RECOGNITION OF IDENTICAL STUBS IN A DECISION TABLE PROCESSOR

By

CHI-DONG LU

B.S., National Taiwan University, 1968

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

1973

Approved by:

Kenneth Conrow

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A decision table is simply a tabular form for expressing conditional logic. It shows the relationship between conditions, the values of those conditions and the associated actions.

The use of decision tables in computer source programs and the development of processors for converting these decision tables to standard source programs has occurred over the past few years. For example, the Census Bureau and Working Group 2 of the Special Interest Group for Programming Language (of the Los Angeles Chapter of ACM) have each independently developed a preprocessor to convert decision tables written in COBOL to standard COBOL statements and paragraphs. These can be converted to a computer object program by a COBOL compiler.

Recently, Smith and Conrow are developing the DECTAL processor to convert decision tables written in PL/I to standard PL/I statements, which can be translated to a computer object program by a PL/I compiler.

Due to the complex logic in decision making, a group of linked decision tables are used for segmenting the logic shown in decision table form. A key feature in DECTAL is the ability to reference condition or action stubs anywhere in a block of decision tables from any other point in the block. Currently it is the programmer's responsibility to supply cross references if he wishes to effect the economy in compile time and execution module size which a cross reference implies. Automatic recognition of identical stubs and provision of the cross references while still providing the desired economies.

The program developed in this work is written to be incorporated

into the DECTAL processor to do the automatic recognition of identical stubs. The cross-reference table already existing in DECTAL can be modified for identical stubs recognized as the result of this program.

II. THE STRUCTURE OF A DECISION TABLE

The general basic structure of a decision table is shown in Figure 1. There are four sections or quadrants usually separated by double lines but often in other ways, e.g. single line, thick line. The sections set our respectively: the full set of conditions applicable (condition stubs), the full set of actions applicable (action stubs), the different combinations of those conditions(condition entries) and the corresponding combinations of actions (action entries). Each combination of conditions and associated actions forms a decision rule set out in parallel to all the other decision rules.

Within this general structure, several different types of decision tables can be defined. One classification is based on whether or not the rules are set out in columns or rows; the former is termed "vertical rule" and the latter "horizontal rule".

A more important difference lies, however, in the extent to which conditions and actions are defined in the stub. Where conditions and actions are wholly specified in the stub (as illustrated in Figure 2), the table is termed "limited entry". Entries are limited to noting the status of particular conditions and actions in a specific rule.

In those cases where conditions and actions are generally identified in the stub, with specific values shown in the entries, the table is termed "extended entry". Figure 2 illustrates horizontal rule as well as extended entry.

The last important difference lies in the way in which the conditions are linked. The conditions can be linked by the connector AND, OR or MIXED AND/OR. In Figure 2 and 3 they are linked by connector AND.

DECTAL handles extended entry decision tables by converting them to limited entry decision tables, so the program described here concerns itself solely with recognition of identical stubs in a limited entry decision table.

III. CROSS REFERENCES

It would be unrealistic to define logic in all cases in a simple table, so a group of tables is desired to express the required logic. A hypothetical example of the linkage between tables in a flowchart is illustrated in Figure 4.

As a result of using a number of tables for segmenting the logic, a table and a stub need to be identified. The symbol Txx is recognized to denote the heading of a table (e.g. T40 means table forty). In the same manner the symbol A(or C)xx designates a specific Action (or Condition) stub in a certain table in the DECTAL processor. The combination of these two symbols can completely identify the position of a stub in the set of tables. Using these identifying symbols, the DECTAL processor constructs a cross-reference table for all the stubs with the Form *Txx (A|C)xx to convey programmer-supplied referencing of the identical stubs among the tables.

This program groups all the identical stubs together and could be incorporated into the DECTAL processor to modify the cross reference table for identical stubs.