ABSTRACT.

A number of authors have presented algorithms for various graph-theoretic properties, and some of these algorithms have been programmed for digital computers. These include algorithms for planarity [F3 , S6], and graph isomorphism [P1 , C1], among others.

In all of the cases cited, an algorithm has been programmed to examine some particular property of a graph.

The Graph-Theoretic Programming Language - GTPL - departs from this approach and provides a language facility in which a wide variety of properties of a graph can be examined or tested.

As far as we are aware, the only previous attempt in this direction is the work of Tabory [T1], who has added a limited number of graph theory subroutines to a FORTRAN system.

Unlike Tabory's "Graph Processor", the present work is a genuine extension of FORTRAN. Statements have been added to the FORTRAN system which

a. compute various characteristics of a graph,

b. derive a graph from a given graph,

c. perform operations on a graph so as to modify the given graph, and

d. test whether or not a graph exhibits some particular property.

The programming system which we describe, has been implemented on an IBM 1620, and two of the programs which have been compiled and executed by the system, are included as examples.