MASSACHUSETTS INSTITUTE OF TECHNOLOGY
A. I. LABORATORY

Artificial Intelligence
Memo No. 280

April 1973

ELEMENTARY GEOMETRY THEOREM PROVING

Ira Goldstein

ABSTRACT

An elementary theorem prover for a small part of plane Euclidean geometry is presented. The purpose is to illustrate important problem solving concepts that naturally arise in building procedural models for mathematics.

Work reported herein was conducted at the Artificial Intelligence Laboratory, a Massachusetts Institute of Technology research program supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract Number N00014-70-A-0362-0003.

ELEMENTARY GEOMETRY THEOREM PROVING

ABSTRACT

An elementary theorem prover for a small part of plane Euclidean geometry is presented. The purpose is to illustrate important problem solving concepts that naturally arise in building procedural models for mathematics.

Ira Goldstein
January 31, 1973

CONTENTS

I. INTRODUCTION

II. ELEMENTARY PROBLEM SOLVING CONCEPTS
1. Domain
2. Consequent Theorems
3. Antecedent Theorems
3.1 Eliminating Synonyms
3.2 Deriving Corollaries
4. Experts
5. Naming
5.1 Canonical Names
5.2 Cycling
6. Controlling Computation
6.1 Diagram Filter
6.2 Uniqueness Filter
7. Failure
8. Planning
8.1 Depth-First Search
8.2 Plausible Move Generation
8.3 Incomplete Bindings and Constructions
8.4 The Use of "Contexts"
8.5 Example

III. KNOWLEDGE
1. Triangle Congruence Expert
2. Segment Equality Expert
3. Midpoint Expert
4. Angle Equality Expert
5. Parallel Line Expert
6. Parallelogram Expert

IV. PROBLEMS AND PROOFS

V. DIRECTIONS FOR FURTHER RESEARCH
1. Global Planning
2. Diagram Generation
3. Quantitative Increase in Mathematical Knowledge
4. Qualitative Increase in Mathematical Knowledge
5. Learning
6. English
7. 100% on Regents
8. Alternative Approaches to Mathematical Reasoning
9. Psychology of Human Problem Solving
10. Education

VI. BIBLIOGRAPHY

I. INTRODUCTION

Euclidean geometry is traditionally considered an excellent subject for introducing rigorous thinking. It is thought to reveal clearly the nature of axiomatic systems and logical rules of inference. However, all too often, the student's impression is that mathematics is either too obvious (THE AXIOMS) or too difficult (THE PROBLEMS) to be interesting.

Building a theorem prover is an exciting alternative to the usual classroom presentation. Its most fundamental virtue is the change in emphasis from:

What are the theorems of geometry?

to

How can geometry be used to solve problems!

Theorems become not simple declarative statements but strategies for solving problems. Planning methods for finding a proof amidst a surfeit of geometric knowledge are clearly brought into focus. The need for formal rigour changes from an arbitrary demand to a natural pre-requisite for building a computer program.

This paper presents an elementary theorem prover for a small part of Euclidean geometry. Section II introduces basic problem solving concepts which arise in building such a program. Section III details the scope of its geometric knowledge. Section IV shows the performance of the program on New York State Regents problems. Section V concludes with directions for further research.

Note for those familiar with Gelernter's program

Gelernter's geometry machine is the most well-known program for geometry theorem proving. Hence, it is worth distinguishing the presentation in this paper from Gelernter's work. Gelernter was primarily concerned with illustrating very general theorem proving ideas such as "chaining backwards" and "syntactic symmetry". This paper extends his work in three ways:

1. The basic approach of "chaining backwards" is still used. The difference from Gelernter's work is the emphasis on developing a high-level, "natural" formalism for representing mathematical knowledge as programs. Such a representation allows the builder to focus on mathematical rather than implementation issues. This is made possible by the recent development of goal-oriented languages such as PLANNER and CONNIVER.

2. A canonical naming scheme is developed to allow the program to identify the many synonyms for a given geometric entity. It provides computational efficiency and clarity by automatically identifying all the various names of such entities as lines, angles, and quadrilaterals.

3. A plausible move generator is discussed for guiding the search for a proof. It is coupled with knowledge for adding constructions. Constructions are not randomly made between unconnected points when all else fails, as was the case with Gelernter's program. Instead, they are generated for the specific purpose of making a desired strategy applicable to the current diagram. [Typical of this is completing a triangle in order to apply the strategy of proving segments equal via triangle congruence.]

II. ELEMENTARY PROBLEM SOLVING CONCEPTS

1. Domain

The first part of most elementary geometry couses develops theorems for proving:

triangles congruent
segments equal
lines parallel
angles equal
and quadrilaterals to be parallelograms.

This will be the domain for the theorem proving program developed in this paper. Excluded is knowledge related to inequalities, similar figures, circles, arcs and area. However, even when restricted to this limited domain, the theorem prover must cope with analyses of significant depth and breadth.

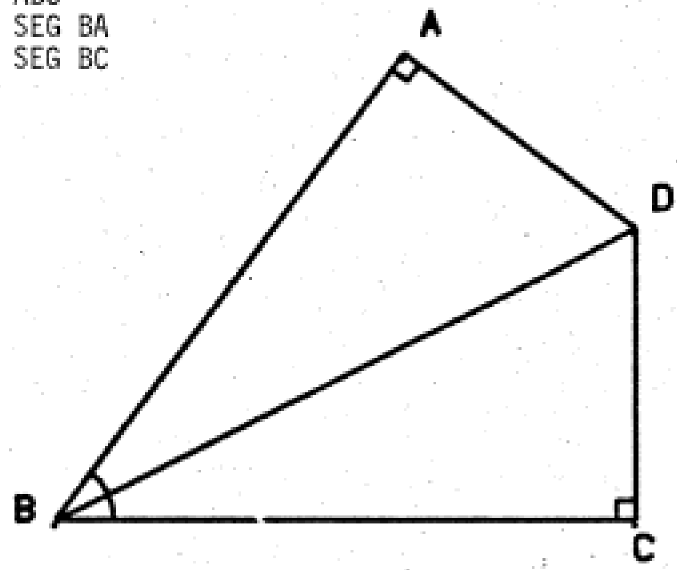
To develop insight into the problems that fall within this domain, two examples are offered. The first is a simple problem. Neither constructions nor a proof of any great depth is required.

PROBLEM: Gelernter, Theorem 1 (G-1)

STATEMENT: The angle bisector is equidistant from the rays of the angle.

HYPOTHESES: SEG DB BISECTS ANGLE ABC
SEG DA PERPENDICULAR SEG BA
SEG DC PERPENDICULAR SEG BC

OBJECTIVE: SEG AD = SEG CD



GELERNTER PROBLEM 1

PROOF: BTP ON GELERNTER, THEOREM 1

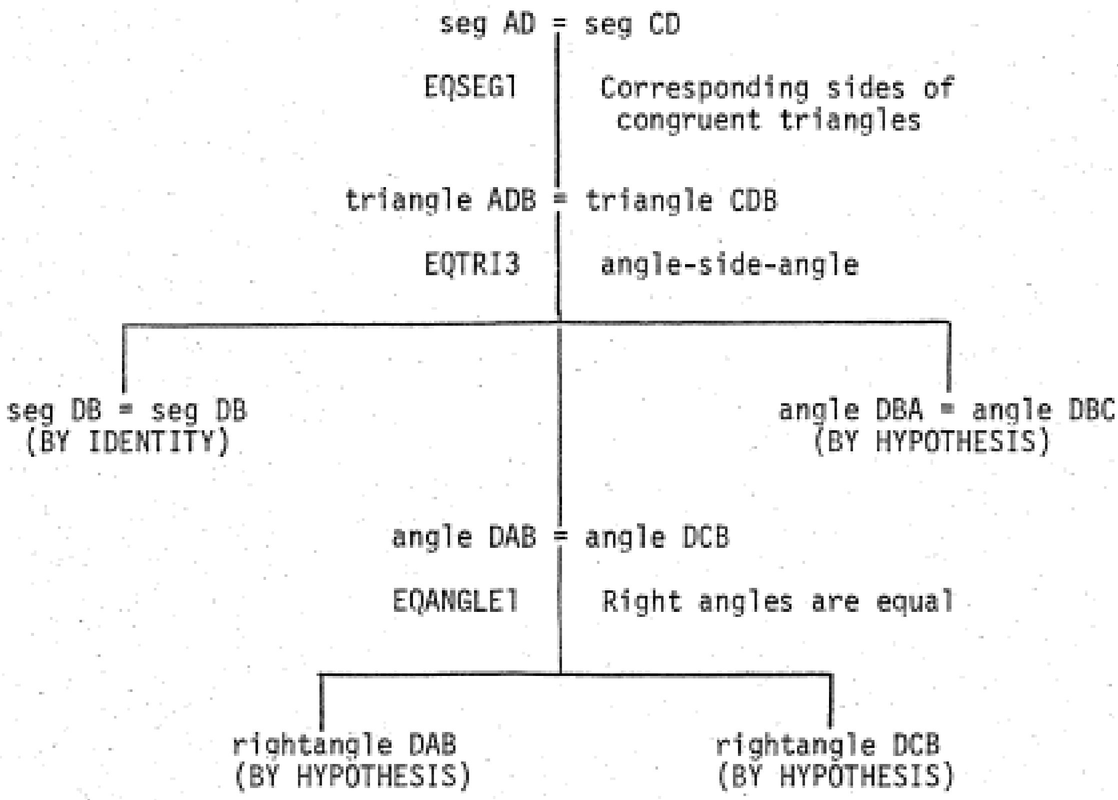
	STEPS	REASONS
1	SEG DB = SEG DB	;by identity
2	ANGLE DBA = ANGLE DBC	;byhhypothesis
3	RIGHTANGLE DAB	;by hypothesis
4	RIGHTANGLE DCB	;by hypothesis
5	ANGLE DAB = ANGLE DCB	;right angles are equal
6	TRIANGLE ADB = TRIANGLE CDB	;by aas
7	SEG AD = SEG CD	;corresponding sides of congruent triangles

QED

The proof is produced by the BASIC THEOREM PROVER (BTP), and implementation in MICRO-PLANNER of the ideas presented in sections I-1 through I-7. The "PROBLEM:" and "STATEMENT:" specifications are entirely for the reader's benefit. The BTP is given only the DIAGRAM, specified via the Cartesian coordinates of the points and a list of connections, the HYPOTHESES and the OBJECTIVE.

The proof for G-1 is stated in a traditional format. The reasoning process of our theorem prover, as well as most humans, is revealed far more clearly by examining a "goal tree".

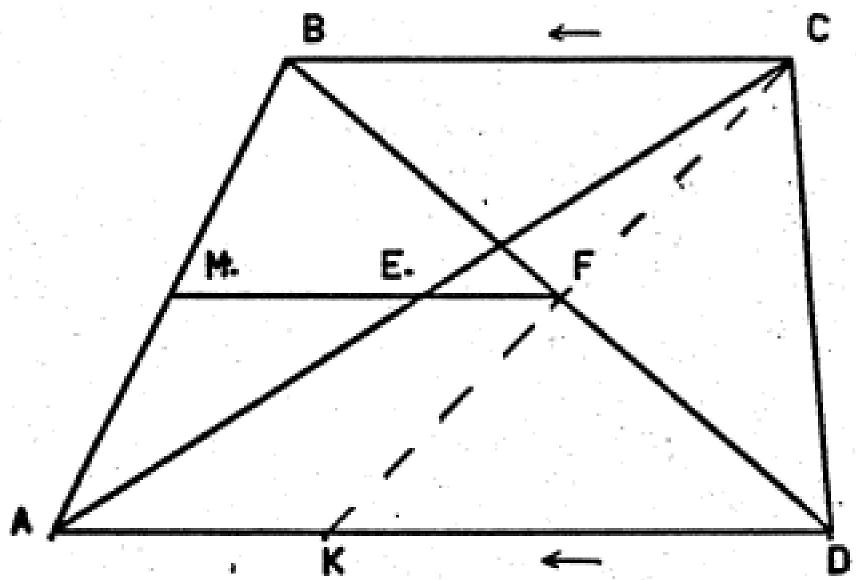
PROOF: BTP ON GELERNTER, THEOREM 1



The nodes of the tree are goals. The labels on edges are the mathematical theorems used to achieve the goals. The tips are known assertions, their justification given in parentheses.

The following example represents the most difficult of the five problems given by Gelernter. It is interesting for both the complexity of the proof and the need for a construction. A discussion of the knowledge needed to find the construction line in an intelligent way is postponed until Section 8. The proof presented below is given the auxiliary line CFK as a hypothesis.

PROBLEM: Gelernter, Theorem 5 (G-5)
STATEMENT: If the segment joining the midpoints of the diagonals of a trapezoid is extended to intersect a side of the trapezoid, it bisects that side.



GELERNTER PROBLEM 5

HYPOTHESES: LINE AD PARALLEL LINE BC
POINT E MIDPOINT SEG AC
POINT F MIDPOINT SEG BD
CONSTRUCT LINE CFK

OBJECTIVE: POINT M MIDPOINT SEG AB

PROOF: BTP ON GELERNTER, THEOREM 5 (G-5)

point M midpoint AB

MIDPOINT2 — A line bisecting one side of a triangle and parallel to the base bisects the other.

triangle ABD (HYPOTHESIS)

MF parallel AD

point F midpoint BD (HYPOTHESIS)

RENAMING

EF parallel AK

PARALLEL3 — The bisector of the sides of a triangle is parallel to the base.

triangle ACK (HYPOTHESIS)

point F midpoint CK

point E midpoint AC (HYPOTHESIS)

MIDPOINT1 — Definition

CF = FK

EQSEG1 — Corresponding sides of congruent triangles are equal.

triangle CFB = triangle KFD

EQTRI3 — Angle-angle-side

BF = FD

angle KDF = angle CBF

angle CFB = angle KFD

point F midpoint BD (HYPOTHESIS)

EQANGLE2 — Vertical angles

colinear CFK
colinear DFB
(HYPOTHESIS)

EQANGLE5 — Alternate interior angles of parallel lines are equal.

BC parallel AD (HYPOTHESIS)

Additional examples from Gelernter's articles and from the New York State Regents are provided in Section IV.

For each of the important goals listed at the beginning of this section, a student has a variety of mathematical theorems at his disposal. For example, triangle congruence could be proved via any of the usual three methods:

side-angle-side
angle-side-angle
side-side-side.

The New York State syllabus expects a student to know about fifty theorems for these objectives. Moreover, this breadth of mathematical knowledge is coupled with the fact that parallelism, congruence and equality are equivalence relations; and therefore justify transitivity. Hence, proof searches are far from trivial in both their possible depth and breadth. See Section III for an outline of the mathematical theorems in this domain.

The following sections examine procedural representations for a variety of mathematical knowledge which together provide a foundation for a theorem proving program.

2. Consequent Knowledge

The theorem prover marches forward by stepping backwards. Its basic mode of operation is using geometry theorems to move from the conclusion to the hypotheses. This provides more direction than aimless deductions beginning from the hypotheses. The importance of this style of deduction was first emphasized by Gelernter.

To direct this analysis, geometry theorems will be given the following procedural representation.

```
STRATEGY EQSEG1:
        TO-PROVE: seg XY = seg UV
        ESTABLISH: triangle XYZ = triangle UVW
        REASON: corresponding parts of congruent triangles
STRATEGY EQANGLE1:
        TO-PROVE: angle XYZ = angle UVW
        ESTABLISH: 10 rightangle XYZ
                   20 rightangle UVW
        REASON: right angles are equal
STRATEGY EQTRI3:
        TO-PROVE: triangle XYZ = triangle UVW
        ESTABLISH: 10 seg YZ = seg VW
                   20 angle XYZ = angle UVW
                   30 angle YZX = angle VWU
        REASON: congruence by asa
```

For the MICRO-PLANNER user, a line-oriented parser can convert these strategies to consequent theorems.

For a strategy to be applied, the "TO-PROVE" line must match the current goal. "ESTABLISH" is followed by a list of sub-goals which must be satisfied to achieve success. The variables X, Y, Z,... are bound to the points in the current goal. In the event that all of the sub-goals cannot be proved, the theorem prover automatically tries any remaining matching strategies.

The use of goal-oriented languages such as "PLANNER" and "CONNIVER" provide these powerful procedural tools of "chaining backwards" and "pattern matching". They free the student to concentrate immediately on the central intellectual problems associated with representing geometric knowledge in procedural terms.

3. Antecedent Knowledge

3.1 Eliminating Synonyms

Antecedent programs serve two simple roles. The first is to pre-process problems into terms understood by the theorem prover. This is made necessary by the many ways geometry provides for expressing the same hypothesis. For example, in the example G-1, the hypothesis "ANGLE ABD = ANGLE CBD" might have been stated as:

SEG BD BISECTS ANGLE ABC.

Of course, a strategy could be added to the theorem prover for proving angle equality by finding a bisector.

```
STRATEGY EQANGLE
        TO-PROVE: angle XYZ = angle WYZ
        ESTABLISH: seg YZ bisects angle XYZ
        REASON: definition
```

However, the problem can be converted to a preferred form before beginning the process of applying strategies to the goal. This has the virtue of not overburdening the strategies with restatements of the same piece of knowledge. Clarity and economy are achieved.

```
CONVERSION ANGLE-BISECTOR
        GIVEN: seg DB bisects angle ABC
        ASSERT: angle ABD = angle CDB
        FORGET: <given>
```

Note that the original assertion, once restated, is no longer necessary; and, hence, forgotten. A parser can transform these conversions to PLANNER antecedent theorems.

Additional examples are:

```
CONVERSION MEDIAN
        GIVEN: seg AM median-to seg BC
        ASSERT: seg BM = seg MC
        FORGET: <given>
CONVERSION PARALLEL
        GIVEN: seg XY parallel seg UV
        ASSERT: line XY parallel line UV
        FORGET: <given>
```

It is a heuristic question as to whether a given term should be eliminated as interface knowledge. The issue is not whether it is formally reducible to other terms; but, whether it is computationally efficient to design an expert around the concept. For example, the objective of a problem might be to prove that a triangle is isosceles. This could be converted into a segment equality proof. However, such a conversion would ignore the special fact that the segments are part of a triangle. The insight for constructing a perpendicular bisector, angle bisector or median would be lost. Hence, it is not worthwhile to eliminate "isosceles". It is a valuable problem solving concept.

3.2 Deriving Corollaries

The second role for antecedent programs is to assert commonly needed corollaries of any hypotheses or objectives successfully proved in the course of analysis.

```
COROLLARY EQTRI-1
        GIVEN: triangle XYZ = triangle UVW
        ASSERT: seg XY = seg UV
                seg YZ = seg VW
                seg XZ = seg UW

COROLLARY EQTRI-2
        GIVEN: triangle XYZ = triangle UVW
        ASSERT: angle XYZ = angle UVW
                angle YZX = angle VWU
                angle ZXY = angle WUV

COROLLARY PARALLELOGRAM-1
        GIVEN: parallelogram ABCD
        ASSERT: seg AB = seg CD
                seg AD = seg BC
```

Here the original assertion remains important and is not forgotten.

The process of "chaining backwards" is initiated by the problem being solved. The derivation of corollaries, however, takes place independent of the particular objective of the proof. The computation is expended in the hope that the corollaries will be useful later. However, they may be unneeded. The cost is generally not prohibitive since such expansions occur only at the infrequent times when a major subgoal has been achieved. Nevertheless, it is a heuristic judgment by the programmer whether any particular corollary should be represented as antecedent knowledge as opposed to being rederived later if needed.

4. Experts

There are usually many strategies for each goal. For example, triangle congruence can be proved by

EQTRI1 - side-side-side
EQTRI2 - side-angle-side
EQTRI3 - angle-side-angle

For conceptual clarity and computational efficiency, strategies with a common goal are organized into experts. For the limited portion of Euclidean geometry discussed in these pages, five major experts are sufficient.

triangle congruence
segment equality
angle equality
parallel lines
parallelograms

An expert attempts to satisfy its goal for a given set of points as follows:

Convert the goal into canonical form.
↓
Has the goal already been proved? —— YES ⟶ success
NO ↓
Is the goal worth trying? —— NO ⟶ failure
YES ↓
Does any strategy succeed? —— NO ⟶ failure
YES ↓
Success - remember as lemma.
↓
Derive corollaries.

Previous sections have explored the representation of theorems as consequent programs [strategies] for proving goals and as antecedent programs for deriving corollaries. The next section will examine the need for a canonical naming scheme. This is followed by an analysis of how computation can be controlled by various filters and a plausible move generator.

5. Naming

5.1 Canonical Names

The theorem prover has a collection of known facts about the problem. Initially, these are simply the hypotheses and diagram. Each time an expert succeeds, the result is remembered as a lemma. This set of assertions is called the "database". An expert begins its analysis by checking whether its goal is already in the database via:

DATABASE? <PATTERN>

Unfortunately, some case must be given as to the form in which hypotheses and lemmas are remembered. In G-1, for example, the goal is to prove "SEG AD = SEG CD". Suppose the theorem prover is given the hypothesis "SEG DA = SEG CD". A human mathematician would not see the problem as he implicitly identifies AD and DA as two names for the same line segment. The program, however, must be given such knowledge explicitly.

One solution is for the expert to check all possible names. For segment equality, this would require the database inquiry to be made for seven additional variants of the basic pattern.

seg XY = seg UV	seg UV = seg XY
seg YX = seg UV	seg UV = seg YX
seg XY = seg VU	seg VU = seg XY
seg YX = seg VU	seg VU = seg YX

Such a solution has obvious combinatorial problems. For triangle congruence, there would be 72 variations.

An alternative scheme is to convert hypotheses and goals to a preferred or "canonical" format. Of course, if this approach is to be computationally less costly, the conversion function must be efficient. The following paragraphs describe one way to associate a unique form to each of the geometric statements which the theorem prover must understand. The technique will be to build an ordering for the various names for each geometric entity. Then the canonical name can be chosen as the first in the resulting sequence.

Geometric statements describe relations between points. The points can be ordered by their position in the diagram. [An alphabetic ordering based on the names for the points might also be used.]

A<B if xcor(A)<xcor(B) or xcor(A)=xcor(B) and ycor(A)<ycor(B)
where xcor(A) = x coordinate in the diagram of point A
and ycor(A) = y coordinate in the diagram of point A

Note that an ordering on sets of points can be built from this ordering on points in the same fashion as this ordering on points was built from a numerical ordering on numbers.

These orderings on points and sets of points plus some additional Cartesian knowledge can be used to define a canonical format for geometric statements. The predicate "CF" is true if its input is in canonical format.

CF (SEG AB)	<=> A < B
CF (SEG AB = SEG CD)	<=> CF (SEG AB) & CF (SEG CD) & {A,B} < {C,D}
CF (TRIANGLE ABC)	<=> A = MIN {A,B,C} & A,B,C in clockwise order
CF (TRIANGLE ABC=TRIANGLE DEF)	<=> A = MIN {A,B,C,D,E,F} & CF (TRIANGLE ABC)
CF (SQUARE ABCD)	<=>A = MIN {A,B,C,D} & A,B,C,D in clockwise order
CF (LINE AB)	<=>A<B & A,B endpoints of line
CF (ANGLE ABC)	<=> A,C endpoints of rays originating at B

These criteria imply procedures for converting any geometric statement involving these terms to canonical format. This is the first step taken by an expert. Subsequent database inquiries, applications of strategies and the final assertion are all performed on the preferred form of the pattern.

5.2 Cycling

There is another dimension to the naming problem not handled by canonical names. Proving two triangles congruent by angle-side-angle is not completely implemented by EQTRI3 as described earlier. The strategy should be:

```
STRATEGY EQTRI3
        TO-PROVE: triangle XYZ = triangle UVW
        ESTABLISH:
          (or
           (and ;first pair of angles
                 seg YZ = seg VW
                 angle XYZ = angle UVW
                 angle YZX = angle VWU)

           (and ;second pair of angles
                 seg XZ = seg UW
                 angle YZX = angle VWU
                 angle ZXY = angle WUV)
           (and ;third pair of angles
                 seg XY = seg UV
                 angle ZXY = angle WUV
                 angle XYZ = angle UVW))
```

The extension of EQTRI3 represents the fact that there are three possible applications of angle-side-angle for any given pair of triangles. For clarity, it is desirable to separate this need to cycle through possible applications from the basic geometric theorem. EQTRI3 can be left in its original form, providing the triangle-congruence expert is given explicit advice with respect to the need for multiple applications of the congruence theorems. Similar cycling advice must be given to the other experts for each of their strategies.

One form that this advice can take is that of a special strategy which applies its brethren to the appropriate cycles of the basic pattern. For example, for EQTRI1 (SSS), EQTRI2 (SAS), and EQTRI3 (ASA), EQTRI0, the cycling strategy, becomes:

```
STRATEGY EQTRIO
      TO-PROVE: triangle XYZ = triangle UVW
      ESTABLISH: triangle YZX = triangle VWU using EQTRI2, EQTRI3
                 triangle ZXY = triangle WUV using EQTRI2, EQTRI3
```

Note that EQTRI1 (SSS) need not be applied more than once.

There are several disadvantages to a separate cycling strategy that takes effect after all of the other strategies have been applied. It has the disadvantage of repeating computation common to all cycles of a given strategy on a given set of points. For example, cycling is used to apply PARALLEL2, [parallel lines via equal corresponding angles], to all four pairs of corresponding angles. Choosing a particular transversal is common to all cycles and is unnecessarily repeated four times. A better solution is to have each basic strategy try all cycles at the time it is applied. This approach results in EQTRI3 having the following representation:

```
STRATEGY EQTRI3
        TO-PROVE: triangle XYZ = triangle UVW
        ESTABLISH: 10 seg YZ = seg VW
                   20 angle XYZ = angle UVW
                   30 angle YZX = angle VWU
        REASON: congruence by ASA
        CYCLE: triangle YZX = triangle VWU
               triangle ZXY = triangle WUV
```

6. Controlling Computation

6.1 Diagram Filter

There are a variety of methods for preventing the theorem prover from pursuing dead ends. The most important is using the diagram to reject false goals. This was an important contribution of Gelernter. In human terms, it amounts to the advice: "If it looks wrong in the picture, don't bother."

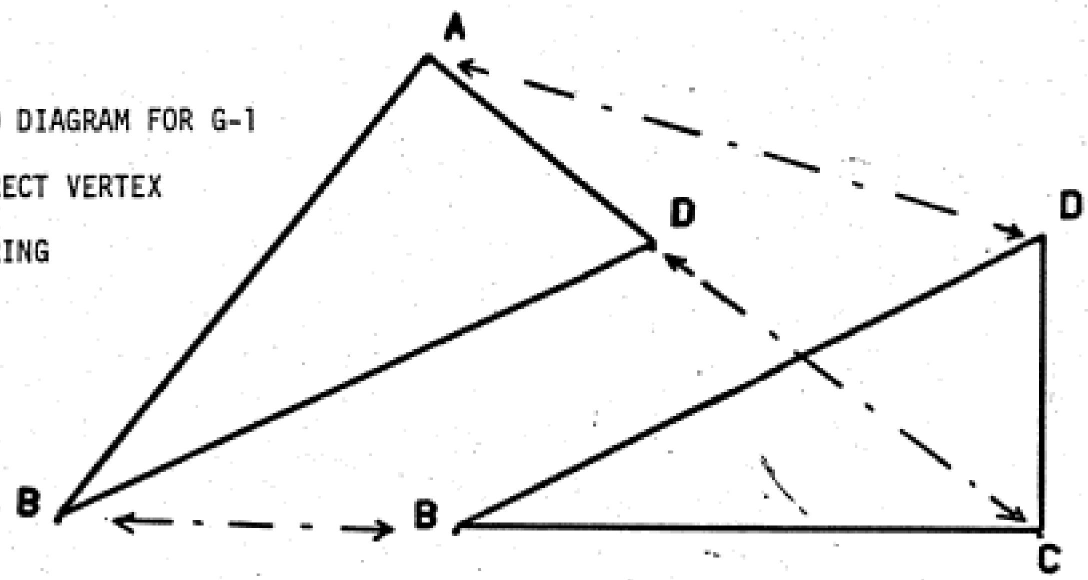
Cartesian and Euclidean geometry provide different formalism for describing the same facts. The points of the diagram must be chosen so that the Euclidean hypotheses are satisfied. If so, then any goal false in the diagram is not provable. Thus, no valuable lines of inquiry are ruled out by the diagram filter.

Of course, using the diagram filter would be pointless if it were very costly in computation time. However, Cartesian predicates are very efficient, requiring only simple numerical computations.

The diagram filter takes the knowledge of the theorem prover out of a purely Euclidean world. The combination of Cartesian geometry, Euclidean geometry and, as we shall see later, planning meta-knowledge is an important virtue of this procedural approach. It mirrors the way in which human mathematical reasoning in built upon an interplay of many levels of knowledge.

An example of the usefulness of this filter can be found in G-1. The theorem prover might have attempted proving AD=CD by establishing

TRIANGLE ADB = TRIANGLE DCB



EXPLODED DIAGRAM FOR G-1
INCORRECT VERTEX
PAIRING

This pairing of the vertices, however, is bound to fail. The diagram filter rejects this false path since for the points chosen:

SEG DB ≠ SEG CB.

The diagram filter cannot replace the theorem prover. It is possible for too much to be true in the diagram. Fortuitously chosen points could result in ANGLE ABD = ANGLE ADB in G-1. This is not a necessary consequence of the hypotheses. Thus, a proof is required to substantiate the claims of the diagram.

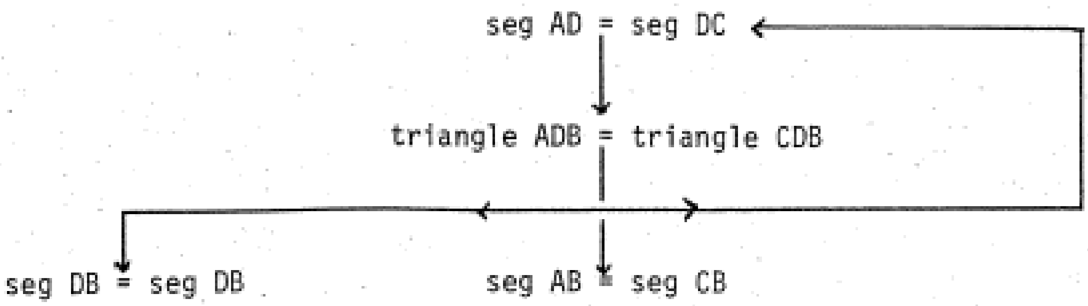
6.2 Uniqueness Filter

Another heuristic for limiting fruitless search is:

Do not try an unsuccessful goal a second time.

The assumption is that the theorem prover applied all of its skill to the goal the first time. A second attempt would simply result in repeated failure. Note that the prohibition is not against using the same strategy more than once. Rather it only restricts using the same strategy on exactly the same points twice.

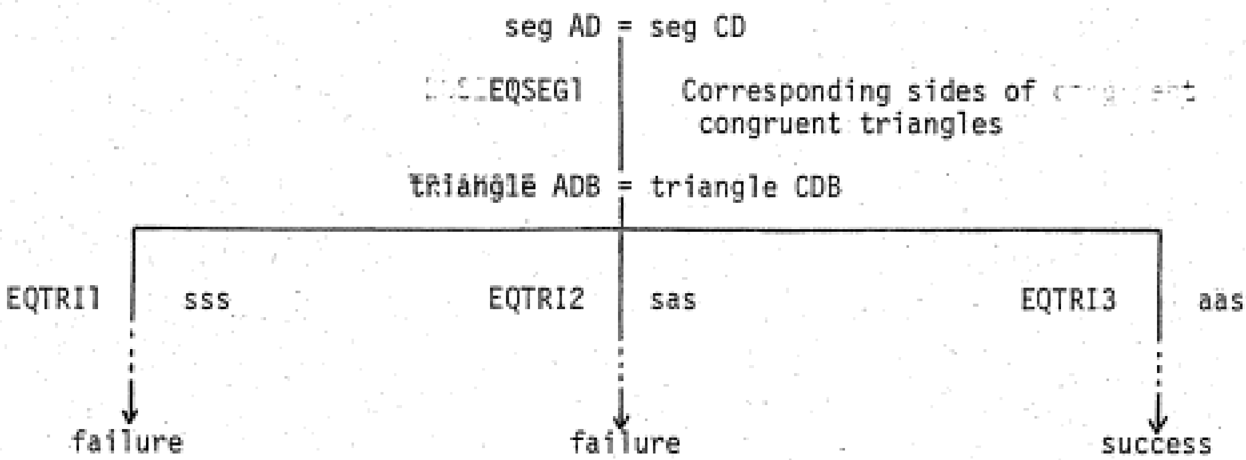
Without a uniqueness restriction, the side-side-side strategy for triangle congruence could result in the following loop in G-1.



Remembering past goals to avoid repetition is costly in terms of space. However, endless looping is disastrous and makes a uniqueness filter necessary.

7. Failure

Even with its computational filters, it is possible for the theorem prover to reach a dead end. For example, in G-1, the side-side-side strategy would fail if used to prove congruence. Goal-oriented languages such as PLANNER and CONNIVER provide the ability to back-up and try alternative strategies when a failure occurs.



Back-up is simplified by the structure of the experts. If successful, an expert summarizes its computation with an assertion. This lemma is remembered even if the larger objective for which the expert was needed fails. The justification is that the proof of the lemma remains valid regardless of the failure of the larger strategy. It is possible that the sub-goal may never be needed in the final proof. However, the heuristic is that the slight cost in space is worthwhile to avoid possible time consuming re-derivations.

If unsuccessful, the uniqueness filter prevents the expert from ever attempting the same goal again. Thus, in both cases, the details of the analysis need not be remembered. The cost of back-up is therefore minimized.

8. Planning

8.1 Depth-First Search

The basic operation of the theorem prover consists of:

goal ⟶ expert ⟶ strategies ⟶ sub-goals ⟶ ... ⟶ hypotheses

With failure back-up, this simple pattern-directed reasoning is adequate to solve many simple proofs. However, more intelligent planning is needed as the problems become more complex.

Consider the problem G-5 presented in Section 1. The proof is reasonably short. However, a theorem prover built of experts with failure back-up requires 50 pages of analysis to find this proof.

> This number is obtained from a trace of the BTP, a MICRO-PLANNER program embodying all of the ideas discussed so far in this paper. This includes a diagram and a uniqueness filter. The excessive search it represents is due to indirect strategies such as transitivity and angle equality via equal supplements.

Such computation is far too tedious to be illuminating.

8.2 Plausible Move Generation

A solution is the introduction of a "plausible move generator". Each expert could have a PMG whose job would be to choose which strategies to apply first.

> A move in the geometry game is the application of a particular strategy for a particular choice of points.

There is a variety of knowledge that a PMG would need. For example, evidence for a strategy, independent of the choice of points might include:

> Context - The location of the strategy and problem in the syllabus can be useful. The page number of a problem is given. The computer has the page numbers for every strategy. Plausibility is the strategy whose page number first precedes the problem.' The theory behind this heuristic is that authors generally have some rationale for their organization of a text. This can include an unspoken theory of the relevance of various theorems to problems. At the very least, a problem is usually not

assigned unless it can be solved from material already presented in the text. [Careful examination of this heuristic should eventually lead to more subtle interconnections between knowledge and classes of problems than the linear structure of a text.]

Repetition - Negative weight is given for repetitive use of any strategy on the current proof path. This can be used as a restriction on the number of times a given strategy can recurse.

Directness - Transitivity and arithmetic are indirect strategies in that they shift the focus of attention to another pair of angles, segments, etc. Negative weight should be charged for such indirectness.

A fully specified move includes the choice of points to which the strategy will be applied. As we have already seen, the points chosen must make all of the goals of the strategy true in the diagram. In addition, evidence for a particular choice of points might include:

The number of sub-goals of the strategy that have already proved or asserted.

The number of other assertions in which these points occur.

8.3 Incomplete Bindings and Constructions

Inlits consideration of the next "move", a PMG must handle "constructive" strategies. These are theorems that require additional points and line segments beyond those mentioned in the basic goal pattern. For example, EQTRI1 (SSS) is not constructive. The existence of two triangles implies the existence of the angles and line segments which make up the triangles. But proving that two line segments are equal because they are corresponding parts of congruent triangles is constructive. The theorem prover must find the additional points needed to form the triangles.

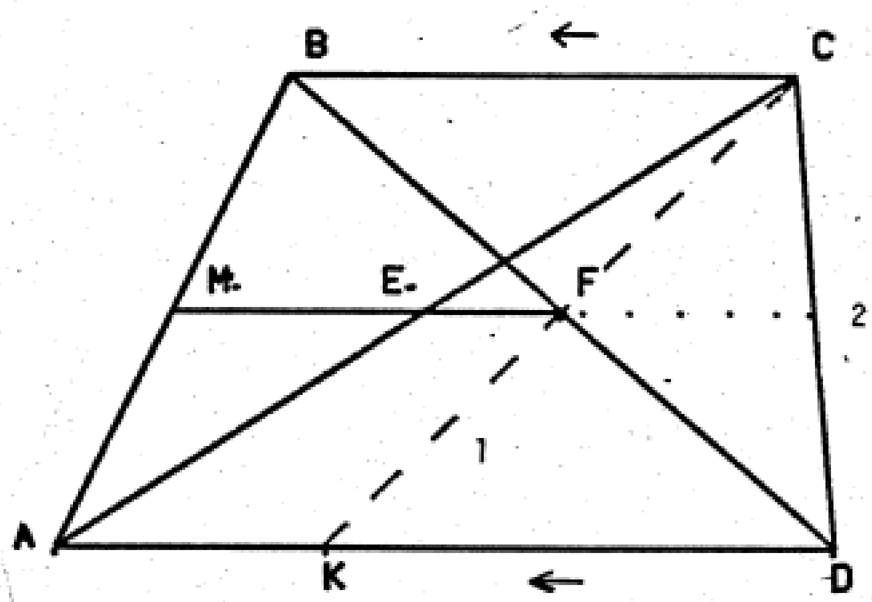
An example of a constructive strategy used to prove G-5 is:

```
STRATEGY MIDPOINT2
        TO-PROVE: point M midpoint seg AB
        ESTABLISH: 10 point C such-that triangle ABC
                   20 point F such-that line BFC
                   30 and (prove: point F midpoint seg BC)
                           prove: line MF parallel line AC)
        REASON: A line bisecting one side of a triangle and parallel to
                the base bisects the other.
```

If the additional points and/or line segments do not exist in the diagram, the theorem prover is faced with the "construction" problem.

Should additional points and/or lines be added; and if so, where? The particular strategy involved plus knowledge of those points which do exist constitute constraints as to where additional points can be created. Sometimes these constraints are sufficient to determine the new point or line.

For example, consider G-5. Proving "LINE MF PARALLEL LINE AD" requires a construction. PARALLEL3 is the theorem that a line bisecting two sides of a triangle is parallel to the base. This strategy is very specific in its needs. Segment AC is the obvious candidate for one side of this triangle. This leads to two constructions:



GELERNTER PROBLEM 5

1. Construct CFK
or 2. Extend MEF through F to CD.

The first construction is preferred since it is a transversal between two lines known to be parallel. This illustrates the way in which strategies direct the construction process.

At other times, there is still some freedom with respect to the new points. The constructor can generate its own constraints. For example, this would include choosing a new point such that the line segment formed is

parallel or perpendicular or a median

to an important existing line segment. This represents an effort to maximize the return in the form of additional properties of a given construction.

The constructor can return the additional points and/or line segments needed to complete the binding as well as an estimate of the plausibility of the construction. This evidence is used by the PMG. This can be added into the plausibility of the total binding.

This does not represent a complete solution to the construction problem. It does not involve any global planning. For example, no effort would be made to put in a line because of symmetry. [See Wong.] Constructions are tied to particular strategies. However, it is an important part of construction knowledge. Witness the comparison with Gelernter's original approach to new line segments - when all else fails, connect any two points.

8.4 The Use of "Contexts"

Constructions for a plausible move might not pay off. The particular path could fail. In such a case, since the constructions were done for a particular purpose, they should be undone; unless interesting lemmas were proved along the way. CONNIVER with its ability to define sub-contexts, provides a powerful tool for undoing constructions.

The number of moves for a given goal may be very large. This would make it prohibitively expensive to generate all possible moves and order them by their plausibility. This is especially foolish since plausibility is only a qualitative distinction between great, good, fair and poor. The alternative would be to try good-great possibilities immediately. This requires that the generation of moves "hang", resuming later if necessary. This switch between depth-first and breadth-first exploration again requires the ability to move between contexts.

The PTP, an extension of the BTP to include a plausible move generator has been simulated, but not yet implemented. The BTP is written in MICRO-PLANNER. The need for contexts and hanging suggests that the PTP should be implemented in CONNIVER.

8.5 Example

Hand simulation of the PTP results in great efficiency and directness. For example, for G-5, analysis requires only a fraction of the BTP's cost, even though the construction line CFK is not given with the hypotheses.

III. KNOWLEDGE

Section II introduced the basic procedural organization for our elementary Euclidean geometry theorem prover. This section provides an outline of the mathematical knowledge included in the program. For each expert, the geometric theorems behind the strategies are stated. Any special characteristics or problems in converting a theorem to program format are mentioned.

This section does not include a discussion of the Cartesian knowledge used by the diagram filter and the orderings used to generate canonical names. The implementation of both of these functions is straightforward.

1. Triangle Congruence Expert

A. Theorems

EQTRI1 - two triangles are congruent if the three sides of one are equal to their counterparts in the other.

EQTRI2 - two triamgles are congruent if two sides and the included angle of one are equal respectively to two sides and the included angle of the other.

EQTRI3 - two triangles are congruent if any two angles and a side of one are equal to their counterparts in the other.

EQTRI4 - two right triangles are congruent if the hypotenuse and a leg of one are equal to the hypotenuse and a leg of the other.

EQTRI-PARALLELOGRAM - a diagonal divides a parallelogram into two congruent triangles.

EQTRI-TRANSIVITY - two triangles are congruent if they are congruent to a third.

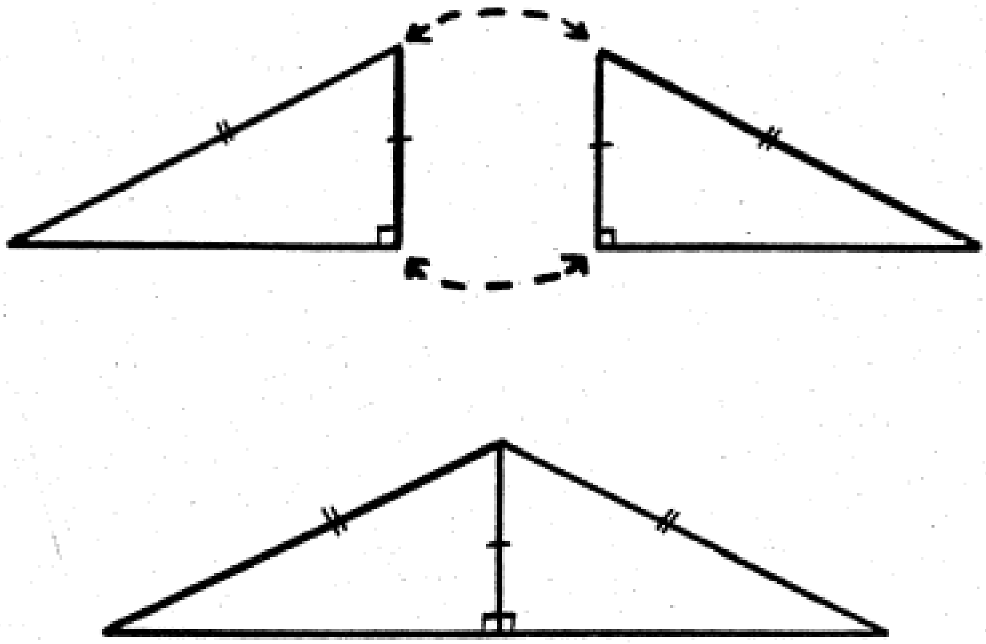
B. Fine Points

a. Triangle congruence refers to a particular identification of the vertices of the two triangles. Thus,

TRIANGLE ABC = TRIANGLE DEF

corresponds to A paired with D, B paired with E, and C paired with F. Proving triangle BAC congruent to triangle DEF is a separate problem.

b. The theorem "the hypotenuse and a leg of one right triangle are equal to the hypotenuse and a leg of another" requires a construction. The triangles are placed back-to-back.



COINCIDENT SIDE CONSTRUCTION

This seems to be a relatively specialized type of construction. Hence, the theorem is repeesented explicitly. The constructor of the PMG does not translate figures in this fashion. Further research is necessary to see whether translation is a technique that the constructor should consider.

c. EQTRI3 is used to represent both angle-side-angle and angle-angle-side. It proves congruence if two angles and <u>any</u> side of one triangle are equal to the corresponding parts in the other. This is mentioned to emphasize that the theorem prover is not simply a copy of the standard axiomitization for Euclidean geometry. Rather, the objective is to find the most powerful representation for doing actual problem solving. As such, we are free to use whatever geometric knowledge we may need to justify the inclusion of some particular strategy. However, we must face such issues as

time versus space tradeoffs
explicit versus implicit representation
heuristic versus algorithmic strategies

that a formal mathematical representation ignores.

d. The information that triangle congruence is an equivalence relation is represented in several ways. The canonical-name transformation identifies reflexive variations. Identity and transitivity are implemented by strategies for "=".

2. Segment Equality Expert

A. Theorems

EQSEG1 - corresponding sides of congruent triangles are equal.

EQSEG2 - the midpoint of a line segment divides it into two equal segments.

EQSEG3 - sums, differences and doubles of equal line segments are equal.

EQSEG4 - halves of equal line segments are equal.

EQSEG5 - sides of a triangle opposite equal base angles are equal.

EQSEG-TRANSIVITY - line segments equal to a third are equal to each other.

B. Fine Points

a. The pattern for EQSEG2 is "SEG AM = SEG MB". Observe that this performs limited filtering in its own right. It only matches two segments if they share a common endpoint.

Ordinarily, one might expect to try EQSEG2 for all four cycles of the segment equality pattern "SEG XY = SEG UV". The use, however, of a Cartesian lexical ordering has an interesting side benefit. The pattern in canonical form guarantees that any midpoint M be such that Y = M = U since XMY are colinear and consequently X<Y = U<V.

b. Surprisingly, all three of the following theorems for the algebra of segments can be represented by the same strategy:

i. Sums of equal line segments are equal.
ii. Differences of equal line segments are equal.
iii. Doubles of equal line segments are equal.

Note that summation for Euclidean geometry is defined as AB + BC = AC if and only if ABC are colinear.

```
STRATEGY EQSEG3
        TO-PROVE: seg AB = seg DE
        ESTABLISH: 10 thamong C (LINE AB)
                   20 thamong F (LINE DE)
                   30 seg CA = seg FD
                   40 seg CB = seg FE
```

There are three regions from which the point C might be chosen:

Region C2 corresponds to AC + CB = AB whereas regions C1 and C3 correspond to BC - AC = AB and AC = BC = AB respectively. This encompasses sums and differences of equals being equal. Doubles of equals being equal is covered by point C being chosen as the midpoint of AB. For this case, CA = FD must still be proved. However,

CB = CA by midpoint expert
CA = FD lemma
FD = FE by midpoint expert.

c. EQSEG5, that "sides opposite equal angles of a triangle are equal", is often proved by dropping a perpendicular to the base and demonstrating triangle congruence via angle-angle-side. This generalizes to a useful heuristic for the constructor.

> To equate parts of a given triangle, construct a perpendicular, angle bisector or median and prove congruence. The particular choice depends on what else is known about the triangle.

Another proof for EQSEG5 called the "pons asinorum" is illustrated in Section IV, Regents problem 32, June 69. This proof is amusing as it involves proving the triangle "congruent to itself".

3. Midpoint Expert

A. Theorems

MIDPOINT1 - M is the midpoint of segment AB if AM = MB.

MIDPOINT2 - a line bisecting one side of a triangle and parallel to the base bisects the other side.

B. Fine Points

Assertions about midpoints are not eliminated via a "conversion" to statements about segment equality. This is because AM = MB is not equivalent to M being the midpoint of AB. The difference is that segment equality does not imply that M is colinear with AB.

4. Angle Equality Expert

A. Theorems

EQANGLE1 - straight angles are equal.

EQANGLE22- vertical angles are equal.

EQANGLE3 - right angles are equal.

EQANGLE4 - corresponding angles of congruent triangles are equal.

EQANGLE5 - alternate interior angles of parallel lines are equal.

EQANGLE6 - corresponding angles of parallel lines are equal.

EQANGLE7 - sums, differences and doubles of equal angles are equal.

EQANGLE8 - havles of equal angles are equal.

EQANGLE9 - base angles of an isosceles triangle are equal.

EQANGLE-TRANSIVITITY - angles equal to the same angle are equal to each other.

RIGHTANGLE1 - perpendicular lines form right angles.

B. Fine Points

a. Proving that two angles are vertical angles is made computationally efficient by relying completely on the diagram. Strictly speaking, a theorem prover that uses such a strategy is not producing a rigorous Euclidean proof. However, such a strategy is in consonance with the basic goal of producing a reasonable model of human problem solving. Indeed, in any area of mathematics, certain lemmas are accepted without tedious formal proof. Thus, the combinatorics for establishing colinearity in terms of Euclidean predicates is not an interesting issue in developing a real-time performance model for this area of mathematics.

b. The calling pattern for EQANGLE2, the vertical angle strategy is:

ANGLE ABC = ANGLE DBF.

Notice that this pattern matches only pairs of angles that have a common vertex. Whether or not the angles actually are vertical, this information is sufficient to eliminate the alternate interior angle strategy. Alternate interior angles cannot share a vertex. This fact is implemented by erasing EQANGLE3 for the duration of the goal.

> The erasing of another strategy is useful for computational control. The cost of the alternate interior angle strategy is avoided even if EQANGLE2, the vertical angle strategy, fails. Such a careful consideration of when discoveries in one strategy allow the theorem prover to prune subsequent computation is important if our model is to be capable of supporting real problem solving.

5. Parallel Line Expert

A. Theorems

PARALLEL1 - alternate interior angles form parallel lines.

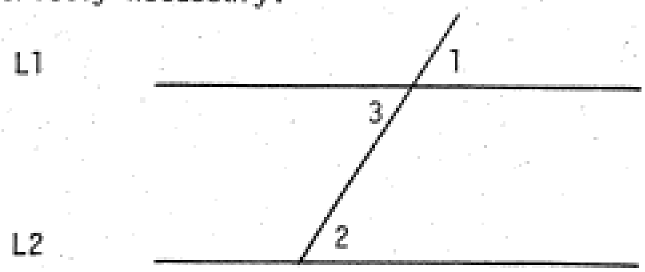
PARALLEL2 - equal corresponding angles form parallel lines.

PARALLEL3 - the bisector of the sides of a triangle is parallel to the base.

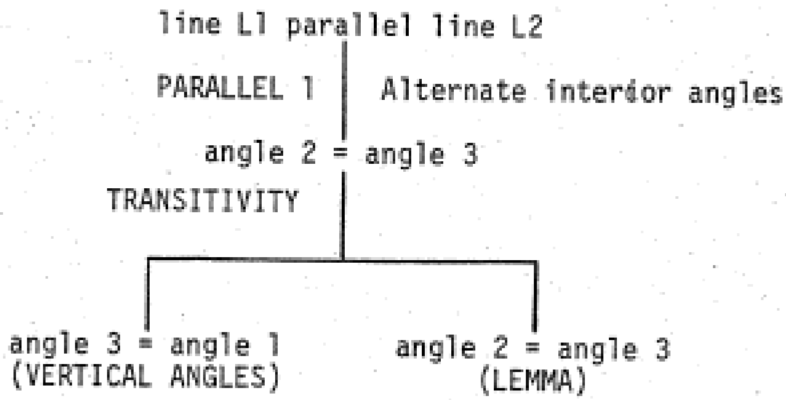
PARALLEL-TRANSITIVITY - two lines parallel to a third are parallel to each other.

B. Fine Points

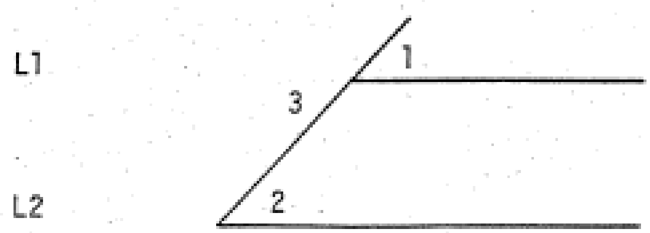
a. Given PARALLEL1, the alternate interior angle strategy, another program using the knowledge that lines are parallel if corresponding angles are equal is not strictly necessary.



A proof using the alternate interior angle strategy is ordinarily possible.



However, it is possible that angle 3 does not exist in the diagram. L1 and L2 may not extend beyond the transversal.



In such a case, the lack of an explicit "corresponding angle" strategy necessitates the construction of extending L1 and L2. This trick must be given to the constructor.

6. Parallelogram Expert

A. Theorems

COROLLARIES - the following corollaries are always derived whenever a quadrilateral is asserted to be a parallelogram [by either a hypothesis or a lemma.]

1. Both pairs of opposite sides are parallel
2. Both pairs of opposite sides are equal
3. Two opposite sides are equal and parallel
4. The diagonals bisect each other
5. Both pairs of opposite angles are equal.

PGRAM1 - prove congruence of the triangles formed by a diagonal. If necessary, construct the diagonal. Give preference to the diagonal that creates the pair of triangles about which the most is already known.

B. Fine Points

Each of the corollaries listed above is an equivalence. Hence, they could also be used as strategies to prove that a quadrilateral is a parallelogram. Examining the proofs for these equivalences indicates that constructing one or both diagonals is the basic technique. The construction of diagonals is generally needed when deriving one of the above properties from another for some quadrilateral as a lemma in a larger problem. Hence, for economy and clarity, only the core strategy of proving the congruence of the triangles formed by the diagonals is represented.

IV. PROBLEMS AND PROOFS

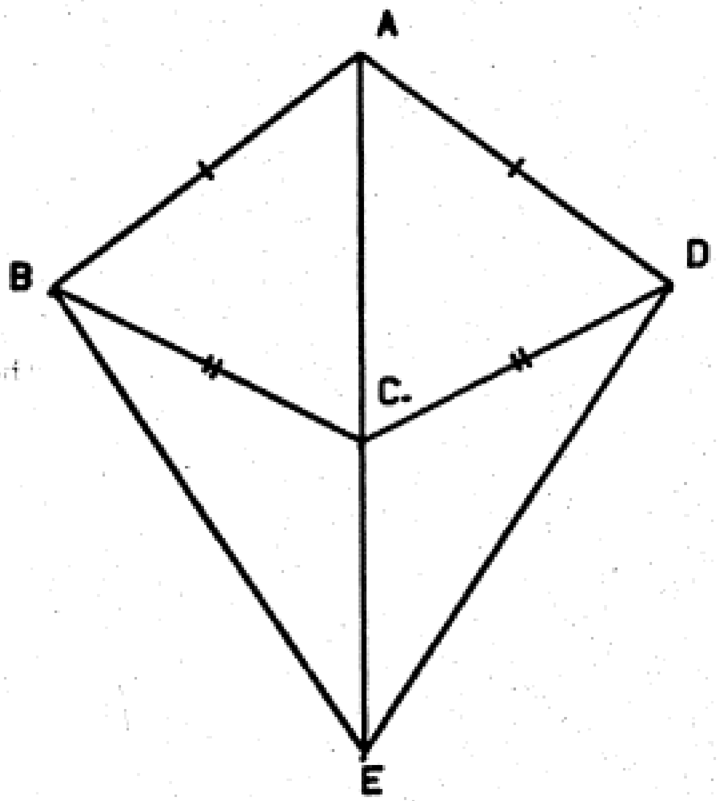
The theorem prover developed in Section II is capable of proving all five examples given by Gelernter in his articles in COMPUTERS AND THOUGHT. The program finds the same proofs given by Gelernter. Its analysis is improved in that constructions are far more motivated. Rather than repeating these problems here, this section provides additional examples taken from the New York State Regents.

Proofs are again represented graphically as trees. The nodes are goals. The edges are labeled with the name and justification of the strategy that satisfied the goal. The top level objective of the problem is represented as the top of the tree. The theorem proving process proceeds basically in a top-down fashion.

PROBLEM: 32, January 64 Regents

STATEMENT: In quadrilateral ABCD, AB=AD and BC=DC. Diagonal AC is extended through C to E and lines BE and DE are drawn. Prove that BE=DE.

DIAGRAM: The program does not process pictures. The diagram is specified by giving Cartesian coordinates for the points and a list of the line segments. For the illustrative purposes of this section, only the resulting picture will be shown.



PROBLEM 32, JAN 64

HYPOTHESES: SEG AB = SEG AD
SEG BC = SEG DC

PROVE: SEG BE = SEG DE

COMMENTS: This problem is no more complex than G-1. The difference is only a recursive application of the triangle congruence expert. Repetitive application of a given line of attack does not confuse a machine. The program is an excellent bookkeeper. Far more serious are problems that require qualitatively different approaches.

PROOF: BY BTP

BE = DE

EQSEG1 — Corresponding sides of congruent triangles

triangle EBA = triangle EDA

EQTRI2 — Side-angle-side

BA = DA (HYPOTHESIS)

angle BAE = angle DAE

AE = AE (IDENTITY)

EQANGLE4 — Corresponding angles of congruent triangles

triangle BAC = triangle DAC

EQTRI1 — Side-side-side

BA = DA (HYPOTHESIS)

BC = DC (HYPOTHESIS)

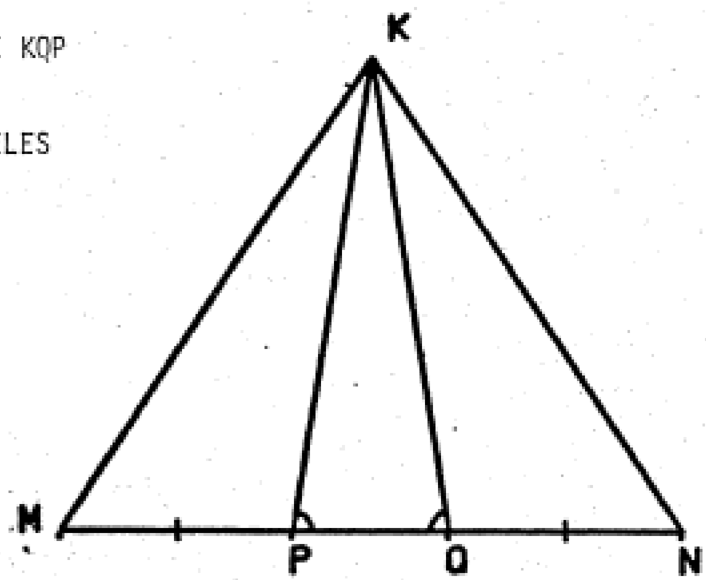
AC = AC (IDENTITY)

PROBLEM: 32, June 69 Regents

STATEMENT: Given segment MPQN with MP=QN and angle KPQ = angle KQP as shown in the accompanying figure. Prove that triangle KMN is an isosceles triangle.

HYPOTHESES: SEG MP = SEG QN
ANGLE KPQ = ANGLE KQP

PROVE: TRIANGLE KMN IS ISOSCELES



PROBLEM 32, JUNE 69

COMMENTS: For this problem, explicit knowledge about the properties of isosceles triangles has been erased. This forces the program to prove that equal base angles imply equal sides. It does this via the amusing route of proving the triangle congruent to itself under certain pairing of vertices. This attack is called the "pons asinorum".

Such a proof, involving the artifice of proving a triangle congruent to itself, is in fact more easily accomplished by a dumb theorem prover than a person. A theory for why this is so is that a person considers a triangle to be the same under all of the permutations of its name [a valuable heuristic]; whereas the computer is not bright enough to realize that the two permutations are related. It will probably be the case that as programs acquire more insight, they will less easily derive this style of proof.

This proof also involves the first example of an indirect strategy. EQSEG3 is used to prove segment equality on the basis of differences of equal line segments being equal.

PROOF: BY BTP

triangle KMN is isosceles

Conversion | Definition

KM = KN

EQSEG1 | Corresponding sides of congruent triangles

triangle KMQ = triangle NKP

EQTRI2 | Side-angle-side

MQ = PN

EQSEG3 | Arithmetic

PQ = QP (IDENTITY)

MP = QN (HYPOTHESIS)

KQ = KP

angle KPN = angle KQM (HYPOTHESIS)

EQSEG1 | Corresponding sides of congruent triangles

triangle KQP = triangle KPQ

EQTRI3 | Angle-angle-side

angle QKP = angle PKQ (IDENTITY)

PQ = QP (IDENTITY)

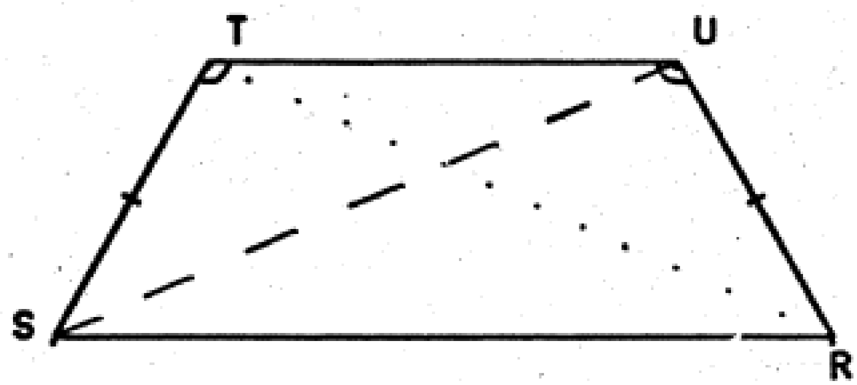
angle KPQ = angle KQP (HYPOTHESIS)

PROBLEM: 32, June 68 Regents

STATEMENT: In quadrilateral TURS, UR = TS and angle R = angle S.
Prove that angle U = angle T.

CONSTRUCTION LINES

GENERATED BY PTP



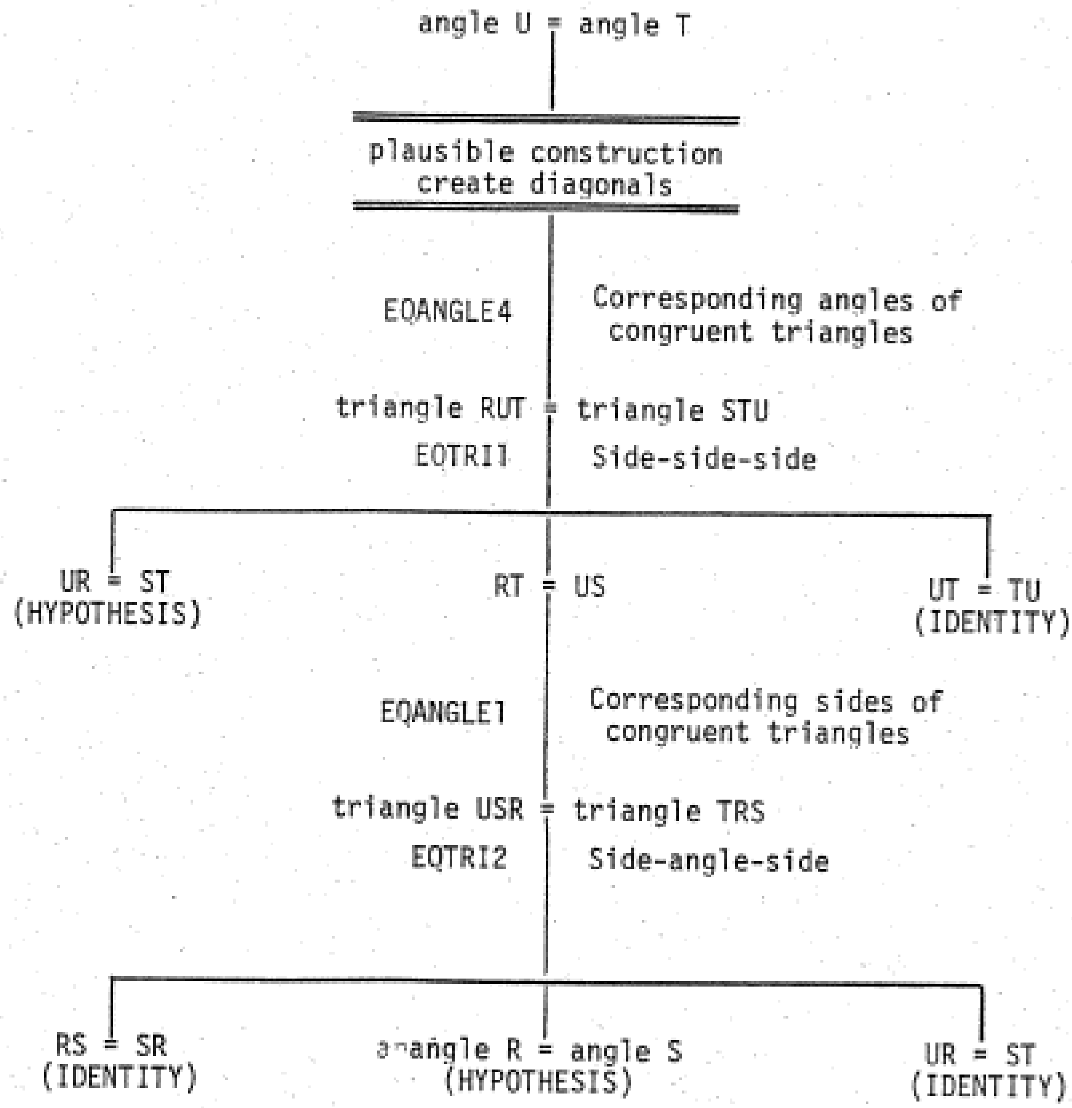
PROBLEM 32, JUNE 68

HYPOTHESES: SEG UR = SEG TS
ANGLE R = ANGLE S

PROVE: ANGLE U = ANGLE T

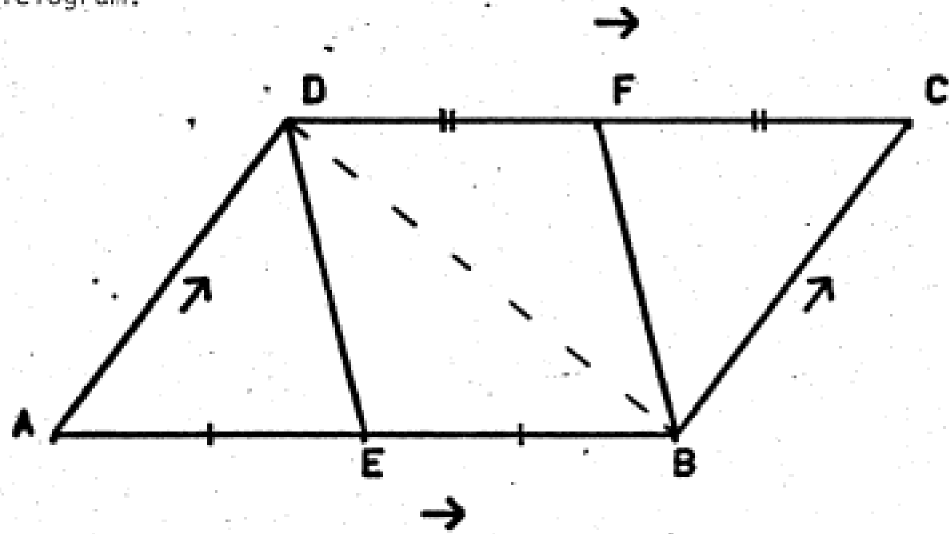
COMMENTS: Just as for parallelograms, an important strategy for relating parts of a quadrilateral is to prove congruence of the triangles formed by one or both diagonals. The difference for quadrilaterals is that the constructor must be more cautious. Corroborative evidence supporting the addition of new lines should be present. For this problem the fact that two hypotheses relate corresponding parts of the created triangles is ample justification for the constructions.

PROOF: BY PTP (SIMULATION)



PROBLEM: 34, January 65 Regents

STATEMENT: Given parallelogram ABCD with E the midpoint of AB and F the midpoint of CD. Lines DE and FB are drawn. Prove that DEBF is a parallelogram.



PROBLEM 34, JAN 65

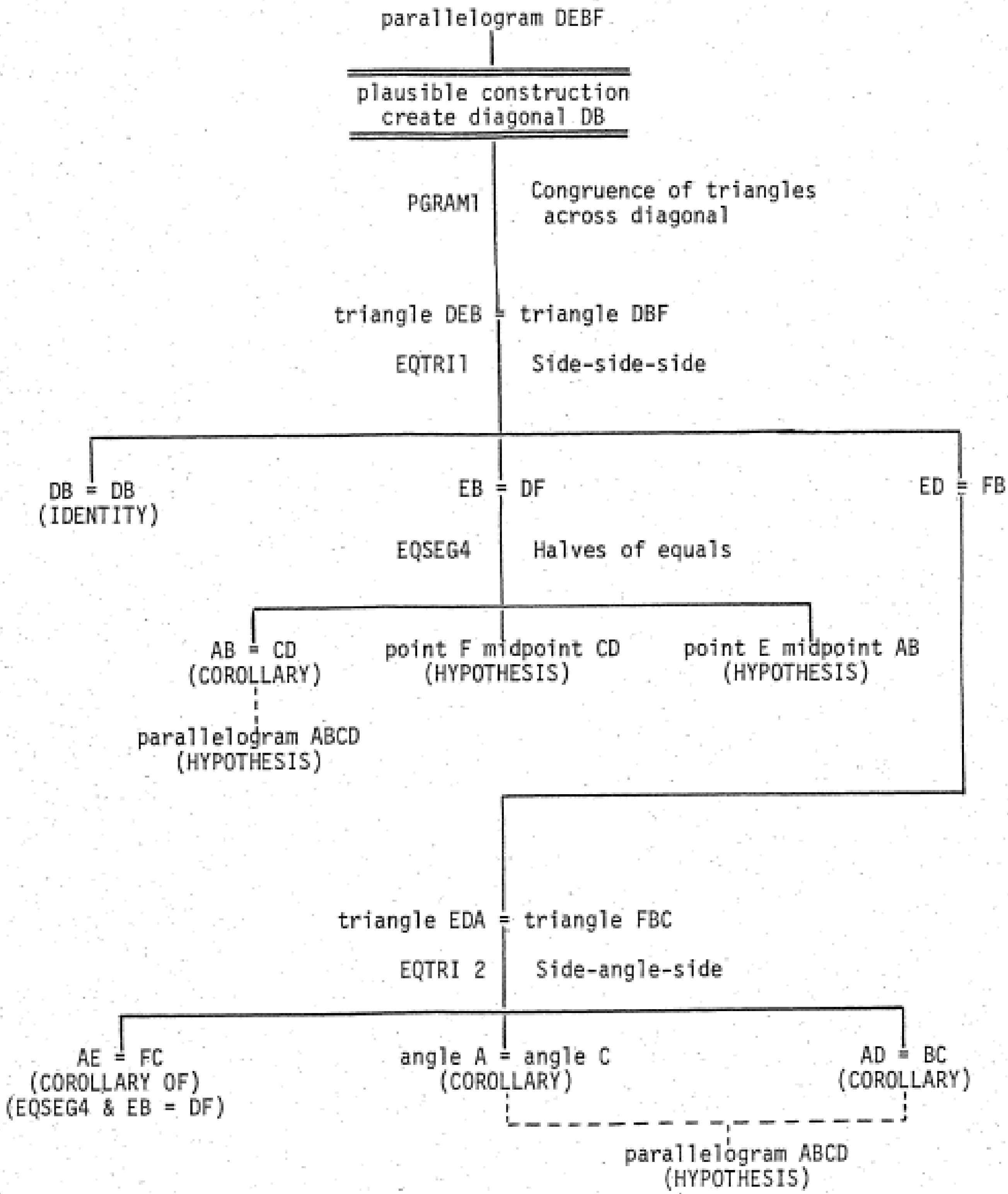
HYPOTHESES: PARALLELOGRAM ABCD
POINT E MIDPOINT SEG AB
POINT F MIDPOINT SEG CD

PROVE: PARALLELOGRAM DEBF

COMMENTS: This problem illustrates the use of antecedent theorems. Various corollaries of ABCD being a parallelogram are deduced immediately upon reading this hypothesis. The fact that the deduction from corollary to hypothesis is not part of the basic process of chaining backwards is indicated by a dotted line being used to connect the two.

As pointed out in Section III, the basic tool for proving a quadrilateral to be a parallelogram is to construct the diagonal and use triangle congruence. This plausible construction is done immediately.

PROOF: BY PEP (SIMULATION)

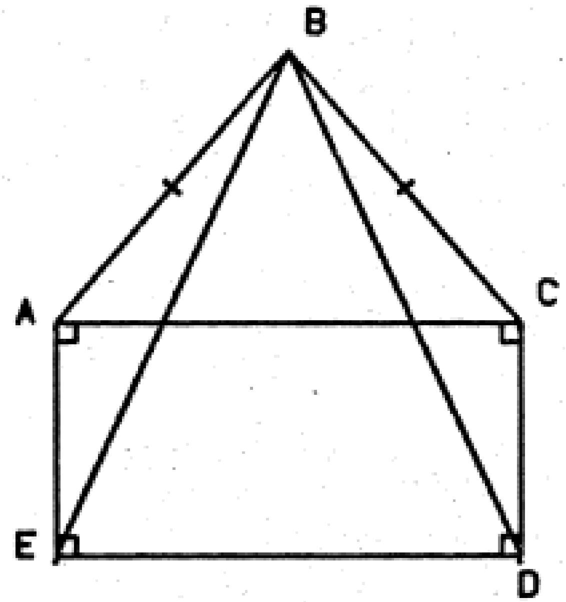


PROBLEM: 32, January 68 Regents

STATEMENT: In the accompanying plane figure, AEDC is a rectangle and AB=BC. If BE and BD are drawn, prove that angle BED = angle BDE.

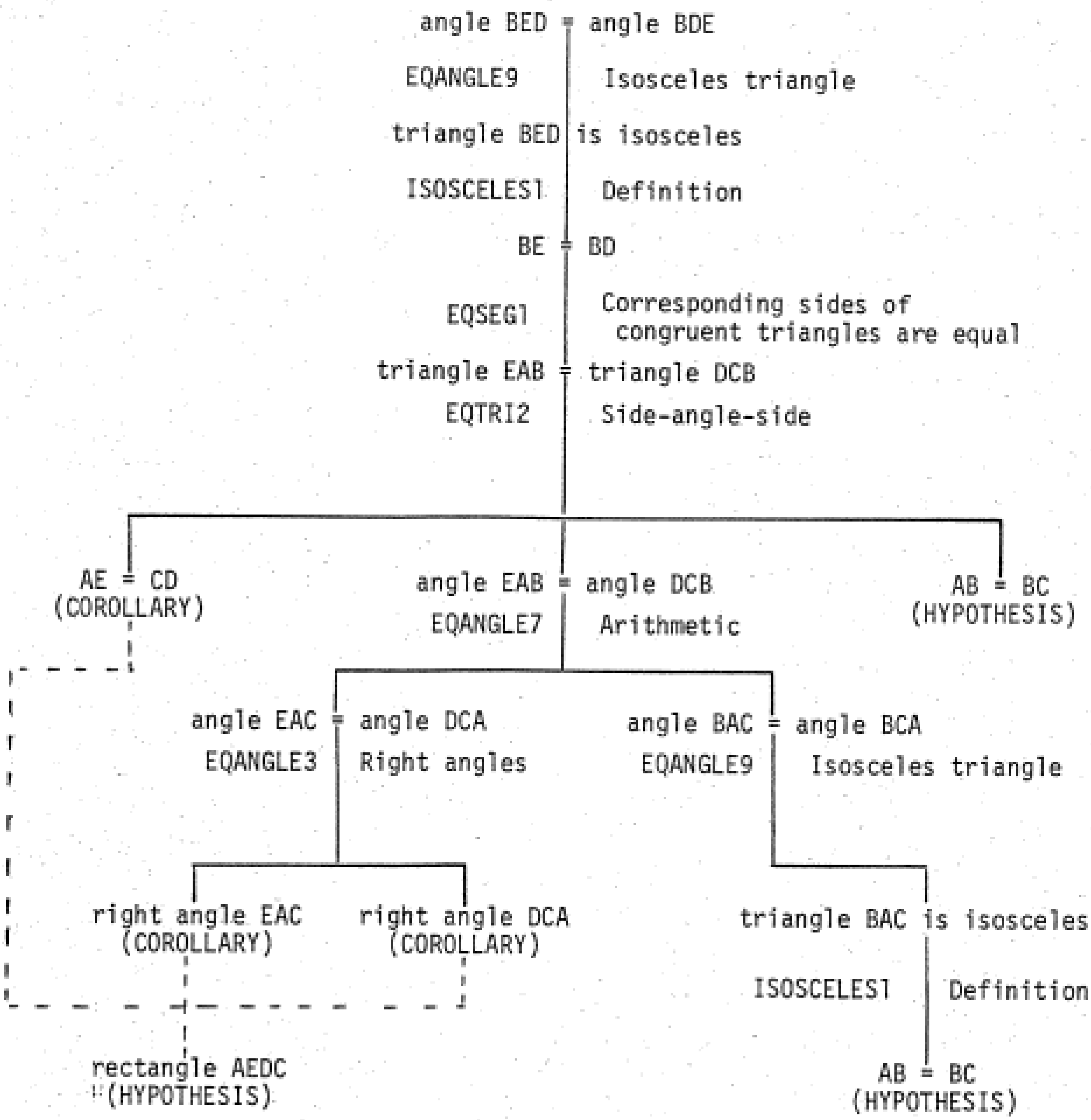
HYPOTHESES: RECTANGLE AEDC
SEG AB = SEG BC

PROVE: ANGLE BED = ANGLE BDE



PROBLEM 32, JAN 68

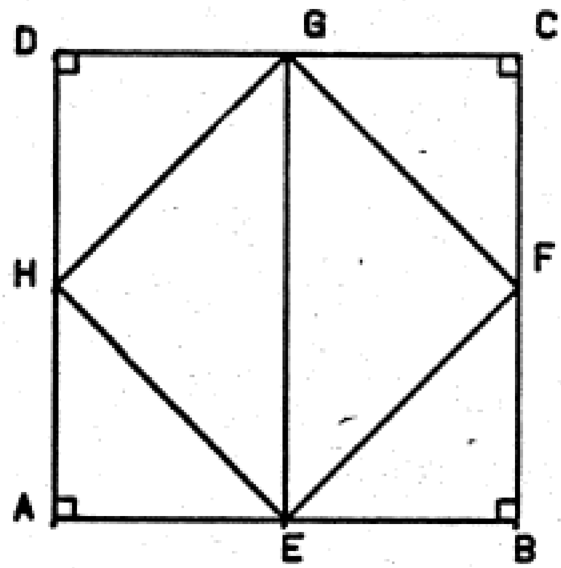
PROOF: BY BTP


angle BED = angle BDE
EQANGLE9
Isosceles triangle
triangle BED is isosceles
ISOSCELES1
Definition
BE = BD
EQSEG1
Corresponding sides of congruent triangles are equal
triangle EAB = triangle DCB
EQTRI2
Side-angle-side
AE = CD
(COROLLARY)
angle EAB = angle DCB
EQANGLE7
Arithmetic
AB = BC
(HYPOTHESIS)
angle EAC = angle DCA
EQANGLE3
Right angles
angle BAC = angle BCA
EQANGLE9
Isosceles triangle
right angle EAC
(COROLLARY)
right angle DCA
(COROLLARY)
triangle BAC is isosceles
ISOSCELES1
Definition
rectangle AEDC
(HYPOTHESIS)
AB = BC
(HYPOTHESIS)

PROBLEM: 32, January 69 Regents

STATEMENT: Given square ABCD with E, F, G, and H the midpoints of AB, BC, CD, and DA respectively. Prove:

a. Triangle EFG = Triangle EGH
b. If AB = 6, find the
    1. Length of EF
    2. Area of triangle EFG.



HYPOTHESES:
SQUARE ABCD
POINT E MIDPOINT SEG AB
POINT F MIDPOINT SEG BC
POINT G MIDPOINT SEG CD
POINT H MIDPOINT SEG AD

PROVE: TRIANGLE EFG = TRIANGLE EHG

COMMENTS: The theorem prover does not attempt questions on area. However, its diagram filter embodies a fairly substantial model for Cartesian geometry.

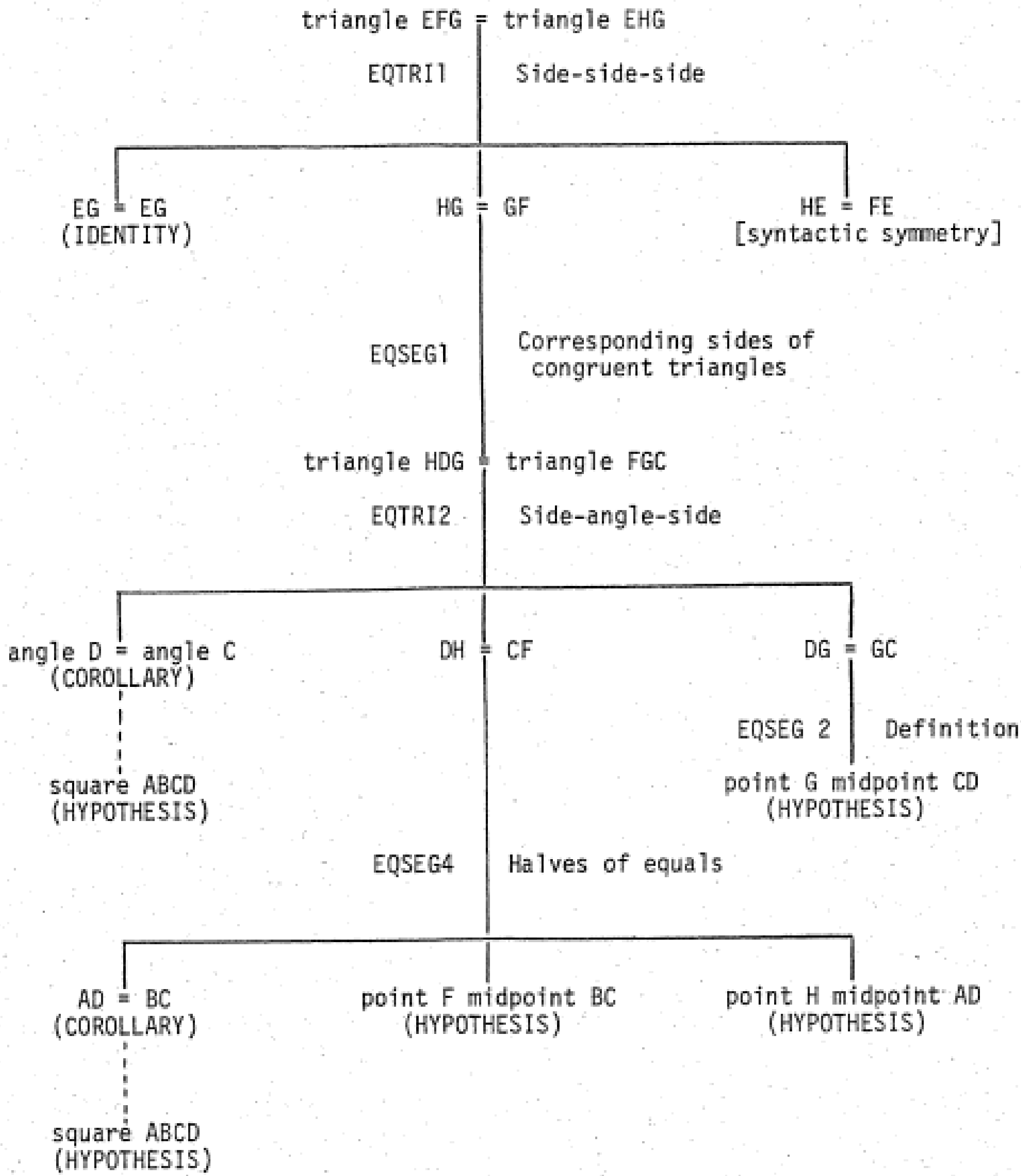
Gelernter used the phrase "syntactic symmetry" to refer to goals that are identical to previously established lemmas under some permutation of the points of the problem. He used a test for such equivalences to avoid repeating symmetric proofs. The theorem prover developed in this paper does not check for syntactic symmetry for several reasons:

a. There is no increase in power. The theorem prover is certainly capable of repeating a given line of attack on the symmetric goal. Indeed, the proof tree given is an abbreviation of the actual one generated. The skeleton of the subtree for the first segment equality (HG = GF) would be duplicated for the second (HE = FE).

b. The test for syntactic symmetry is costly, and therefore not worth the expense.

c. It is too syntactic. A more semantic approach is the observance of various geometric symmetries such as reflections across lines and points. These symmetries can be used to help plan the proof. [Eubank].

PROOF: BY BTP


triangle EFG ≡ triangle EHG
EQTRI1
Side-side-side
EG ≡ EG
(IDENTITY)
HG ≡ GF
HE ≡ FE
[syntactic symmetry]
EQSEG1
Corresponding sides of congruent triangles
triangle HDG ≡ triangle FGC
EQTRI2
Side-angle-side
angle D ≡ angle C
(COROLLARY)
square ABCD
(HYPOTHESIS)
DH ≡ CF
DG ≡ GC
EQSEG 2
Definition
point G midpoint CD
(HYPOTHESIS)
EQSEG4
Halves of equals
AD ≡ BC
(COROLLARY)
point F midpoint BC
(HYPOTHESIS)
point H midpoint AD
(HYPOTHESIS)
square ABCD
(HYPOTHESIS)

V. DIRECTIONS FOR FURTHER RESEARCH

The procedural model for geometry presented in the preceding pages is only a first step in understanding the reasoning and knowledge of mathematics. There is much more to do. The following paragraphs describe various directions that future research might pursue.

1. Global Planning

Planning, plausible moves and constructions as presented in the elementary model all have a local nature. They are tied to knowledge about particular strategies. An alternative approach would be to plan a line of attack from global considerations. This might include adding constructions on the basis of symmetry considerations [Wong] and considering different proof skeletons as, for example, proof by contradiction, hypotheticals, symmetry or inequalities.

Better global planning could use more sophisticated models for relevance, cost and provability. This might include using the diagram to help plan the proof. A model based on the diagram as a physical structure might be useful. For example, would the fact that the hypotheses imply a mechanical linkage between parts of the diagram be evidence for a geometric relationship being provable of those parts?

2. Diagram Generation

The diagram is an important computational filter. Accidentally choosing Cartesian coordinates that allow non-essential properties in the diagram weakens this filtering. A diagram generator that creates a general instance would prevent this. However, such a generator cannot be built in isolation from the basic theorem prover. There must be an interactive relationship between a generator and prover [Price]. Exploring this interaction should result in a more intelligent program.

3. Quantitative Increase in Mathematical Knowledge

How far can the basic techniques discussed so far be extended? Can the same basic structure, for example, be used to represent Euclidean knowledge about circles? What new issues arise when the amount of knowledge in the theorem prover grows large? Must it resort to a big switch structure or can all of its consequent theorems be active at the same time by virtue of the difference in their calling patterns? Is the knowledge grouped into a tree of mini-worlds with well-defined entry conditions?

4. Qualitative Increase in Mathematical Knowledge

Can the basic structure used so far be extended to other areas of mathematics, such as three dimensional geometry, topology, non-Euclidean geometry, and compass and straight-edge constructions?

5. Learning

It is only possible to consider the problem of learning after a variety of representations for the end product of such a process have been considered. Having constructed a model for geometry, we are only now in a position to consider the processes by which it was built.

One type of learning is met when the theorem prover finds a proof. The question that then arises is whether the new theorem should be added explicitly. Moreover, should it be added as an antecedent or consequent theorem?

6. English

It should be straightforward to extend Winograd's language system to process English statements of geometry problems.

7. 100% on Regents

What would it take for the theorem prover to be able to get 100% on the New York State Regent's exam? The syllabus contains such diverse topics as:

- Euclidean knowledge of
  - angle and segment equalities
  - circles
  - area
  - regular polygons
  - similar triangles
  - inequalities
- Cartesian knowledge
- right triangle trigonometry
- loci
- constructions
- logic
- algebraic manipulation

8. Alternative Approaches to Mathematical Reasoning

Some approaches to theorem proving such as "resolution" do not provide much mathematical insight. Essentially combinatorial, they take advantage of the speed of a digital computer and sacrifice deep understanding of the mathematical domain. The procedural model built in the preceding pages has the characteristic of being a natural one in the sense that it is easily understood by anyone familiar with geometry. Its naturalness should make it amenable to sophisticated planning, thereby becoming more powerful than the standard combinatorial programs. However, this has yet to be proved!

9. Psychology of Human Problem Solving

The program constructed, because of its "reasonable" nature, is a candidate for a model of human problem solving. Being explicit, it offers far more precision for psychological analyses than traditional introspective or discursive techniques. Witness such vague analyses as those of Poincare and Hadamard. It might be interesting to attempt a Newell-Simon style of psychological research into mathematical creativity using the model presented in this paper as opposed to a production system.

10. Education

A procedurally oriented study of mathematics provides an alternative approach to learning Euclidean geometry.

1. Theorem proving approaches mathematics as something to do.
2. It provides a model of problem solving for the student.
3. The student, himself, develops this model.
4. The basic constraint is not the teacher's whim but the actual performance of the program.

For a student, building a procedural model is like creating a scientific theory. The individual looks for basic pieces of knowledge that can support actual problem solving. Considerations of logical consistency or completeness are secondary. The first task is to construct a model that begins to work on some interesting set of problems.

The particular knowledge chosen is a personal choice. An individual begins with that knowledge with which he is most comfortable. Arbitrary claims that a given set of axioms capture the essence of geometry are rejected.

VI. BIBLIOGRAPHY

Eubank, Porter. "The Geometry Strategist: A Program for Proving Geometry Theorems." Unpublished paper. August 30, 1972.

Eubank, Porter and Jacobson, Mike. "Geometric Theorem Proving and Line Construction." Unpublished paper. May 22, 1972.

Feigenbaum, Edward A., and J. Feldman. COMPUTERS AND THOUGHT. McGraw-Hill, New York, 1963.

Gelernter, H. "A Note on Syntactic Symmetry and the Manipulation of Formal Systems by Machine," Information and Control 2 (1961), 80-89.

Gelernter, H. "Realization of a Geometry-Theorem Proving Machine," in Feigenbaum and Feldman (eds.), COMPUTERS AND THOUGHT, 134-152.

Gelernter, H. et. al. "Empirical Explorations of the Geometry-Theorem Proving Machine," in Feigenbaum and Feldman (eds.), COMPUTERS AND THOUGHT, 153-163.

Gilmore, P.C. "An Examination of the Geometry Machine." Artificial Intelligence 1 (1970), 171-187.

Hadamard, Jacques. THE PSYCHOLOGY OF INVENTION IN THE MATHEMATICAL FIELD. Princeton University Press, 1945.

Poincare, Henri. "Mathematical Creation" in THE FOUNDATIONS OF SCIENCE. Trans. by G. Bruce Halsted. The Science Press, New YOrk, 1913, p. 387.

Price, Keith. "Satisfying Geometric Constraints." Bachelor's Thesis in Electrical Engineering, M.I.T., May, 1971.

Wong, Richard. "Construction Heuristics for Geometry and a Vector Algebra Representation of Geometry." Master's Thesis in Electrical Engineering, M.I.T., May, 1972.