1

Representation of Knowledge in a Geometry Machine

E. W. Elcock

Department of Computer Science
University of Western Ontario

PART 1

In their book *Mathematics and Logic* Kac and Ulam (1971) comment:

> "The point of view as it has evolved through centuries is that one need not know what things *are* as long as one knows what *statements* about them one is *allowed* to make. Hilbert's famous *Grundlagen der Geometrie* begins with the sentence: 'Let there be three kinds of objects; the objects of the first kind shall be called "points", those of the second kind "lines", and those of third "planes"'. That is all, except that there follows a list of initial statements (axioms) that involve the words "point", "line" and "plane", and from which other statements involving those undefined words can now be deduced by logic alone. This permits geometry to be taught to a blind man and even to a computer!"

Leaving aside the attitude implicit in Kac & Ulam's use of the word 'even' in the phrase 'even to a computer', it has become clear that programs to prove theorems in first order axiomatic theories such as geometry, working in this 'blind' way, are unlikely to be successful. In parentheses, one might remark that mathematicians, however they express their proofs, usually do not construct them by working entirely within the formal syntactic system (i.e. blind).

What does it mean *not* to be blind? In the case of geometry, one of the ways would be to use a diagram in which 'points' and 'lines' referred to in the premises of a theorem to be proved are made concrete in a diagram, and predicates and functions of the theorem such as collinear, intersection, etc. are given their usual geometric interpretation and can be evaluated by procedures operating on the diagram. The actual points and lines made concrete in the diagram should, of course, be chosen so that the premises of the theorem to be proved are true in the diagram. Thus for the following theorem:

Premises:

ΔABC
M is the mid-point of segment BC
BD is the perpendicular from B to AM
CE is the perpendicular from C to AM.

To prove:

segment BD = segment CE.

an appropriate diagram could be that of Figure 1(a).

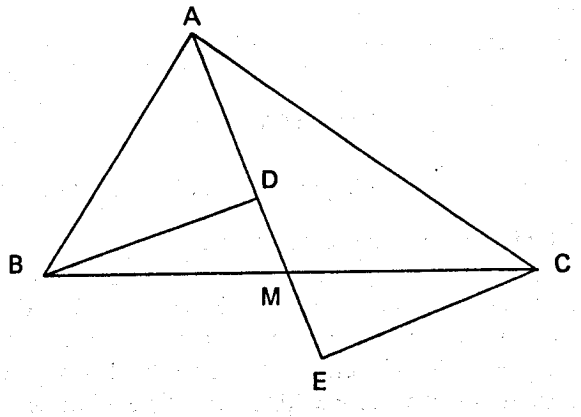

FIG. 1a

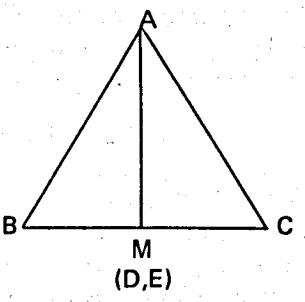

FIG. 1b

It is readily verified that the premises are true in the diagram. It is also worth remarking something that will be important later, namely that many things will be true in the diagrams which are *not* consequences of the premises of the theorem. Some of these, such as the fact that the length of segment AB is the particular multiple of the length of segment AC that is embodied in the diagram will usually be of no concern. Others might be. For example, if it were not for the premise

distinct (D,M,E),

a possible diagram would be that based on the isosceles triangle of Figure 1(b). Yet a "natural" illustrative diagram might still be taken to be that of Figure 1(a) when (potentially misleading) statements such as ΔBDM and ΔCEM are true in the diagram but are *not* implied by the (new) premises.

Despite an overabundance of things which are true in the diagram, for our purpose of proving theorems the diagram has a very important "inverse" property (to be stated more carefully later): anything which is *false* in the diagram is certainly *not* a consequence of the premises. Thus it is 'clear' in the diagram (in the sense of the processes underlying our perceptual comparison of angles) that ∠BAM ≠ ∠CAM. From what we have said, ∠BAM = ∠CAM cannot be a consequence of the premises (true in the diagram) of the theorem.

How is it that such a property of a diagram should be of great use in developing a proof of the associated theorem? Since this question motivates much of this paper, we will attempt an informal answer immediately. This we will do by suggesting the evolution and motivation of proof steps in the context of a simple example—the theorem stated earlier. So:

Premises:

ΔABC
M is the mid-point of segment BC
BD is the perpendicular from B to AM
CE is the perpendicular from C to AM
distinct (D,M,E)

Prove (G1):

segment BD = segment CE.

Informal proof:

As mentioned, we can draw a diagram (Figure 1(a)) to illustrate the theorem. We recall that there is a theorem of plane geometry which says that:

THEOREM 1: If two triangles are congruent, then their corresponding sides are equal.

We can clearly use this known theorem to prove that BD = CE if we can show that BD and CE are corresponding sides of two congruent triangles. The diagram suggests that we try:

Prove (G2):

ΔBDM = ΔCEM.

If we can prove G2, then Theorem 1 establishes our original goal G1 since BD

and CE are, indeed, corresponding sides of ΔBDM and ΔCEM. We shall see that the proof of G2 is straightforward but, before continuing the proof, let us pause and comment on the mechanisms which underly the apparent ease with which we, in fact, set up goal G2.

First, given that we have decided that we are going to use the tactic implicit in Theorem 1 and so attempt G2, how do we know that we can even assert ΔBDM and ΔCEM? We can say that it is 'obvious' (or can be 'assumed') from the diagram. More precisely, B, D and M are perceptually distinct and not collinear (the necessary and sufficient conditions for the assertion ΔBDM) in the diagram. The same is true of C, E and M.

Second, granted that we are going to choose a Δ with BM as a constituent side and a Δ with CM as a constituent side and try to prove them congruent, why did we choose the particular triangles ΔBDM and ΔCEM? Instead of G2 we could have set us any of the goals

$$\begin{aligned} &G2' \quad \Delta BDM \equiv \Delta CEA \\ &G2'' \quad \Delta BDA \equiv \Delta CEM \\ &G2''' \quad \Delta BDA \equiv \Delta CEA. \end{aligned}$$

Why did we not choose one of these for deeper (formal) exploration? It is suggested that these subgoals are not proposed for formal examination because in our diagram we can 'see' that these subgoals are patently false. Thus in the case of subgoal

$$\Delta BDA \equiv \Delta CEA:$$

we can 'see' that the necessary condition $\angle ABD = \angle ACE$ is false where by the phrase 'see. . .is false' we again emphasize that we imply some evaluative procedure (computation) on the diagram. On the other hand ΔBDM and ΔCEM 'look' congruent where again we mean that evaluative checking procedures in the diagram succeed (e.g. $\angle DBM = \angle ECM$ where the equality is in the framework of the visual procedures).

Finally, why choose a tactic based on Theorem 1 rather than some other? For example, with the particular initiating goal of proving two segments equal, we might have brought to bear tactics based on:

Theorem 1′: If ΔXYZ is such that its base angles $\angle XYZ$ and $\angle XZY$ are equal, then the sides XY and XZ opposite these angles are equal. (Tactic: to prove two segments equal, prove they are slant sides of a triangle whose base angles are equal), or

Theorem 1″: If segment XY = segment UV and segment RS = segment UV then segment XY = segment RS. (Transitivity of segment equality). (Tactic: to prove two segments equal, find a third segment which is equal to the original segments).

Again, the suggestion is that although such tactics might be tentatively con-

sidered as candidates for formal exploration, they are rejected on the basis of evaluative procedures in the diagram. Thus there is no triangle with equal base angles in the diagram, nor in the diagram is there a segment which is distinct from BD and CE and which appears to be equal to BD and CE.

This digression from our proof is motivated, as was mentioned in the introduction to the theorem to be proved, to show the important and many facetted role played by the diagram in a process of constructing a geometrical proof. For our expository purposes, the digressions are certainly more important than the emergent detailed proof, and soon we will want to consider both the formalisation and mechanisation of this role played by the diagram. Before doing this, however, let us complete the sketch of our proof: there are more insights still to be gained.

Our current goal is:

Prove (G2):

$\Delta BDM \equiv \Delta CEM.$

We now recall:

Theorem 2: If Δ's ΔXYZ and ΔRST are such that segment XZ = segment RT and $\angle XYZ = \angle RST$ and $\angle XZY = \angle RTS$ then the Δ's are congruent (i.e. the other pairs of corresponding sides and the other corresponding angles are each equal). (Tactic: if the goal is to prove that Δ's ΔXYZ and ΔRST are congruent, then try to prove the three goals

$$\left.\begin{array}{l}\text{segment } XZ = \text{segment } RT \\ \angle XYZ = \angle RST \\ \angle XZY = \angle RTS\end{array}\right)$$

In the context of our goal

$$\Delta BDM \equiv \Delta CEM$$

considered in isolation (i.e. forgetting for the moment its motivational history), there are a number of instantiations of the general tactic. By an "instantiation" we refer to the process by which, in applying a theorem or tactic we have to say which concrete points in the diagram we are going to associate with (substitute for) the 'general' points of the theorem or tactic. Our earlier remark, concerned with choosing candidate triangle pairs for congruence, to the effect that ". . . on the other hand BDM and CEM 'look' congruent . . .", implies that our computational procedures on the diagram reject unsuitable associations such as:

$$X/B,\ Y/M,\ Z/D,\ R/E,\ S/C,\ T/M$$

which would lead to an attempt to prove the subgoal

$$\angle BMD = \angle ECM \text{ (instantiation of } \angle XYZ = \angle RST)$$

which is clearly false in the diagram. This is just another example of the kind of use of the diagram already explained. Rather different is the situation that arises if we try the association

$$X/B,\ Y/M,\ Z/D,\ R/C,\ S/M,\ T/E$$

which would lead to an attempt to prove the three subgoals

$$\text{segment } BD = \text{segment } CE$$
$$\angle BMD = \angle CME$$
$$\angle BDM = \angle CEM$$

none of which are obviously false in the diagram. In fact, the last two, of course, can be proven true ($\angle BMD = \angle CME$ since vertically opposite angles are equal, and $\angle BDM = \angle CEM$ since all right angles are equal), but the first subgoal is the original theorem we set out to prove! If we, or our geometry program, do not recognise this, we are in danger of repeating the proof path to this point over and over again indefinitely!

It is clear that our proof style leads to a proof structure which is a hierarchy (tree) of subgoals as in Figure 2. Each node represents a subgoal which is proved if its descendent subgoals can be proved. We must monitor that no node (subgoal) is identical to one of its ancestor nodes.

On the assumption that we avoid such pitfalls, let us briefly complete our proof. We are trying to prove G2:

$$\Delta BDM \equiv \Delta CEM$$

this in turn being motivated by G1:

$$\text{segment } BD = \text{segment } CE.$$

which together with other diagrammatic evidence suggests the appropriate instantiation of the tactic associated with theorem 2 is:

$$X/B,\ Y/D,\ Z/M,\ R/C,\ S/E,\ T/M$$

when the three subgoals to be proved to establish G2 are:

$$\text{segment } BM = \text{segment } CM$$
$$\angle BDM = \angle CEM$$
$$\angle BMD = \angle CME.$$

The first of these three subgoals is a premise of our original subgoal. The second can be proved making use of the tactic: 'if you want to prove two angles equal prove they are both right angles', this last also being given in the premises of the original goal. The third subgoal can be proved making use of the tactic: 'if you want to prove two angles are equal, then prove that they are vertically opposite

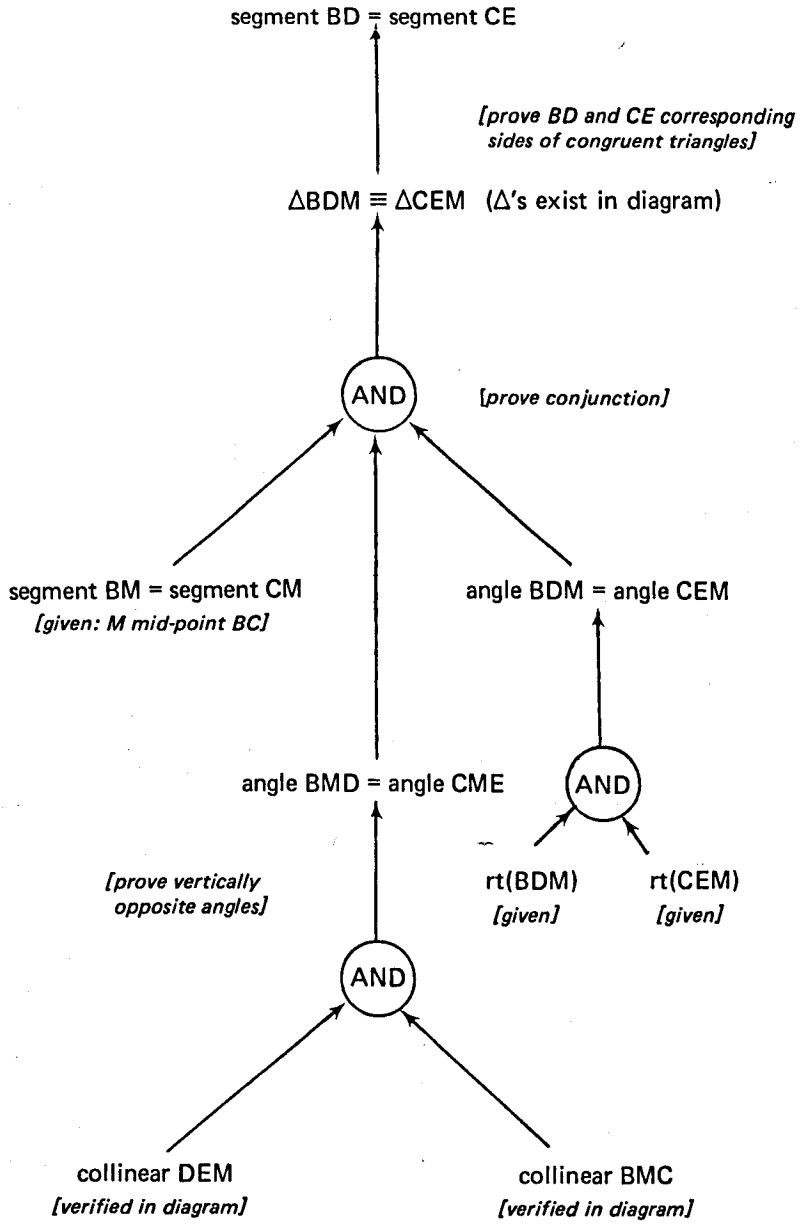

FIG. 2

angles'. This last involves descendent subgoals to establish the collinearity of D, M and E and the collinearity of B, M and C. As in some earlier examples, these subgoals can be "established" by procedures whose domain is the diagram. The final proof tree is shown in Figure 2.

Summarising: in this introduction we have attempted to both illustrate a

proof style and indicate the role of a diagram in facilitating proof discovery within that style.

The proof style essentially uses just one kind of tactic of the general form: 'if you want to prove B and you know a theorem 'if A_1 and A_2 and .... and A_n are true, then B is true' then try independently to prove A_1, A_2 ... and A_n". This proof style has been given the descriptive name 'backward chaining': as already seen, it can be illlustrated by a proof tree which is complete when all terminal nodes are 'givens' (or validated directly by procedures acting on the diagram)–i.e. when we have managed to chain backward from the theorem to be proved to these givens or things 'obviously true in the diagram'.

A *proof tree*, of which an example is shown in Figure 2 does not, of course, illustrate the full process of *proof search*. As we have attempted to indicate in the informal sketch above, the process of backward chaining might set up a subgoal for which a number of tactics might be applicable. Each of these, in turn, gives rise to a subtree in the search tree. Many of these a-priori applicable tactics might turn out, when examined, to be inappropriate. However, the discovery of their inappropriateness might involve elaboration of the subtree to some depth. The growth of the proof search tree is potentially explosive and it is vitally important that its growth be controlled and, in particular, subtrees which are going to fail (more precisely, cannot be part of a proof tree), should be detected and their exploration abandoned as soon as possible.

In the sketch above we have illustrated the role of the diagram as a factor in this control of the generation of (irrelevant) subtrees by rejection of proposed subgoals (root nodes of potentially large subtrees) which can be shown to be false in the diagram by computational procedures (as opposed to formal proof in the axiomatic system) over (the points of) the diagram.

This is not the only control mechanism which might be operative in the search process. For example, given a subgoal and given a set of potentially applicable strategies, it might be possible to order the alternative strategies according to some likelihood criterion perhaps based on some context in which the subgoal is embedded. This last kind of control mechanism has been less well explored and is less well understood. It will not be of concern in this paper.

In part 2 below we will examine briefly some work on the implementation of a geometry machine which follows the paradigm of part 1. As part of this, some of the points covered in part 1 will be made precise in a precise context. Weaknesses as well as strengths of current work in the paradigm will be considered and an attempt made to indicate how a 'seeing' machine geometer might develop.

PART 2

In two fascinating papers written fifteen years ago (Gelernter, 1959 and Gelernter, Hansen and Loveland, 1960), the authors wrote about what they called a geometry theorem proving machine. The 1960 paper begins with the (stirring) words:

> "In early spring, 1959, an IBM 704 computer, with the assistance of a program comprising some 20,000 individual instructions, proved its first theorem in elementary Euclidean plane geometry (Gelernter, 1959b). Since that time, the geometry-theorem proving machine (a particular state configuration of the IBM 704 specified by the aforementioned machine code) has found solutions to a large number of problems taken from high school textbooks and final examinations in plane geometry. Some of these problems would be considered quite difficult by the average high school student. In fact, it is doubtful whether any but the brightest students could have produced a solution for any of the latter group when granted the same amount of prior "training" afforded the geometry machine (i.e., the same vocabulary of geometric concepts and the same stock of previously proved theorems)."

The papers, whilst leaving much to be inferred by the reader, make clear that the 'geometry theorem proving machine' is based on the powerful paradigm described informally in part 1 of this paper. However, until very recently little attempt was made to build on this work. The ensuing years have seen an emphasis on the mechanization of complete uniform proof procedures for first order predicate calculus. It has become increasingly clear that this work by itself is unlikely to take one into the domain of interesting theorems. It now seems generally accepted that proof procedures must be capable of exploiting the specificity of the problem domain, be it geometry, number theory, whatever. Reiter (Reiter, 1972) discusses possible alternative ways of exploiting specificity and gives reasons for focussing attention on a particular extension of the paradigm of part 1 of this paper. We will try to indicate why later. First let us return to the first implementation by Gelernter and his co-workers. As mentioned earlier, their papers left much of their method to be inferred. In what follows and indeed in part 1 we have made use of Gilmore's careful and detailed analysis (Gilmore, 1970) to which readers are referred for a more formal treatment.

The geometry machine uses a given set of universally quantified statements (axioms) of the general form:

for all $x_1, x_2, \ldots x_n$:
if S_1 and S_2 and ... and S_n *then* S;

where $x_1, x_2 \ldots x_n$ are variables which are to be instantiated by (replaced consistently by) names of points. $S_1, S_2 \ldots S_n$ and S are applications of simple predicates of geometry such as:

triangle $(x_1 x_2 x_3)$;
collinear $(\{x_1 x_2 x_3\})$; between $(x_1 \{x_2 x_3\})$;
equal (segment $(x_1 x_2)$, segment $(x_3 x_4)$);
equal (angle $(x_1 x_2 x_3)$, angle $(x_4 x_5 x_6)$);

congruent (triangle $(x_1x_2x_3)$, triangle $(x_4x_5x_6)$);
mid-point $(x_1$, segment $(x_2x_3))$

etc. or their negations.

For example (leaving the statement of universal quantification over the variables as understood):

if between $(x_2\{x_1x_3\})$ and between $(x_2\{x_4x_5\})$
then equal (angle $(x_4x_2x_3)$, angle $(x_1x_2x_5)$);
if distinct $(\{x_1x_2x_3\})$ and not (collinear $(\{x_1x_2x_3\})$);
then triangle $(x_1x_2x_3)$.

Apart from simple substitution, the only other mechanism for deriving theorems is the simple (inference) rule:

Given the axioms

if S_{11} and S_{12} and . . .and S_{1n} *then* S_1;
if S_{21} and S_{22} and . . .and S_{2m} *then* S_2;
if S_1 and S_2 *then* S;

we can conclude

if S_{11} and . . . and S_{1n} and S_{21} and . . . S_{2m} *then* S.

The theorems of the system have precisely the same form as the axioms: they are all (universally quantified) implication sentences.

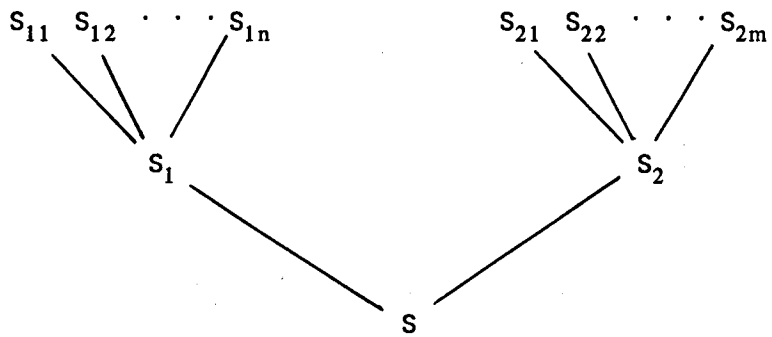

FIG. 3

The inference rule can be expressed by the tree of Figure 3. From this stems the notion of a proof of a theorem

if S_1 and S_2 . . . and S_k *then* S

as a tree in which nodes are labelled with sentences and:

(i) each node is labelled with a simple sentence in the set $S_1 \ldots S_k$, or

(ii) is connected to a set of descendant nodes labelled with the antecedent sentences of some axiom with consequent the label S of the parent node.

An example of such a proof tree has been given in Figure 2.

The theorem proving algorithm of the geometry machine is, as already intimated, based on 'backward chaining' which can now be expressed more formally as a process for searching for a proof tree by starting from a root node (labelled with the consequent of the theorem to be proved) and exploring the *set* of trees which can be generated at any stage by the inference rule and the set of applicable axioms (those with consequent sentences labelling a terminal node of the tree).

Gilmore (Gilmore, 1970) shows that this process is both a theorem proving algorithm and a decision process. By a decision process is meant that if an implication sentence is (is not) a theorem in the particular system defined by the particular set of implication sentences taken as axioms, then the process will terminate successfully (unsuccessfully). Being a theorem proving algorithm implies that successful termination also returns the proof tree.

The particular axiom set used in the Geometry Machine is not important here (other than to recognise, of course, that it determines the particular fragment of geometry captured by the machine) and we shall focus our attention on the formal counterpart in the theorem proving algorithm of the Geometry Machine of the paradigm use of a diagram in the proof style of part 1.

In order to do this with some precision, we need to explain the notion of a model. For this we return to the opening quotation from Kac & Ulam. Plane geometry is a first order axiomatic theory. It deals with undefined objects called points and lines and the system is defined by a (small) set of axioms stating relations which hold over the objects of the system together with a method of inference (that of first order predicate calculus) which allows new relations to be deduced–the new relations being called theorems. The proof of a theorem in the system consists in exposing its generating chain of inferences: so-called syntactic proof.

Alternatively it is possible to set up a mechanism for assigning a meaning to a well-formed sentence in the system. This is done by choosing some definite domain D of objects and mapping the objects, function and predicates of the well-formed sentence in the system onto objects in D and functions and relations over D respectively. Such a mapping is called an interpretation or model of the well-formed sentence and the sentence will have a truth value in this model. The notion of a theorem in the axiomatic system now becomes that of a well-formed sentence which is true in all models: a so-called semantic notion of proof. It turns out that the syntactic and semantic notions of proof are equivalent: i.e. categorize the same set of sentences. The second notion, however, has an interesting property. Since a sentence is a theorem if and only if it is true in *all* models, disproof can simply consist in exhibiting a *single* model in which the

sentence is false (the method of counter example). It is this last which lies at the heart of the use of diagrams in the Geometry Machine.

The models we shall use in the Geometry Machine will be ones in which D is the domain of ordered real number pairs. A named point in a theorem to be proved will be mapped into a particular pair of D (conventionally: its coordinates in the Cartesian plane). A line determined by two points P_1P_2 is mapped into the set of pairs (x,y) defined by the *algebraic* relation

$$y\text{-}y_1/x\text{-}x_1 = y_2\text{-}y_1/x_2\text{-}x_1.$$

Other geometrical functions and predicates are mapped into their usual algebraic interpretations in the Cartesian plane. We can now show that a sentence is not true by simply showing that it has a denotation in the Cartesian plane which is false.

How does this help us? First, let us clarify the relationship between a theorem and a diagram. A theorem refers to a set of named points and to certain relationships holding over them. The function of a diagram is to explicate in some model the denotations of these particular relations out of the total set of relations holding over the set of points.

In the particular case of an implication sentence

"if S_1 and S_2 . . . and S_n then S"

for the Geometry Machine, the diagram would consist of a set of number pairs, one for each point named in the implication sentence, and chosen so that the premises S_1 to S_n of the sentence are true in the diagram. The general properties of the (Cartesian) model guarantee that the axioms of the Geometry Machine are true in the model. It follows that anyting false in the model is not derivable from the axioms and the premises of the implication sentence to be proved.

Again, how does this help us? It gives us the possibility (illustrated informally in part 1) of mediating the search for a syntactic proof by semantic notions. For example, it might be desirable to establish at a particular point in proof search whether a relation such as "mid-point $(P, \{P_1P_2\})$" holds or not. Computationally it might be difficult or just lengthy to decide this by syntactic methods. On the other hand, if (x,y), (x_1y_1) and (x_2,x_2) are the number pairs in the diagram denoted by P, P_1 and P_2 respectively, then a simple arithmetic evaluation of the expressions

$$2x\text{-}x_1\text{-}x_2$$
$$2y\text{-}y_1\text{-}y_2$$

resulting in a value for either which is sensibly different from zero makes it obvious that the relation is false in the diagram and, therefore, not derivable syntactically. On the other hand, if both these expressions are close to zero (machine arithmetic with finite precision!), then although this cannot be taken as establishing the relation, it might be taken as an indication that the effort of

examining the truth of the relation by syntactic methods was worthwhile.

Examples of arithmetic evaluation in the diagram abound: they parallel the 'perceptual computations' on the ink-mark drawings which were used as diagrams in part 1. We clearly have considerable potential here for a rich interplay of syntax and semantics in proof search.

Not all these possibilities are exploited in the Geometry Machine: we will finish this part by a fairly abstract characterization of the particular use the Geometry Machine, as described so far, makes of the diagram. In part 3 below we shall briefly explore other possibilities.

The Geometry Machine is given:

(i) an implication sentence
if S_1 and S_2 and . . . and S_n then S
to prove;

(ii) a denotation of each named point in $S_1 \ldots S_n$ as a number pair, the number pairs being carefully selected to make each S_i $1 \leqslant i \leqslant n$ true as discussed above;

(iii) the set of mentioned line segments in S_i $1 \leqslant i \leqslant n$ and S where constructs, such as triangle (ABC), formed from line segments, are treated as a mention of the implied line segments.

In searching for a proof tree, the Geometry Machine will only use simple sentences that are true in the diagram (where "truth in the diagram" has the meaning already discussed).

Since the Geometry Machine has only a finite number of functions and predicates, and since the implication sentence to be proved refers to only a finite number of points, and since the inference rules of the Geometry Machine do not allow new named points to be generated: there are just a finite number of simple sentences which are true in the diagram and these could be computed once and for all. If we call this set of simple sentences D_T then we can assert that if the implication sentence

if S_1 and S_2 and . . . and S_n *then* S

is a theorem, then S must be a member of D_T. The diagram can be regarded simply as a convenient device for computing the set D_T. The set D_T in general will be vastly smaller (for the reasons explored above) than the total set of simple sentences that can occur in the set of *all* (syntactically allowable) substitution instances of axioms using the point names of $S_1 \ldots S$, D_I say.

In searching for a proof tree, we need only use sentences from D_T to label nodes. The vast difference in size between D_T and D_I is another way of characterizing the exploitation of the diagram in the Geometry Machine theorem proving algorithm.

Another use of the diagram by the Geometry Machine parallels important

operations mentioned in part 1 which informally made use of certain kinds of relations among points which could be said to be "obvious from the diagram" and not requiring a (sometimes irritatingly tedious) syntactic proof. Terminal nodes in the proof tree are allowed to be labelled by certain sentences in D_T which are not premises but which would stem typically from axioms of order or from axioms necessary because "although the geometry is a point geometry, some of the simple sentences of the axioms are expressed in terms of angles, line segments. . ." (Gilmore, 1970). For example, if the diagram contains the equivalent of Figure 4 then a terminal node of a putative proof tree might be labelled:

angle (ABD) = angle (EBC)
[vertically opposite angles in diagram].

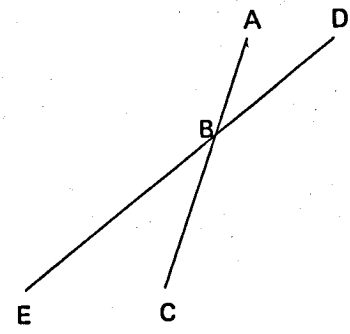

FIG. 4

The justifying axiom (not made overt in the system) would be: "*if* A B C are collinear in that order and D B E are collinear in that order, *then* angle (ABD) = angle (EBC)". The organization of the computational procedures that embody such axioms and the method by which they are invoked in the actual Geometry Machine is not clearly stated. Nevertheless, this judicious blurring of the notion of syntactic proof and truth in an appropriate model for certain kinds of sentences is one that lies at the heart of doing mathematics, and which Reiter (Reiter, 1972) has shown must be a component of a good theorem-prover.

Finally, the Geometry Machine also uses the set of given line segments to order the substitution instances considered. If an axiom mentions a line segment, then only line segments from the initially given set are used as substitution instances until all possible such substitutions are exhausted without proof. If this happens, then arbitrary point pairs in the given list for the diagram are used to define new line segments for substitution. These new line segments are added to the diagram list and could be regarded as a form of "weak" construction. The reason for ordering is obvious, but not overly convincing as a motivating mechanism for controlling search.

It should be emphasised that these weak constructions do not introduce any

new named points, only line segments between existing named points. The axioms of the Geometry Machine are such that all named points in the conclusion of an implication sentence are mentioned in at least one of its antecedents: no mechanism exists in the Geometry Machine for the introduction of new points. However, without such a mechanism the Machine is cut off from the more interesting class of theorems. In an attempt to remedy this deficiency, the Geometry Machine was, indeed, extended by the addition of a single axiom involving existential quantification. The axiom asserts that "if a line segment xy is not parallel to a line segment zw then there exists a point of intersection u (xy∩zw)": i.e.

for all xyzw

if not (parallel (xy,zw))

then there exits u *such that*

collinear (x y u) and collinear (z w u).

However, this axiom is used quite differently to those introduced earlier. In particular, it is never used to label a node in a putative proof tree: rather it is invoked only when the current search space is exhausted. The axiom, if applicable, is then used to introduce a new named point into the diagram and the process of search for a proof tree started again. It is clear that the notion of proof is unaltered in the extended machine, but the theorem proving algorithm is now no longer a decision process as well since the existential axiom allows for non-terminating successive introduction of new points.

Since constructions produced by this new axiom are so ill-motivated, and since we will discuss constructions in a wider context in part 3 below, the point will not be elaborated here. Rather, we will let the undoubted merit of the seminal work on the Geometry Machine be judged from the described unextended machine.

PART 3

As mentioned earlier, it is only comparatively recently that interest in the Geometry Machine has revived.

One factor has been the development of very high-level goal oriented languages such as Planner (Hewitt, 1971) which are claimed to provide a powerful and natural formalism for structuring mathematical knowledge as programs. Certainly, language constructs in Planner are well-matched to the kind of tactical interpretation of axioms and backward chaining of the Geometry Machine. Indeed, a Geometry Machine in the style of the original was implemented very straightforwardly by Goldstein (Goldstein, 1973) using just such a language. The description of this implementation is detailed enough to indicate the total set of implication sentences used to achieve the level of performance indicated.

Goldstein also discussed the possibility of extending the Geometry Machine

to obtain a more motivated search for a proof and sketches how this might be coupled with "knowledge" for making constructions in the diagram. This last is not presented as a uniform procedure, but rather as if controlled by a set of heuristics associated with particular goals or strategies. This in a sense prejudges the needs of the set of infinite individual proofs. However, it is difficult to infer enough from the few remarks made to be quite sure what is intended. Since this point of constructions being a response to the evolving total state of a syntactic proof is a central issue in some recent careful and detailed work by Reiter (Reiter, 1972), we will base our discussion on his work.

This discussion will return to the informal style of part 1 and with the same intent: simply to give a feeling for the problems and possibilities of a major extension of the proof methods of the Geometry Machine. Readers interested in a fuller and inevitable formal discussion are referred to Reiter's paper in which the complete inference system is given and motivated by some nicely designed examples which illustrate its potential power.

As far as the author is aware, there is as yet no machine implementation based on these ideas and, indeed, it is clear that there are challenging problems to be solved in a design for an implementation.

The idea to be explored briefly below is a continuation of the theme of the desirability of a rich interplay between syntax and semantics and particularly that aspect concerned with the generation of constructions appropriate to the evolving syntactic proof. The idea will be explored in a very informal treatment of an example taken from Reiter's paper.

Suppose then we want to prove the following theorem (implication sentence):

> *If* ABCD is a trapezoid *and* BC is parallel to AD *and* the line joining the mid-point E of AC to the mid-point F of BD meets AB at M *then* MA = MB.

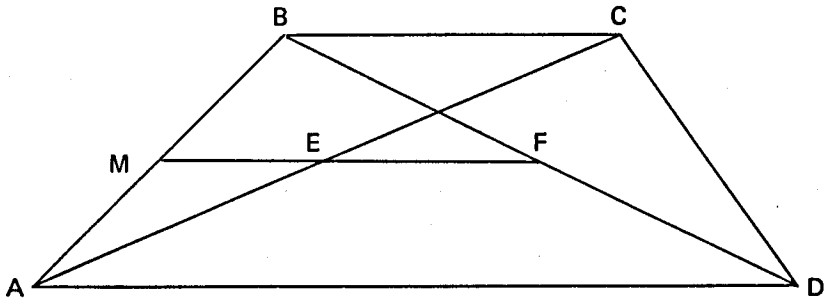

FIG. 5a

Figure 5(a) shows a drawing of the intial diagram (model). We notice that MF is parallel to AD in the model: i.e., say

$(y_D\text{-}y_A)(x_F\text{-}x_M) \cong (y_F\text{-}y_M)(x_D\text{-}x_A)$.

If we can, indeed, prove this parallelism syntactically, then an application of the axiom

if Δxyz and collinear (xuy) and collinear (xvz) and parallel (uv,yz) and xv = vz
then xu = uy

with the substitution instance ΔBAD with BF = FD given, and parallel (MF,AD), will allow BM = MA (our original goal) to be inferred.

The problem then is to prove parallel (MF,AD).

We are here touching on the extremely interesting topic of attending to (relevant) truths in the model. Here we must simply leave as an open question whether exploration of truth in the model should always be in response to goals in a developing syntactic proof or whether some prior exploration of truth in the model might suggest appropriate sets of syntactic possibilities and facilitate their ordering for detailed consideration.

To return to the problem of producing a syntactic proof of parallel (MF,AD). Computation in the model produces predicates parallel (ME,AD) and parallel (EF,AD) as equivalent to parallel (MF,AD): their equivalence can be justified syntactically by the axiom

A1: if parallel (xy,uw) and collinear (uvw)
then parallel (xy,uv).

This axiom can be interpreted as making a remark about the model which bears on the possibility of just this issue raised earlier of syntactically motivated constructions. The remark is that, if you want to prove some instantiation (in the model) of

parallel (xy,uv)

then check by a computation in the model (on, say, Cartesian point coordinates) that the instantiation is true in the model, when you can also be sure that a consistent instantiation of

parallel (xy,uw) and collinear (uvw)

can be found which will be true in the existing model if there is a named point on uv in the model, or true in the existing model extended by introducing a new named point anywhere on uv to serve as an appropriate denotation of w. This extension would be a syntactic proof motivated construction in the model producing a new model in which the semantic requirements for the use of the axiom are met.

If the axiom A1 is, indeed, applied in this spirit to the goal parallel (EF,AD), we have the instantiation x/E, y/F, u/A, v/D. There is no named point in the

model which can instantiate w. We know however that there does exist an extension of the model in which this instantiation can be made and so we proceed to the attempt to prove

parallel (EF,Aw) *and* collinear (AwD).

In the spirit of deferring this particular extension of the model (instantiation of w) we try first to prove

parallel (EF,Aw).

Assume the axiom:

A2: *if* Δxyz *and* collinear (xry) *and* collinear (xsz)
and xr=ry *and* xs=sz
then parallel (rs,yz).

An appropriate instantiation is r/E, s/F, y/A, z/w and again the use of the axiom can be justified semantically only if the instantiation can be completed so that

ΔxAw and collinear (xEA) and collinear (xFw)
and xE = EA and xF = Fw

is true in some extension of the model in which w is instantiated by a point on AD (this last constraint on extension making it unnecessary to return and check the other predicate of the conjunction above). Now the predicates

collinear (xEA) *and* xE = EA

give x a unique instantiation, C, in the model and it is also quite easy to conceive of appropriate computations (say over a "list-of-points-on" EA initiated by the collinear predicate etc.) which would find this instantiation. Now, given the instantiation x/C we have to satisfy

collinear (xFw) *and* Fx = Fw

i.e.

collinear (CFw) *and* CF = Fw

in the model. This time there is no instantiation of w in the model. The predicates, however, fix the denotation of w in an extended model uniquely. However, an extended model has to satisfy collinear (AwD). The instantiation of w in the extended model has, therefore, to satisfy

ΔCAw *and* collinear (CFw) *and* FC = Fw *and* collinear (AwD).

The point K (CF$\cap$AD) in Figure 5(b) is readily computed as the required unique instantiation of K which makes the conjunction of assertions true in the (extended) model.

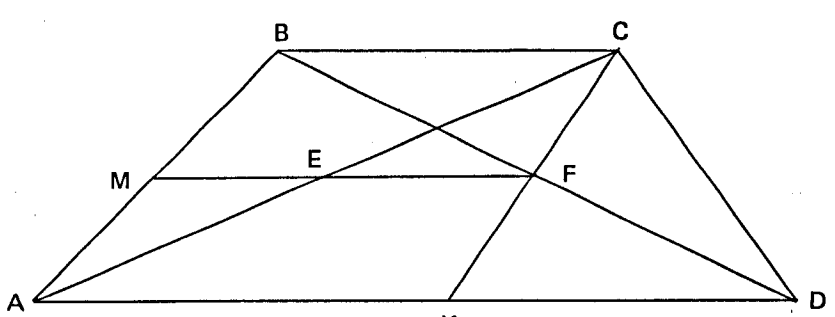

FIG. 5b

Taking K then to be the point named by CF∩AD, we continue the syntactic proof and try to establish

$$\Delta CAK \textit{ and } \text{collinear (CEA) } \textit{and } \text{collinear CFK}$$
$$\textit{and } \text{collinear (AKD) } \textit{and } EC = EA \textit{ and } FC = FK.$$

The first five literals of this conjunction are easily proved leaving the goal

$$FC = FK$$

still to be established. This can be done by proving the Δ's FKD and FCB congruent.

The example has been carried far enough for our expository purpose, that is, to illustrate the possibilities of interaction of syntactic and semantic methods in proof search, and to make it plausible that the gap between our informal presentation of an extended proof style and an extended machine is worthy of serious research and a continuance of an excellent tradition.

REFERENCES

Gelernter, H. (1959) Realization of a geometry theorem proving machine. *Proc. Int. Conf. Inform. Processing*, Paris.

Gelernter, H., Hansen, J.R., and Loveland, D.W. (1960) Empirical explorations of the geometry theorem machine. *Proc. Western Joint Computer Conf.*

Gilmore, P.C. (1970) An examination of the geometry machine. *Artificial Intelligence*, 1, 171-187.

Goldstein, I. (1973) Elementary geometry theorem proving. *A.I. memo. 280*, A.I. Laboratory, M.I.T., Cambridge, Mass.

Hewitt, C. (1971) Procedural embedding of knowledge in PLANNER. *Proc. IJCAI-2*, London. Portsmouth: Eyre and Spottiswoode Ltd.

Kac, M. and Ulam, S. (1971) *Mathematics and Logic*. Penguin Books, England.

Reiter, R. (1972) The use of models in automatic theorem proving. *Tech. Report 72-09*, Dept. of Computer Science, Univ. of British Columbia.