APRIL EUCLID TO PASCAL TRANSLATOR

by

DAVID PAUL ROESENER

B. S., Kansas State University, 1979

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1980

Approved by:

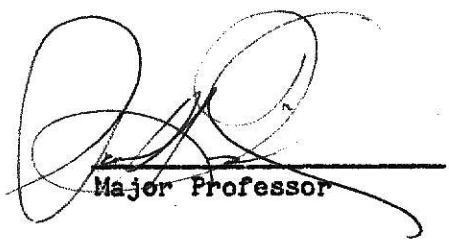

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to thank my major professor, Dr. Rodney Bates, for all the input he has made to this project. He has cleared up my misunderstandings, given me ideas about the translation, read this paper over too many times, and in general has put up with me graciously over the past six months.

CHAPTER 1

INTRODUCTION

1.1 PURPOSE

The purpose of this project is to install a version of Euclid, a relatively new programming language, in the Kansas State University Computer Science computer laboratory. The machine selected for this installation project is the Interdata 8/32, a minicomputer with a functioning Pascal compiler. The Euclid available for this project is an April Euclid compiler, written in April Euclid. The compiler creates PDP-11 object code as its target language designed to run under the UNIX or 11/UNIX operating system.

One solution to the installation problem would be to design a translator from April Euclid to the compilable Pascal. This course of action was selected and the translator created is discussed in this paper.

Before discussing the translator, however, this paper will briefly discuss the Euclid programming language, and the overall installation process.

1.2 EUCLID

April Euclid is a subset of Euclid, developed by B. W. Lampson, J. J. Horning, R. L. Landon, J. G. Mitchell, and G. J. Popek. Its

description appears in the "Report On The Programming Language Euclid" in the February 1977 issue of SIGPLAN Notices [1]. The compiler acquired for this project was developed by the joint effort of the Computer Systems Research Group of the University of Toronto and I. P. Sharp Associates Ltd., both of Toronto, Canada. Its description appears in the documentation accompanying the compiler [2]. Euclid is a language similar to Pascal [3], but carries the philosophy of Pascal further. Euclid has strict type checking, as does Pascal, but also has parameterized type declarations. That is, one type can be used to declare many variables of types differing only by a constant. For example, all variables whose type is a character array can be declared using:

```
type CHARARRAY (lower, upper) =
    lower .. upper of char
```

Euclid's variables and types also have implied characteristics which can be referenced in the program. For example, THIS_VARIABLE.Size returns the number of bytes the variable uses in machine code. Euclid also includes ASSERT statements that aid in run-time program proving. The compiler generates a run-time check that produces an error if the ASSERT statement is false. Other features of Euclid are mentioned later in this paper in the discussions of translation techniques.

April Euclid has fewer features than Euclid. In Euclid, modules have both an initialization procedure and a finalization procedure. April Euclid has only the initialization procedure. It also does not allow for parameterized types. Euclid also permits variables to contain sets of objects. April Euclid does not allow this.