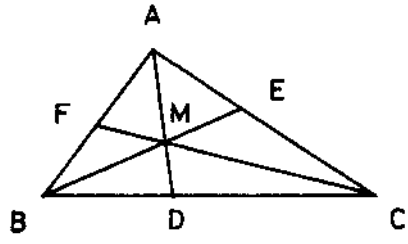
GENERATION OF DESCRIPTIONS FOR LINE DRAWINGS

A. H. Dixon
Assistant Professor
Department of Computer Science
The University of Western Ontario
London, Canada N6A 5B9

A line drawing is often represented by a graph with the two primitives, edges and vertices, corresponding to lines and their intersections. Both primitives usually have additional properties associated with them. The problem is to generate a concise description such that it is the most appropriate among a set of alternate descriptions and also that it is complete -- all observable properties and relations in the diagram are given by or derivable from statements in the description. The concept of "appropriateness" is a qualitative one, to be measured by comparison with the responses of humans when generating descriptions.

It is obvious that knowledge of the domain from which the diagrams will be chosen can affect the formation of a description. Such knowledge certainly includes objects and relations which can be constructed from the primitives. More importantly, however, some instances of this knowledge take the form of contextual properties within the diagram. As an example, the domain of Euclidean geometry diagrams provides such additional information in at least two ways. Consider the following diagram.



One possible description corresponds to its initial internal representation as a graph with seven labelled vertices and twelve edges. Exploiting knowledge about possible constructs could include looking for triangles and might yield: "triangles AME, AMF, BMF, BMD, CMD, CME". This description does not exhaust the possible triangles that may be found, nor would one wish a description which included explicit reference to all of them. Rather, they should be deducible from the description, in this case possibly by exploiting collinear relationships which themselves are deducible from the coordinate positions of the points.

The diagram can also be described as "triangle ABC, with cevians AD, BE, CF intersecting at point M". Among practising geometers, this is likely to be a popular choice. But what contextual information is available for directing the analysis toward this description? The fact that the triangle in the diagram is labelled "ABC" and that it is explicitly referred to in the description suggests that labelling plays an important role. It draws attention to particular parts of the diagram for special scrutiny. This is so only because certain conventions have been adopted to minimize the difficulty of identifying salient features. Of course, the interpreter must be aware of the conventions, and this information is included in his corpus of knowledge about Euclidean geometry

While labelling provides important cues in affecting the formation of a description, selective orientation of parts of the diagram can also be used to capture attention. The main point is that there is a set of rules which constrain the way diagrams are presented. These conventions, if embedded in a knowledge base, can influence which of several possible interpretations is generated.

Using an Euclidean geometry textbook as a source for diagrams, a set of heuristics corresponding to some of the conventions followed by that text has been implemented in a system for generating descriptions. The rules form a heterarchy of experts so that the effect of rule omission on description quality can be conveniently assessed. Pre-conditions for elaboration of some rules depend on a partially completed description. As this description is a function of which rules were used to generate it, the effect of the order of application of rules can also be observed.

To facilitate these design considerations, the system was programmed in a PLANNER-type language. The content of an associative database reflects the current description of a diagram during execution. The description is modified by insertions into the database which have the side-effect of invoking antecedent theorems. These theorems in turn control the formulation of further refinements to the description including the removal of redundant information. Execution terminates when no further changes can be made. The final description is the collection of insertions still in the database.