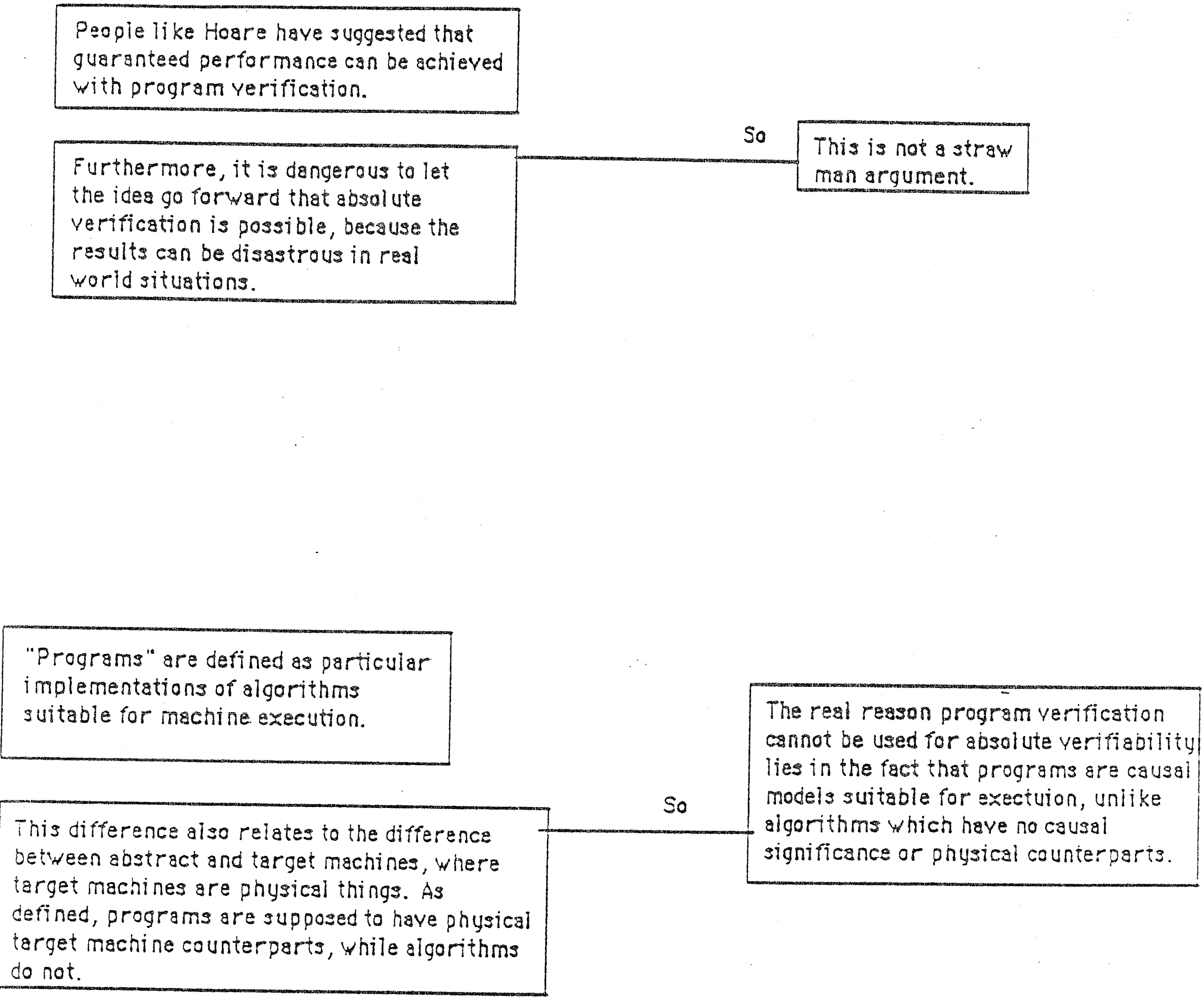


Figure 11-4. Toulmin Representation of Fetzer Article

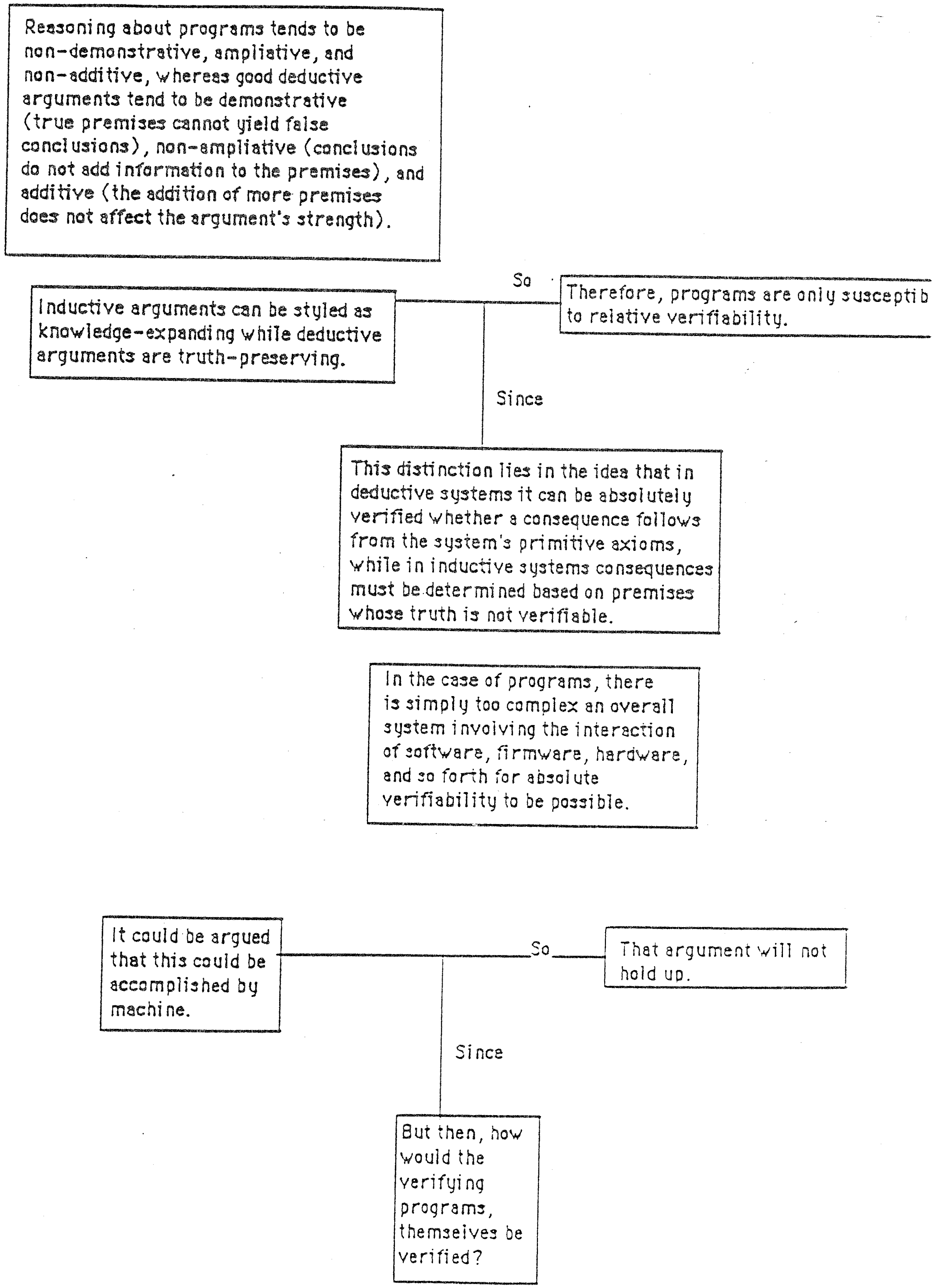
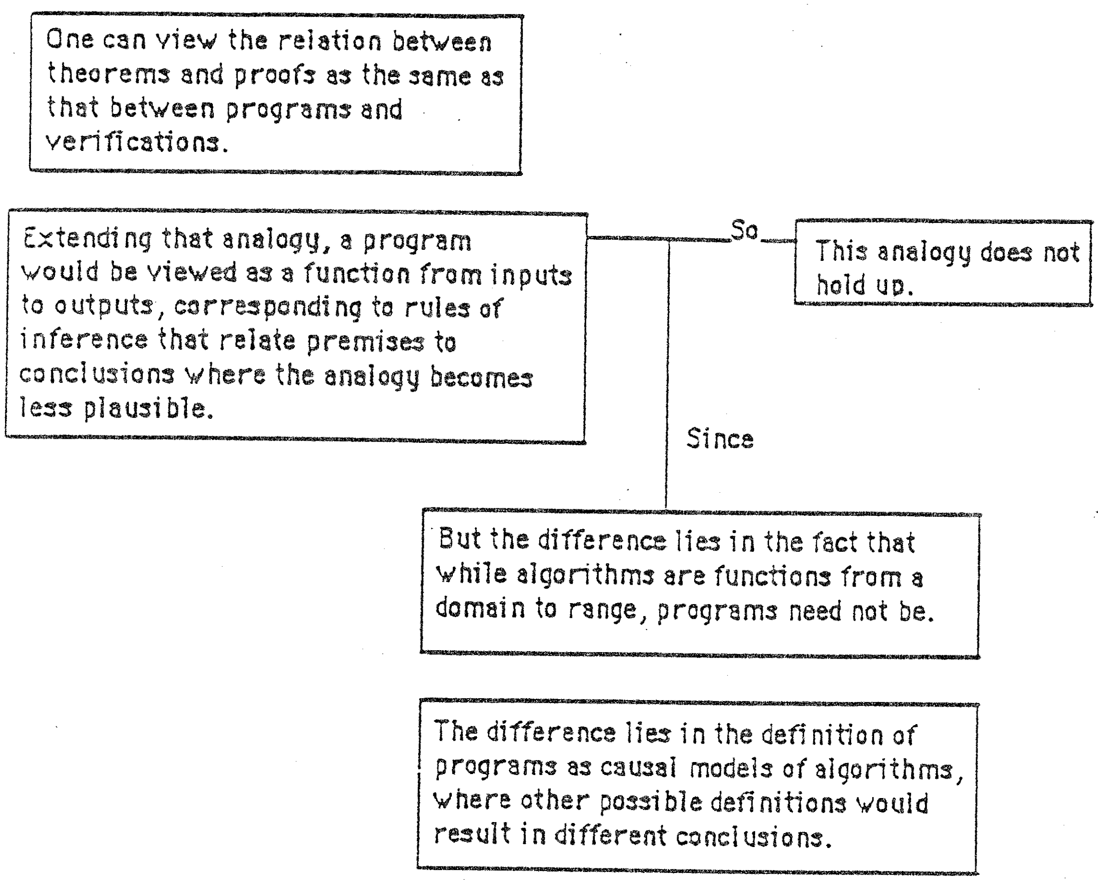


Figure 11-5. Toulmin Representation of Fetzer Article

I.



I.,

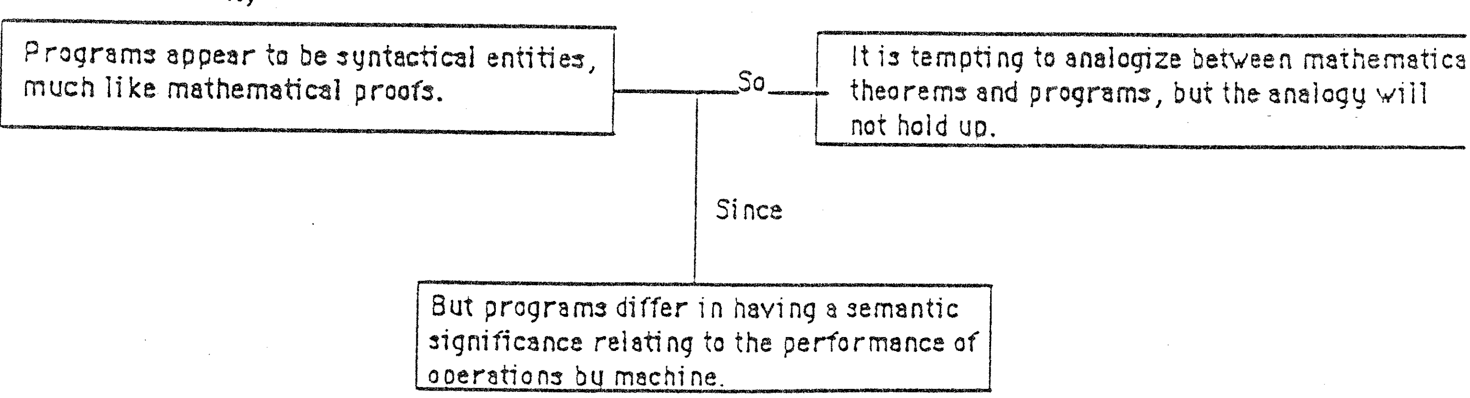


Figure 11-6. Toulmin Representation of Fetzer Article

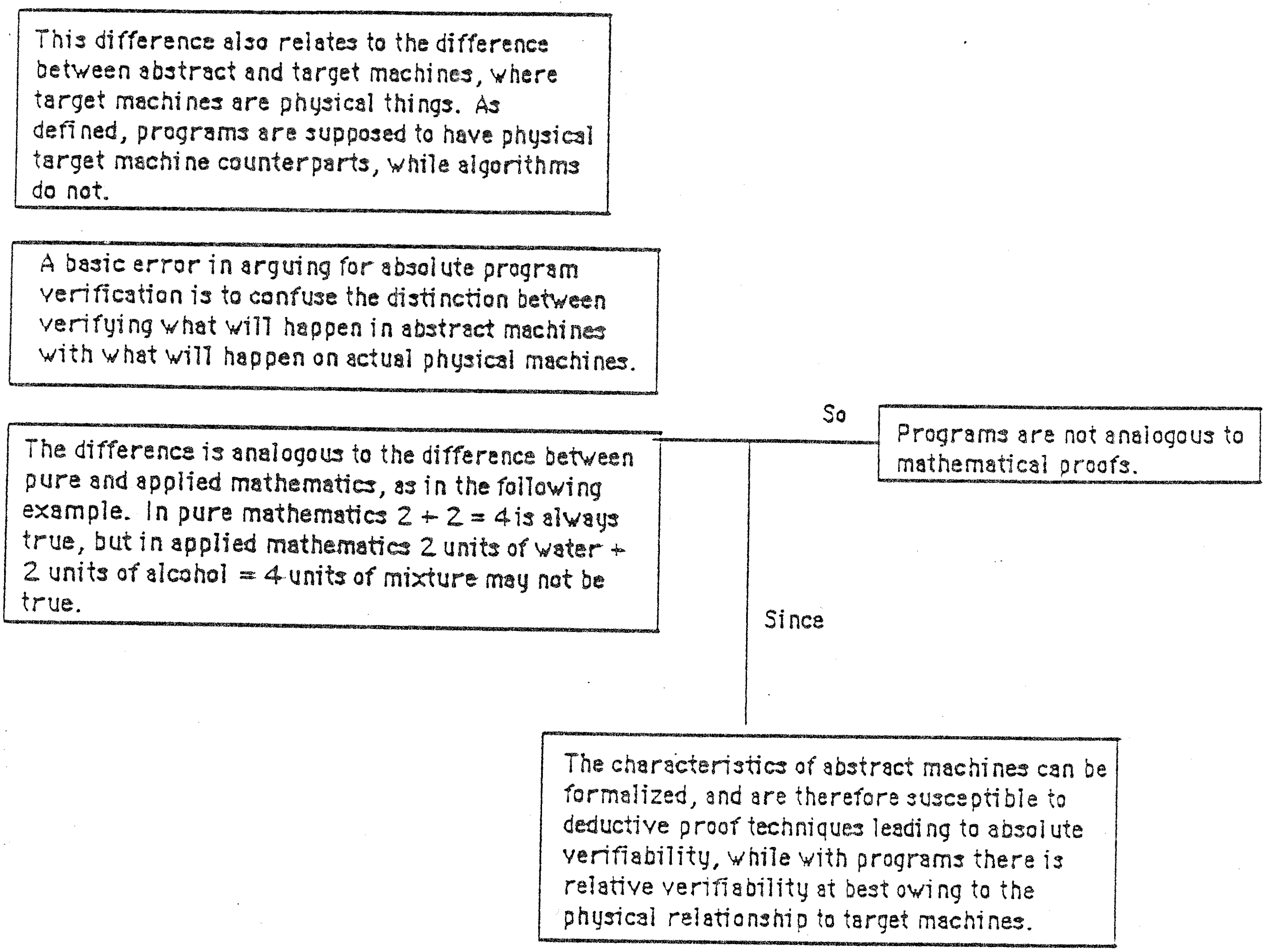


Figure 11-7. Toulmin Representation of Fetzer Article

```
/*This way you seem to miss the interconnectedness of the
arguments*/

absolute_var(X) :- like_pure_math(X).

like_pure_math(X) :- deductive_proof_applies(X).

deductive_proof_applies(X) :- demonstrative(X), non_ampliative(X),
additive(X).

inductive_proof_applies(X) :- non_demonstrative(X), ampliative(X),
non_additive(X).

relative_ver(X) :- ?  /*is relative verifiability the same as
induction?*/

absolute_ver(X) :- low_complexity(X).

low_complexity(X) :- provable_premises(X).


/*argument about the argument*/

argument_matters(X) :- believes_argument(X,Z),
affects_real_world(Y), person_in_field(Z).


/*arguments about DeMillo, Lipton, and Perlis arguments*/

absolute_ver(X) :- social_processes(X).
social_processes(X) :- actual_processes(X).
social_processes(X) :- could_be_processes(X).

absolute_ver(X) :- not_too_tedious(X).
not_too_tedious(X) :- actually_is_done(X).
not_too_tedious(X) :- could_be_done(X).

absolute_ver(X) :- not_probabalisitic(X).
not_probabalisitic(X) :- deductive_proof_applies(X).
/*Note: this suggests that Fetzer has really already answered this
argument about all proofs being probabalisitic*/
```

Figure 12-1. Prolog Representation of Fetzer Article