EXECUTION MODELS FOR TRANSLATOR DESIGN

BY

MILES T. CLEMENTS JR.

B.S., NORTH GEORGIA COLLEGE, 1965

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

1977

APPROVED BY:

MAJOR PROFESSOR

TABLE OF CONTENTS

Chapter Page

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

THIS BOOK CONTAINS NUMEROUS PAGE NUMBERS THAT ARE ILLEGIBLE

THIS IS AS RECEIVED FROM THE CUSTOMER

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERANTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

Chapter 1

INTRODUCTION

The fundamentals for this report were covered in a Translator Design course taught by Dr.Bill Hankley in the summer of 1975 and summer of 1976. The CS700 stack machine was designed and implemented by the 1975-1976 Translator Design classes. This report combines class presentations and existing CS700 documentation to explain the stack concept and implementation.

An overview of stack machines and the CS700 instruction set is presented to describe the environment. A diagram of an interpreter is shown to graphically show this reports' relationship to other CS700 reports. The tagged architecture is explained in chapter 2 by representing the execution stack as a data structure. Execution models are presented in chapter 3 for selected instructions. Indexing and I/O are not included. A description of a design specification language and formal algorithms for the selected instruction set are included as appendixes.

Stack Machine

A basic stack machine is characterized by module calls which are dynamically linked at the point of call. A given