ANALYZING CHANGES IN COBOL PROGRAMS DURING MAINTENANCE

by

IE-HONG LIN

B.S., NATIONAL CHANG KUNG UNIVERSITY
Tainan, Taiwan  1980

M.S., KANSAS STATE UNIVERSITY
Manhattan, Kansas  1985

A MASTER'S THESIS

submitted in partial fulfillment of the
requirement for the degree

MASTER OF SCIENCE

Department of Computing and Information Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas
1988

Approved by:

Major professor
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Tables and Figures</td>
</tr>
<tr>
<td>Chapter 1. Introduction</td>
</tr>
<tr>
<td>1-1 Definition and Importance of Software Maintenance</td>
</tr>
<tr>
<td>1-2 Objectives and Scopes</td>
</tr>
<tr>
<td>Chapter 2. Data Collection</td>
</tr>
<tr>
<td>2-1 The COBOL Programs being Analyzed</td>
</tr>
<tr>
<td>2-2 Measurement on Types of Statement</td>
</tr>
<tr>
<td>2-3 The Changes; Altered, Deleted, and Added statements</td>
</tr>
<tr>
<td>2-4 Tools for Analyzing Data Programs</td>
</tr>
<tr>
<td>Chapter 3. Result of Analysis</td>
</tr>
<tr>
<td>3-1 Program Characteristics</td>
</tr>
<tr>
<td>3-2 The Changes</td>
</tr>
<tr>
<td>3-3 Types of Maintenance</td>
</tr>
<tr>
<td>3-4 Precise Rules</td>
</tr>
<tr>
<td>3-5 Result and Discussion</td>
</tr>
<tr>
<td>Chapter 4. Classification</td>
</tr>
<tr>
<td>4-1 Data Programs</td>
</tr>
<tr>
<td>4-2 Result</td>
</tr>
<tr>
<td>4-3 Validation</td>
</tr>
<tr>
<td>Chapter 5. Conclusion and Recommendation</td>
</tr>
<tr>
<td>References</td>
</tr>
</tbody>
</table>
Appendix A. The Shell Program Maintain ............... A-1
Appendix B. Example COBOL Program Version.1 ........ B-1
Appendix C. Example COBOL Program Version.2 ........ C-1
Appendix D. Result from Running Maintain ............... D-1
Appendix E. The Shell Program Classify ................. E-1
Appendix F. Result from Running Classify ............... F-1
LIST OF TABLES AND FIGURES

Table 1. Number of Versions and Average Number of Lines of Program Set A ............... 7
Table 2. Numbers of Statements of Program Set A ..... 17
Table 3. Numbers of Altered Statements in 28 Deltas 19
Table 4. Numbers of Deleted Statements in 28 Deltas 20
Table 5. Numbers of Added Statements in 28 Deltas ... 21
Table 6. Classification of Program Set A .............. 31
Table 7. Classification of Program B-1 ................ 40
Table 8. Classification of Program B-2 ................ 41
Table 9. Classification of Program B-3 ................. 42
Table 10. Classification of Program B-4 ............... 43
Table 11. Classification of Program B-5 ............... 44
Table 12. Classification of Program B-6 ............... 45
Table 13. Classification of Program B-7 ............... 46
Table 14. Classification of Program B-8 ............... 47
Figure 1. Flow Chart of the Shell program Maintain... 10

Figure 2. Numbers(TYPE) of Program Set A ............. 18

Figure 3. Numbers of Delta on Types Maintenance of Program Set A .................... 33

Figure 4. Percentages of Occurrence on Each Type of Maintenance of Program Set A ............. 34

Figure 5. Percentages of Occurrence on Each Types of Maintenance of Program Set B ............. 48
Acknowledgments

I sincerely appreciate the knowledgeable assistance that was given to me by my major professor, Dr. David A. Gustafson. A special note of thanks is extended to my parents for their help, understanding, and guidance. Also, I am truly grateful to all my friends who express their moral support.
Chapter 1. Introduction

1-1 The Definition and Importance of Software Maintenance

Software maintenance is the final phase of the software life cycle. It is frequently viewed as a phase of lesser importance than the design and development phases. The definition of software maintenance is the performance of those activities required to keep a software system operational and responsive after it is released for use [Liu76]. The software maintenance activities modify a program to generate new output, to change the logic to incorporate a new feature, to expand functions, to add new files, etc [Liu76]. Generally, software maintenance covers not only changes to source code but also changes to specification and design notation. The reasons to perform software maintenance are to correct error and design defects, to improve the design, to convert the program to meet more advanced features, to interface the program to other programs, and to satisfy users' demands.

Maintaining an application software system tends to consume a major portion of the total life cycle costs. Statistical data shows that maintaining 2 to 10 years old software systems demands possibly as high as 40% to 60% or even 70% of the amount of the development effort for most
companies [Lie78]. Many organizations expend approximately three-fourths of their data processing budget on maintaining existing programs. And the effort is increasing as more software is produced. Many managers are dismayed by the actual expenses on maintenance.

Although it is a complex and costly phenomenon, software maintenance remains the least understood of the software processes and receives little attention. Little research and few technical approaches or "methods" have been proposed for the software maintenance. In order to differentiate the types of maintenance and bring maintenance under control, the manager needs methods to classify different types of maintenance. Proper methods of classifying types of maintenance should help in managing the maintenance effort.

A characterization of three types of maintenance activities has been proposed by Swanson [Swa76]. The three types are corrective, adaptive, and perfective maintenances. As defined, corrective maintenance is performed to correct errors that are uncovered after the software is brought to use. Adaptive maintenance is applied to properly interface with changes in the external processing environment. Perfect maintenance is applied to eliminate inefficiencies, enhance performance, or improve maintainability based on the requests from the user group [Lie78].
1-2 Objectives and Scopes

The purpose of this research focuses on classifying different types of maintenance activities based on data obtained from analyzing COBOL programs. The classification is basically a refinement of the earlier work of Swanson. Whether the previous classification methods are good enough to distinguish the maintenance activities will be discussed and compared with a proposed method from this study.

Two sets of COBOL programs, each with several versions, were used as data programs. The first set, from a Kansas company, is named as organization A programs, or program set A, throughout the study. The second set, organization B programs or program set B, however, came from a data processing environment. A shell program was developed as a tool to analyze the differences between two consecutive versions from program set A. The result lists the numbers of each statement in the first version as well as the altered, deleted, and added statements changed from the first to the second version. The rules for classifying the types of maintenance were identified from the results and then converted into a second shell program. The input for the second shell program is the output from the first shell program.
The organization B programs were later analyzed with the two shell programs to test the results and verify the rules. For convenience, the first shell program was named Maintain and the second one as Classify.

The objectives of this research are to study real-life COBOL programs to better understand what goes on in the software maintenance phase, to develop a method of classifying types of maintenance from program set A, and to check the proposed method program set B.

Chapter 2 discusses the data collection process. Explanation of the COBOL programs and a brief description of program sets A and B are given in the first section. Section 2 gives the definition of the measurements applied to calculate the data. The changes and altered, deleted, added statements are defined in Section 3. The shell programs Maintain and Classify are described in Section 4. The shell programs are the basic implementation tools employed to analyze the data programs.

Chapter 3 gives the results from running the organization A programs. Characteristics of program set A are illustrated in tables and figures. The changes between two consecutive versions are displayed. The rules of classifying the types of maintenance are listed. The reasons for iden-
tifying the maintenance are explained in detail. The rules were then written on to shell program Classify. The last section in Chapter 3 presents the results from running COBOL programs A and discusses the insights into the maintenance of the programs.

Chapter 4 involves the verification of the results in chapter 3 by classifying the program set B. All the procedures and tools employed are the same as in Chapter 3. Program data and corresponding results are represented in table or graphic forms.

Chapter 5 concludes the study and suggests recommendations for the future work.
Chapter 2. Data Collection

2-1 The COBOL Programs being Analyzed

Why choose COBOL program to analyze? COBOL is a programming language that has been designed expressly for administrative data processing. It is a high-level language and provides efficient data collection, data processing, and production of required reports. COBOL is widely used in industry and business fields.

In Chapter 1, we mentioned the program sets A and B which are the data programs in the study. The program set A, which consists of 5 COBOL programs, was analyzed in the beginning. These programs have various numbers of versions. The number of versions are 4, 5, 6, 7 and 11, respectively. The total number of versions is 33. The lines of codes also vary quite differently. The average number of the shortest program is 270; while the value of the largest is more than 4650. Table 1 displays the number of versions and average number of lines in the 5 COBOL programs.

As stated earlier, the program set B was applied to verify the results from running the program set A. The program set B, which includes 8 COBOL programs, has 20
versions on each program. All B programs have been operational for many years.

<table>
<thead>
<tr>
<th>Program no.</th>
<th>Number of versions</th>
<th>Average number of lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>270</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1430</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>4650</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>2070</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>470</td>
</tr>
</tbody>
</table>

Table 1. Number of Versions and Average Number of Lines of Program Set A

2-2 Measures on Types of Statements

In a COBOL program, a statement is defined as a syntactically valid combination of words and characters. Measuring the numbers of statements that have been changed between two sequential versions of a program is the basic step of collecting data for the entire research. Classified by their functions, types of statement fall into 8 categories: comment, declaration, assignment, conditional, branch, input-output, label, and other statements.

The following notations are used throughout the study. The notations were devised by Dr. David A. Gustafson and the participants in a software seminar at Kansas State University.

TYPE represents the types of statements and ALL stands for the collection of all statement types.
TYPE ::= ALL | comment | declaration |
       assignment | conditional | branch |
       input-output | label | other

comment ::= spacing purposes | textual
assignment ::= MOVE | ADD | SUBTRACT | COMPUTE
conditional ::= IF | ELSE | ON | AT END
branch ::= CALL | PERFORM | GOTO | NEXT | EXIT
input-output ::= DELETE | DISPLAY | OPEN | READ |
                WRITE | REWRITE
other ::= EXAMINE | INSPECT | SEARCH | SORT |
       SET | EXEC CICIS | GOBACK

2-3 The Changes: Altered, Deleted, and Added Statements

Measuring the change to the code is an objective indicator of the maintenance process itself. Analyzing changes between versions is a good approach to investigate what types of maintenance are really made to the programs.

In reality, statements referring to changes have three different kinds: altered, deleted, and added statements. Altered statements can be meant to specified statements existing in two versions; however, a variable is different in its values, a statement is moved to "comment" statement because of putting asterisk in front of it by special purpose, or statements switched to another type based on programmer's need, etc. Deleted statements show on the origi-
nal version but are missing from the second version. Added statements are inserted to the original version.

For the convenience of notational representation, let

\[
Changes ::= \text{Altered} \mid \text{Deleted} \mid \text{Added}
\]

\[
Changes[\text{ALL}] ::= \text{Altered[ALL]} + \text{Deleted[ALL]} + \text{Added[ALL]}
\]

\[
Changes[\text{TYPE}] ::= \text{Altered[TYPE]} + \text{Deleted[TYPE]} + \text{Added[TYPE]}
\]

\[
\text{Altered[TYPE]} : \text{Number of Statements of specified TYPE that have been altered.}
\]

\[
\text{Deleted[TYPE]} : \text{Number of Statements of specified TYPE that have been deleted}
\]

\[
\text{Added[TYPE]} : \text{Number of Statements of specified TYPE that have been added.}
\]

2-4 Tools for Analyzing Data Programs

The shell program Maintain which invokes several UNIX utilities such as diff and grep, was written to analyze the COBOL programs. Six modules are included in the program; they are checking, preprocessing, distinguishing, difference, calculation, and report modules. Each module has its special function. Figure 1 gives the flow chart of the shell program Maintain. The inputs are two versions of a COBOL programs. The input sequence has to be the same order for the comparison purposes. The executing command "Maintain
Figure 1  Flow chart of the Shell program Maintain
version.1 version.2" reports the output which lists the results changed from version.1 to version.2. In reverse, the command "Maintain version.2 version.1" generates the report which contains the data modified from version.2 to version.1. The output of the program Maintain, assuming the command "Maintain version.1 version.2", includes the numbers of statements in version.1 and the overall numbers of statements in altered, deleted, and added statements from version.1 to version.2. Appendix A lists the program Maintain. Two example COBOL programs Version.1 Version.2 are given in Appendix B and C. Appendix D shows the result from running the Maintain program on the two versions. The function of each module in the program Maintain is described as below.

a. Checking module

Every COBOL program consists of four divisions in the following order: identification, environment, procedure and data divisions. The checking module checks to verify the existence of all four divisions in the two input data programs. Error messages referring to the absence of divisions are printed out in case of the missing of any division. Program terminates if errors are detected.

b. Preprocessing module

The preprocessing module removes all superfluous blank
spaces, tabs, and blank lines. If the difference of a statement in version 1 and the version 2 is only the addition or deletion of a blank space, it should not be marked as altered. The same situation can be extended to the insertion or removal of blank lines simply for spacing purposes. This module also removes numbers from tail end of lines. The numbers attached at the end of lines have no meaning. COBOL programs use it as marked symbol to easily identify a sequence of codes.

c. Distinguishing module

This module assigns unique characters to every statement in order to identify types of each particular statement. The function ensures that each statement can be properly identified after changes are made. For example, all the statements in the identification division are prefixed with "Comment". In the environment division, the statements are marked with "Env". The FILE SECTION and WORKING-STORAGE statement in procedure division are attached with "DeSetn". The rest of statements in procedure division are added with "Dl rtn".

d. Difference module

The difference module utilizes the "diff" function to find the differences between two versions. It compares two versions of program and notes altered, deleted, and added statements. Three temporary files are created once the
altered, deleted, and added statements are in existence. The files which store the deleted and added statements copy the statements from the analyzed versions. The file having the altered statements contains the old statements and the new statements. If there are several places that statements are altered, we name each place as a block of altered statements.

e. Calculation module

The calculation module computes the numbers of respective type of statements. The module generates overall numbers of types of statements for version 1. Three temporary files, if they exist, are also analyzed by this module to produce output.

f. Report module

The report module produces output for the shell program Maintain. The output includes the overall analysis of the statements in version 1. It displays the number of statements in version 1. The result also lists the numbers of statements in altered, deleted, and added statement, if they exist. The actual altered, deleted and added statements are displayed at the end of the output. It is easy to identify the statements by the use of the special characters which were added in the distinguishing module.

The algorithms for developing the shell program Classi-
are based on the rules for classifying types of maintenance. The rules and corresponding algorithms will be described in chapter 3.
Chapter 3. Result of Analysis

3-1 Program Characteristics

Program set A, consisting of 5 different programs with 33 versions, was initially introduced to be analyzed by the tools, Maintain and Classify shell programs. Table 2 illustrates the characteristics of program set A. The minimum and maximum numbers of statements are given to represent the structure of statements in each program. Figure 2 gives the programs' characteristics by means of graphic form. Of the 8 types of statements, numbers of input-output, label, and other statements are ignored because of their relatively small number compared to the rest of the five types of statements. Investigating the graph, it is easy to realize that assignment statements play an important role in program set A. However, comment statements also are significant due to their frequent occurrence.

3-2 The Changes

The shell program Maintain runs two sequential versions of a program. The results of analysis on program set A
are 28 deltas. The contents of a delta includes Number[TYPE] of each statement and Altered[TYPE], Deleted[TYPE], and Added[TYPE] between the original and new versions. In addition to these numbers, the statements being changed are also listed as part of the content of a delta. The raw data of Altered[TYPE], Deleted[TYPE], and Added[TYPE] in the 28 delta are shown on Table 3, 4, and 5, respectively.

3-3 Types of Maintenance

From the 28 deltas with attached listings of changed statements, six types of maintenance were identified; they were corrective, adaptive, retrenchment, retrieving, pretty printing, and documentation maintenance. Compared with the three classical types of maintenance proposed by Swan-son, it is clear that perfective maintenance was excluded from the classification and replaced with retrenchment, retrieving, pretty printing, and documentation maintenances. The reason of excluding perfective maintenance from classification is due to the difficulty of predicting the intention of the programmer doing the enhancement. The reason for the programmer to update sources code is too complicated to trace simply from investigating changed statements. The changes on the rest of statements may be a side effect of
<table>
<thead>
<tr>
<th>Program No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Version</td>
<td>6</td>
<td>4</td>
<td>11</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>ALL</td>
<td>270</td>
<td>278</td>
<td>1428</td>
<td>1438</td>
<td>4641</td>
</tr>
<tr>
<td>comment</td>
<td>58</td>
<td>69</td>
<td>195</td>
<td>248</td>
<td>782</td>
</tr>
<tr>
<td>declaration</td>
<td>80</td>
<td>80</td>
<td>607</td>
<td>607</td>
<td>1171</td>
</tr>
<tr>
<td>assignment</td>
<td>51</td>
<td>51</td>
<td>378</td>
<td>406</td>
<td>1436</td>
</tr>
<tr>
<td>MOVE</td>
<td>41</td>
<td>41</td>
<td>313</td>
<td>340</td>
<td>1302</td>
</tr>
<tr>
<td>conditional</td>
<td>30</td>
<td>31</td>
<td>78</td>
<td>94</td>
<td>606</td>
</tr>
<tr>
<td>IF</td>
<td>22</td>
<td>23</td>
<td>61</td>
<td>71</td>
<td>433</td>
</tr>
<tr>
<td>ELSE</td>
<td>3</td>
<td>3</td>
<td>14</td>
<td>18</td>
<td>166</td>
</tr>
<tr>
<td>ON</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>branch</td>
<td>21</td>
<td>21</td>
<td>63</td>
<td>66</td>
<td>390</td>
</tr>
<tr>
<td>CALL</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>6</td>
<td>25</td>
</tr>
<tr>
<td>PERFORM</td>
<td>4</td>
<td>4</td>
<td>10</td>
<td>10</td>
<td>165</td>
</tr>
<tr>
<td>GOTO</td>
<td>17</td>
<td>17</td>
<td>45</td>
<td>48</td>
<td>190</td>
</tr>
<tr>
<td>EXIT</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>STOP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>input-output</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>label</td>
<td>16</td>
<td>17</td>
<td>45</td>
<td>46</td>
<td>162</td>
</tr>
</tbody>
</table>

Table 2. Numbers of Statements of Program Set A
Figure 2. Numbers[TYPE] of Program set A
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>1</td>
<td>3</td>
<td>11</td>
<td>10</td>
<td>28</td>
<td>17</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>73</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>21</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>declaration</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>assignment</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>19</td>
<td>9</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>47</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>13</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>MOVE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>9</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>13</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>conditional</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IF</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ELSE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ON</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>branch</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CALL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PERFORM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GOTO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EXIT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STOP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>input-output</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>label</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>other</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3. Numbers of Altered Statements in 28 Deltas
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>15</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>comment</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>declaration</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>assignment MOVE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>conditional IF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ELSE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ON</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>branch CALL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PERFORM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GOTO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EXIT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>STOP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>input-output</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>label</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>other</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4. Numbers of Deleted Statements in 28 Deltas
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>7</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>29</td>
<td>26</td>
<td>14</td>
<td>51</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>11</td>
<td>19</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>declaration</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>13</td>
<td>0</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>assignment</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td>0</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>MOVE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>13</td>
<td>0</td>
<td>13</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>conditional</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELSE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>branch</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>CALL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>PERFORM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>GOTO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>EXIT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>input-output</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>label</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>other</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Numbers of Added Statements in 28 Deltas
changes to comment statements.

The definitions of six types of maintenance are given as below:

a. Corrective maintenance:

The corrective maintenance is the maintenance attributed to minor revision on the original version. Correcting errors or failures from source codes is also identified as corrective maintenance. The corrective maintenance results in statements being altered as well as a few statements being added and/or deleted.

b. Adaptive maintenance:

The adaptive maintenance concerns the addition of new functions or the deletion of old functions from the original codes to meet external environment requirement. In this type of maintenance, there are a lot of statements added or deleted in addition to a few statements changed.

c. Retrenchment maintenance:

The retrenchment maintenance temporarily removes a function from executable code by adding asterisks in front of the statements. The original code is converted to document or comment statements. An example of retrenchment maintenance is as follow:

Statements in original version:

IF MAORD = 'A'
GO TO 0530-READ-MATLDESC-PURCHSPECS

IF MAORD = 'N'
GO TO 0530-READ-MATLDESC-PURCHSPECS

Statements in new version:
* IF MAORD = 'A' GO TO 0530-READ-MATLDESC-PURCHSPECS
* IF MAORD = 'N' GO TO 0530-READ-MATLDESC-PURCHSPECS

Note that they are not only adding comments but also combining four lines into two lines.

The result of retrenchment maintenance increases the number[comment] from the original version to the new version. It may also decrease the numbers of statements between two versions.

d. Retrieving maintenance:

The retrieving maintenance removes asterisks from comment or documentation statements and brings the statements back into service. The changed statements become part of the executable code. The retrieving maintenance is the reverse of retrenchment maintenance. An example of retrieving maintenance is listed as below:

Statements in original version:
*
* IF EIBTRMID NOT = '302M' OR EIBTRMID NOT = '700Q'
* EXEC CICS
* SEND TEXT FROM(NICE-TRY-MESSAGE_
Statements in new version:

IF EIBTRMID NOT = '302M' OR EIBTRMID NOT = '700Q'

   EXEC CICS SEND TEXT FROM(NICE-TRY-MESSAGE LENGTH ERASE FREEKB

   END-EXEC EXEC CICS RETRUN ND-EXEC

e. Pretty printing maintenance:

   The pretty printing maintenance simply add asterisks for spacing purposes. There is no other function added or deleted. The objective of pretty printing is to allow program to be easy to read. The pretty printing maintenance increases Number[comment] in the new version and does not change other statements.

f. Documentation maintenance:

   The documentation maintenance is the addition of comment statement to a program. This is different from pretty printing maintenance. The documentation maintenance puts descriptions or explanations just before a block of source codes. Well-documented program can reduce the effort of the
reader to understand the program. The number of statements changed in documentation maintenance is similar to that of pretty printing maintenance.

3-4 Precise Rules

The precise rules for classifying types of maintenance are analyzed from the empirical data received from executing the shell program Maintain on program set A. The rules were converted into the shell program Classify (see algorithm in Appendix E). The input for the Classify program is the output from the Maintain program. Appendix F gives an example result from running the program Classify. The discussion of the rules on correction and adaptive maintenances is grouped together because of their similar situation. The same condition can be applied to retrenchment and retrieving, as well as pretty printing and documentation.

a. Corrective and Adaptive maintenances:

Here a block of statements altered is defined as a series of statements altered. If there are more than three blocks of statements in which the lines of codes are modified, or the addition and deletion of statements other than comment statements is greater than 10, the delta is classified as adaptive maintenance. Otherwise, it is said to be corrective maintenance. The term "modified block" is de-
fined as a block where the ratio of Number[TYPE] in two versions is greater than 2 if Number[TYPE] in two versions are both more than 10, or the ratio is greater than 5 if one of Number[TYPE] is less than 10. Detailed algorithms can be found in the shell program Classify in Appendix B.

b. Retrenchment and Retrieving maintenances:

In both types of maintenance, there are some altered and no deleted or added statements. The altered statements cause the changes of comment statements. Number[comment] is decreased from original version to new version in retrenchment maintenance. The value, however, is increased in retrieving maintenance.

c. Pretty printing and Documentation maintenances:

For both types of maintenance, the increase or decrease of Number[comment] is due to the addition or deletion of comment statements. If the goal of added or deleted comment statements is for spacing purpose only, the type of maintenance is classified as pretty printing maintenance. Otherwise, it is documentation maintenance. The shell program Classify can distinguish between these two types of maintenance.
3-5 Result and Discussion

The result of types of maintenance in 28 deltas from program set A is illustrated on Table 6. The empirical data are collected in Table 3, 4 and 5. Of the 28 deltas, there exists single types and combination of two or three types of maintenance. For simplicity, the combination of corrective and documentation maintenance is expressed as corrective & documentation maintenances. This example extends to any combination.

Program A-1 consists of 6 versions and 5 deltas. As there is only 1 statement altered, delta 1 is classified as corrective maintenance. In delta 2, original Number[comment] and new Number[comment] are equal to 59 and 69, respectively. The increment of 10 comment statements is due to 3 statements in original version converting to 10 comment statements in new version. As result, delta 2 is classified as retrenchment maintenance. The decrement of 11 comment statements in delta 3 is from 11 original statements modifying to 3 new statements. From the classification rules, it is clear that delta 3 is retrieving maintenance. The modification in delta 4 is similar to that in delta 1. Delta 5 has two types of maintenance, corrective & pretty printing. The pretty printing maintenance can be identified from the value of Added[comment] which is equal to the
increment of Number[comment] between two versions. Corrective maintenance is determined by the values of Deleted[TYPE] and Added[TYPE].

Program A-2 includes 4 versions. Delta 6 and 7 are retrenchment and the reason of classification is the same as from delta 2. Delta 8 has three types of maintenance, corrective & retrenchment & pretty printing. The 4 added comment statements result in the pretty printing maintenance. The retrenchment maintenance is attributed to altered statements from 4 to 2 statements. The 9 added statements, however, are classified as corrective maintenance.

Deltas 9 to 18 belong to program A-3. Of the ten deltas, five deltas are classified as corrective maintenance. Delta 11 has corrective & retrenchment types. Delta 13 has three types of maintenance. The 3 added comment statements belong to pretty printing maintenance. One block of altered statements belongs to retrenchment maintenance. The other block and the added statements, however, are classified as corrective maintenance. Delta 15 and 16 have corrective & pretty printing maintenances. In delta 18, the increment of comment statement is identified as retrenchment maintenance which came from altered statements. Of the altered statements, some contributed to corrective maintenance.
Of six deltas in program A-4, there are 3 deltas classified as corrective maintenance. Deltas 21 and 22 have the similar modification except that delta 21 is documentation and delta 22 is pretty printing maintenance. The shell program Classify can make the distinction. In delta 24, 5 added comment statements have two types, pretty printing & documentation. The other changed statements contribute to adaptive maintenance.

In program A-5, delta 25 is adaptive type because lots of statements added or deleted. Deltas 26 and 27 are classified as adaptive & documentation maintenance. The altered 15 comment statements cause the decrease of comment statements in delta 28, therefore, the delta is said to be retrieving maintenance in addition to corrective maintenance.

Figures 3 and 4 illustrate the overall analysis on program set A. Figure 3 displays the numbers of deltas on types of maintenance. The number of deltas which only has corrective type is 10. None of the deltas which have documentation and pretty printing maintenance. There exists only 3 deltas owning three types of maintenance. It is verified that program maintainer did not change many things on any version of program set A. Figure 4 lists the percentage of occurrence on each type of maintenance.
The rules described in previous section were used to classify types of maintenance in program set A. In next chapter, the rules are verified with program set B, which are received from different environment.
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL] from</td>
<td>1</td>
<td>3</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>28</td>
<td>17</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>73</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>to</td>
<td>1</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>28</td>
<td>20</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>76</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>block</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Number[comment]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>original</td>
<td>59</td>
<td>59</td>
<td>69</td>
<td>58</td>
<td>58</td>
<td>195</td>
<td>222</td>
<td>242</td>
<td>782</td>
<td>782</td>
<td>782</td>
<td>856</td>
<td>856</td>
<td>860</td>
</tr>
<tr>
<td>new</td>
<td>59</td>
<td>69</td>
<td>58</td>
<td>58</td>
<td>63</td>
<td>223</td>
<td>242</td>
<td>248</td>
<td>782</td>
<td>782</td>
<td>856</td>
<td>856</td>
<td>860</td>
<td>860</td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6a. Classification of Program Set A
<table>
<thead>
<tr>
<th>delta</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL]</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>21</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>original</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>30</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>11</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>new</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>15</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>7</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>29</td>
<td>26</td>
<td>14</td>
<td>51</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number[comment]</th>
<th>original</th>
<th>860</th>
<th>856</th>
<th>860</th>
<th>860</th>
<th>390</th>
<th>390</th>
<th>390</th>
<th>391</th>
<th>394</th>
<th>394</th>
<th>79</th>
<th>79</th>
<th>90</th>
<th>108</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>856</td>
<td>860</td>
<td>860</td>
<td>870</td>
<td>390</td>
<td>390</td>
<td>391</td>
<td>394</td>
<td>394</td>
<td>399</td>
<td>79</td>
<td>90</td>
<td>108</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Added[comment]</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>11</td>
<td>19</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification</th>
<th>Corrective</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Retrenchment</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Retrieving</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pretty printing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6b. Classification of Program Set A
Figure 3. Numbers of Delta on Types Maintenance of Program Set A
4-1 Data Programs

Program set B, which includes 8 programs, was obtained from a data processing environment. These programs have been in use for more than 10 years. For each program, 20 versions were received and analyzed with the shell programs Maintain and Classify to verify the classification rules proposed in chapter 3. The result generates 19 deltas on each program and a total number of 152 on program set B. From the sequence of maintenance on a specific program, it indicates the types of maintenance and identifies what is really made to the modification on the source codes during the maintenance period. The classifications were then compared to the maintainers' comments.

4-2 Result

In the 19 deltas from program B-1, there are 3 corrective, 13 corrective & documentation, 2 corrective & retraining & documentation maintenance, and 1 adaptive & retraining & documentation maintenance (see Table 7). The 13
corrective & documentation maintenance delta have similar changed value set. It is easy to make distinctions on these types of maintenance. All the increment of Number[comment] results from the number of Added[comment]. Additional changed statements, however, contribute to corrective maintenance.

On program B-2, 10 corrective and 7 corrective & documentation maintenance were found during the analysis. Delta 11 is classified as corrective & retrenchment maintenance. Delta 1 does not have any modification at all. The result of program B-2 is represented on Table 8.

From delta 1 through 11, the modifications on program B-3 were steady except for delta 4 being classified as adaptive & pretty printing maintenance. The changes in deltas 12, 13, and 14 were large comparing to the rest of deltas. In these phases, a combination of 4 types of maintenance were identified. In addition to the high occurrence of maintenance types, the altered numbers were also very high. Delta 15 was back to general modification. Delta 16 was adaptive & documentation maintenance. Delta 17 includes retrenchment and retrieving maintenance together, which is seldom found in the classification of program set B. Delta 18, like delta 1 in program B-2, has not any change. The last delta is only corrective maintenance.
During the changes on program B-4, the modifications were steady from the result shown on Table 10. Only types of corrective and corrective & documentation maintenance were identified. Of the 19 deltas, 16 delta were corrective & documentation and 3 were corrective maintenance.

There is only corrective & documentation maintenance in changes after delta 9 on program B-5. During the early changes, 4 of 8 deltas were also classified as corrective & documentation maintenance. Two deltas were corrective maintenance only. Delta 3 is classified as corrective & retrenchment & documentation maintenances and delta 5 is adaptive & documentation maintenance (see Table 11).

Program B-6 includes 7 corrective and corrective & documentation maintenances, respectively. The rest of the deltas contains combination of types of maintenance. Delta 16, 17, and 18 all include adaptive maintenance. Delta 9 is the only one without corrective maintenance. It is classified as retrenchment & documentation maintenances. Detailed results are shown in Table 12.

Corrective and documentation maintenances are two types which exist in the changes on program B-7 (see Table 13). Nine corrective and 9 corrective & documentation maintenance types are classified from the program Classify.
The results on program B-8 is similar to that on program B-7. Fourteen deltas with corrective & documentation maintenance were classified. Delta 16 is the only one classified as adaptive maintenance.

From the results collected from 152 deltas on program set B, the combination of corrective and documentation types were the most frequently occurring maintenance; 81 deltas belong to this combination types of maintenance. Added comment statements in the identification division contribute to documentation maintenance and the changed statements resulting in corrective maintenance. It is concluded that most maintainers explained what they modified in the identification division and gave actual changes in the procedure division. The percentages of occurrence on each type of maintenance for program set B are represented on Table 5. In contrast to the same representation from program set A, pretty printing maintenance happened at lower percentage.

4-3 Validation

The classifications were checked against the explanations for the changes given by the maintainers as comments in the environment section. None of the explanations contradicted the classifications made by the classify program. However,
the classifications gave a better indication than the comments about what types of activity had occurred. Thus, we feel that rules were successful in classifying the maintenance activities.
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>from</td>
<td>6</td>
<td>15</td>
<td>9</td>
<td>52</td>
<td>41</td>
<td>412</td>
<td>45</td>
<td>19</td>
<td>42</td>
<td>26</td>
<td>45</td>
<td>19</td>
<td>14</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>10</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>12</td>
<td>18</td>
<td>9</td>
<td>99</td>
<td>46</td>
<td>430</td>
<td>87</td>
<td>28</td>
<td>33</td>
<td>13</td>
<td>36</td>
<td>50</td>
<td>25</td>
<td>30</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>14</td>
<td>34</td>
</tr>
<tr>
<td>block</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>26</td>
<td>22</td>
<td>132</td>
<td>13</td>
<td>10</td>
<td>4</td>
<td>26</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>7</td>
<td>9</td>
<td>4</td>
<td>44</td>
<td>18</td>
<td>8</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Number[comment]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>original</td>
<td>213</td>
<td>216</td>
<td>219</td>
<td>219</td>
<td>228</td>
<td>235</td>
<td>238</td>
<td>248</td>
<td>251</td>
<td>257</td>
<td>258</td>
<td>262</td>
<td>267</td>
<td>271</td>
<td>277</td>
<td>279</td>
<td>285</td>
<td>286</td>
<td>291</td>
</tr>
<tr>
<td>new</td>
<td>216</td>
<td>219</td>
<td>219</td>
<td>228</td>
<td>235</td>
<td>238</td>
<td>248</td>
<td>251</td>
<td>257</td>
<td>258</td>
<td>262</td>
<td>267</td>
<td>271</td>
<td>277</td>
<td>279</td>
<td>285</td>
<td>286</td>
<td>291</td>
<td>293</td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[comment]</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Classification

| Corrective | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| Adaptive | | | | | | | | | | | | | | | | | | | |
| Retrenchment | | | | | | | | | | | | | | | | | | | |
| Retrieving | | | | | | | | | | | | | | | | | | | |
| Pretty printing | | | | | | | | | | | | | | | | | | | |
| Documentation | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |

Table 7. Classification of Program B-1
<table>
<thead>
<tr>
<th>Delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL] from</td>
<td>0</td>
<td>77</td>
<td>30</td>
<td>1</td>
<td>2</td>
<td>146</td>
<td>1</td>
<td>35</td>
<td>1</td>
<td>7</td>
<td>27</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>46</td>
<td>39</td>
<td>109</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>to block</td>
<td>0</td>
<td>84</td>
<td>53</td>
<td>1</td>
<td>2</td>
<td>163</td>
<td>1</td>
<td>59</td>
<td>1</td>
<td>7</td>
<td>35</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>41</td>
<td>95</td>
<td>101</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>33</td>
<td>22</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Number[comment] original</td>
<td>157</td>
<td>157</td>
<td>160</td>
<td>162</td>
<td>162</td>
<td>162</td>
<td>169</td>
<td>169</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>181</td>
<td>181</td>
<td>181</td>
<td>183</td>
<td>186</td>
<td>189</td>
<td>193</td>
<td>198</td>
</tr>
<tr>
<td>new</td>
<td>157</td>
<td>160</td>
<td>162</td>
<td>162</td>
<td>162</td>
<td>169</td>
<td>169</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>181</td>
<td>181</td>
<td>181</td>
<td>183</td>
<td>186</td>
<td>189</td>
<td>193</td>
<td>198</td>
<td>198</td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>0</td>
<td>56</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Added[comment]</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Classification**

- Corrective
- Adaptive
- Retrenchment
- Retrieving
- Pretty printing
- Documentation

Table 8. Classification of Program B-2
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL] from</td>
<td>16</td>
<td>14</td>
<td>2</td>
<td>79</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td>49</td>
<td>186</td>
<td>9</td>
<td>4</td>
<td>2782</td>
<td>1096</td>
<td>1091</td>
<td>4</td>
<td>1420</td>
<td>913</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>to block</td>
<td>22</td>
<td>25</td>
<td>1</td>
<td>69</td>
<td>16</td>
<td>11</td>
<td>2</td>
<td>62</td>
<td>184</td>
<td>3</td>
<td>2</td>
<td>2633</td>
<td>1090</td>
<td>1087</td>
<td>3</td>
<td>1460</td>
<td>904</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>101</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>12</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>13</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>9</td>
<td>2</td>
<td>137</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Number[comment] original</td>
<td>388</td>
<td>391</td>
<td>393</td>
<td>393</td>
<td>397</td>
<td>399</td>
<td>404</td>
<td>407</td>
<td>413</td>
<td>417</td>
<td>421</td>
<td>424</td>
<td>436</td>
<td>439</td>
<td>441</td>
<td>443</td>
<td>467</td>
<td>470</td>
<td>470</td>
</tr>
<tr>
<td>new</td>
<td>391</td>
<td>393</td>
<td>393</td>
<td>397</td>
<td>399</td>
<td>404</td>
<td>407</td>
<td>413</td>
<td>417</td>
<td>421</td>
<td>424</td>
<td>436</td>
<td>439</td>
<td>441</td>
<td>443</td>
<td>467</td>
<td>470</td>
<td>470</td>
<td>470</td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>424</td>
<td>125</td>
<td>125</td>
<td>0</td>
<td>165</td>
<td>134</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[comment]</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>11</td>
<td>10</td>
<td>2</td>
<td>24</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Classification

- Corrective
- Adaptive
- Retrenchment
- Retrieving
- Pretty printing
- Documentation

Table 9. Classification of Program B-3
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>from</td>
<td>4</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>5</td>
<td>16</td>
<td>3</td>
<td>19</td>
<td>5</td>
<td>2</td>
<td>46</td>
<td>3</td>
<td>1</td>
<td>17</td>
<td>4</td>
<td>3</td>
<td>38</td>
<td>3</td>
</tr>
<tr>
<td>to</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>5</td>
<td>18</td>
<td>3</td>
<td>21</td>
<td>6</td>
<td>2</td>
<td>54</td>
<td>8</td>
<td>1</td>
<td>31</td>
<td>4</td>
<td>3</td>
<td>38</td>
<td>5</td>
</tr>
<tr>
<td>block</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Number[comment]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>original</td>
<td>95</td>
<td>95</td>
<td>98</td>
<td>101</td>
<td>104</td>
<td>106</td>
<td>108</td>
<td>112</td>
<td>114</td>
<td>117</td>
<td>120</td>
<td>120</td>
<td>125</td>
<td>128</td>
<td>130</td>
<td>136</td>
<td>136</td>
<td>138</td>
<td>141</td>
</tr>
<tr>
<td>new</td>
<td>95</td>
<td>98</td>
<td>101</td>
<td>104</td>
<td>106</td>
<td>108</td>
<td>112</td>
<td>114</td>
<td>117</td>
<td>120</td>
<td>120</td>
<td>125</td>
<td>128</td>
<td>130</td>
<td>136</td>
<td>136</td>
<td>138</td>
<td>141</td>
<td>143</td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[comment]</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

**Classification**

| Corrective | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| Adaptive     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Retrenchment |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Retrieving   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Pretty printing |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Documentation |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

**Table 10. Classification of Program B-4**
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL] from</td>
<td>75</td>
<td>2</td>
<td>60</td>
<td>2</td>
<td>516</td>
<td>2</td>
<td>3</td>
<td>71</td>
<td>42</td>
<td>70</td>
<td>4</td>
<td>274</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>20</td>
<td>69</td>
<td>284</td>
</tr>
<tr>
<td>to</td>
<td>114</td>
<td>2</td>
<td>59</td>
<td>2</td>
<td>492</td>
<td>2</td>
<td>3</td>
<td>158</td>
<td>86</td>
<td>76</td>
<td>4</td>
<td>378</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>24</td>
<td>71</td>
<td>319</td>
</tr>
<tr>
<td>block</td>
<td>37</td>
<td>2</td>
<td>17</td>
<td>2</td>
<td>86</td>
<td>2</td>
<td>3</td>
<td>16</td>
<td>2</td>
<td>11</td>
<td>4</td>
<td>38</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>12</td>
<td>3</td>
<td>2</td>
<td>76</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Number[comment]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>original</td>
<td>289</td>
<td>289</td>
<td>291</td>
<td>329</td>
<td>331</td>
<td>338</td>
<td>340</td>
<td>342</td>
<td>353</td>
<td>361</td>
<td>364</td>
<td>366</td>
<td>375</td>
<td>379</td>
<td>382</td>
<td>386</td>
<td>388</td>
<td>391</td>
<td>392</td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[comment]</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

**Classification**

- Corrective: 
- Adaptive: 
- Retrenchment: 
- Retrieving: 
- Pretty printing: 
- Documentation:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
</table>

**Table 11. Classification of Program B-5**
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL] from</td>
<td>9</td>
<td>36</td>
<td>13</td>
<td>4</td>
<td>135</td>
<td>3</td>
<td>11</td>
<td>7</td>
<td>6</td>
<td>36</td>
<td>32</td>
<td>16</td>
<td>2</td>
<td>11</td>
<td>11</td>
<td>1</td>
<td>34</td>
<td>106</td>
<td>14</td>
</tr>
<tr>
<td>to block</td>
<td>6</td>
<td>11</td>
<td>4</td>
<td>4</td>
<td>61</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>14</td>
<td>28</td>
<td>5</td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>5</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>17</td>
<td>0</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>Number[comment] original</td>
<td>440</td>
<td>445</td>
<td>493</td>
<td>493</td>
<td>500</td>
<td>502</td>
<td>502</td>
<td>502</td>
<td>505</td>
<td>510</td>
<td>510</td>
<td>510</td>
<td>510</td>
<td>510</td>
<td>517</td>
<td>521</td>
<td>525</td>
<td>519</td>
<td>530</td>
</tr>
<tr>
<td>new</td>
<td>445</td>
<td>493</td>
<td>493</td>
<td>500</td>
<td>502</td>
<td>502</td>
<td>502</td>
<td>505</td>
<td>510</td>
<td>510</td>
<td>510</td>
<td>510</td>
<td>510</td>
<td>512</td>
<td>517</td>
<td>521</td>
<td>525</td>
<td>519</td>
<td>530</td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[comment]</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>8</td>
<td>2</td>
</tr>
</tbody>
</table>

**Classification**

| Corrective | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| Adaptive | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| Retrenchment | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| Retrieving | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| Pretty printing | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |
| Documentation | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | * |

Table 12. Classification of Program B-6
<table>
<thead>
<tr>
<th>delta</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altered[ALL] from to block</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1240</td>
<td>1380</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1273</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>399</td>
<td>902</td>
<td>105</td>
<td>0</td>
<td>52</td>
</tr>
<tr>
<td>Deleted[ALL]</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[ALL]</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Number[comment] original new</td>
<td>96</td>
<td>96</td>
<td>96</td>
<td>96</td>
<td>96</td>
<td>98</td>
<td>98</td>
<td>102</td>
<td>102</td>
<td>104</td>
<td>104</td>
<td>104</td>
<td>107</td>
<td>107</td>
<td>107</td>
<td>109</td>
<td>116</td>
<td>126</td>
<td>129</td>
</tr>
<tr>
<td>Altered[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Deleted[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Added[comment]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

**Classification**

- Corrective
- Adaptive
- Retrenchment
- Retrieving
- Pretty printing
- Documentation

Table 13. Classification of Program B-7
|          | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  | 17  | 18  | 19  |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| delta    |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Altered[ALL] from | 3   | 3   | 1   | 2   | 32  | 11  | 5   | 2   | 20  | 41  | 7   | 1   | 1   | 14  | 13  | 642 | 132 | 8   | 10  |
| to block | 6   | 7   | 1   | 2   | 14  | 17  | 6   | 2   | 35  | 58  | 3   | 2   | 1   | 17  | 17  | 55  | 654 | 113 | 12  | 15  |
| Deleted[ALL]     | 1   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 1   | 0   | 0   | 0   | 0   | 5   | 0   | 0   | 0   | 0   |     |
| Added[ALL]       | 5   | 3   | 3   | 2   | 16  | 5   | 0   | 2   | 6   | 15  | 4   | 0   | 0   | 11  | 7   | 16  | 0   | 7   | 4   |
| Number[comment] original | 108 | 113 | 116 | 118 | 120 | 128 | 133 | 133 | 135 | 136 | 138 | 141 | 141 | 141 | 145 | 151 | 154 | 157 |
| new            | 113 | 116 | 118 | 120 | 128 | 133 | 133 | 135 | 136 | 138 | 141 | 141 | 141 | 145 | 151 | 154 | 157 | 161 |
| Altered[comment]     | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 94  | 6   | 0   | 0   |
| Deleted[comment]     | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |
| Added[comment]       | 5   | 3   | 2   | 2   | 8   | 5   | 0   | 2   | 1   | 2   | 3   | 0   | 0   | 4   | 0   | 3   | 3   | 4   |     |

Classification

- Corrective: ...
- Adaptive: ...
- Retrenchment: ...
- Retrieving: ...
- Pretty printing: ...
- Documentation: ...

Table 14. Classification of Program B-8
Figure 5. Percentages of Occurrence on Each type of Maintenance of Program Set B
Chapter 3 gave the results of this study on the COBOL program set A. The classification rules were presented to distinguish various types of maintenance. Six types of maintenance were identified from the changes on sequential versions of the programs. The rules are in the effort to classify types of maintenance on program set A. Then the classification rules were used to verify the results from the analysis on the program set B, which came from another environment. The result of classification are also found to be satisfactory by comparing them with the maintainers' comments.

The study presents a method of classifying types of maintenance from empirical data of changes between two versions during maintenance. By means of the tools, the shell programs Maintain and Classify, managers can easily identify the types of maintenance existing in the sequential modification to a specific program. In an effort to reduce the cost of maintenance, managers can take the method as a reference with which to evaluate their maintenance effort; this method will give them an objective classification of their effort.
Recommendations for the future work for this study are proposed as following:

1) Validate the presented method with more COBOL programs from different sources to check the effectiveness of classifying types of maintenance from the tools.

2) Modify and extend the tools to test the results of programs written by other languages. Are the method and rules good enough to classify types of maintenance in other languages?

3) Six types of maintenance were classified in this study from the analysis on program set A. More classification types could be embedded in COBOL programs.

4) The presented rules were satisfactory in classifying maintenance types. Refinement of the rules is suggested to distinguish more maintenance types.
References


Appendix A. The Shell Program Maintain

```
# Shell program: Maintain
# The program calls the following subprogram:
# maint. aux  maint. auxa
# maint. auxb  maint. auxc
# maint. aux1

echo ANALYSIS FOR: $1 $2 > $1.list
echo $1 $2 > $1.list

echo $1 ' $2' >> $0.stats

******************************************************************************
# Checking module
******************************************************************************

# checks to see if divisions are present in file one
awk
BEGIN { iflag = 0; eflag = 0; dflag = 0; pflag = 0 }
/IDENTIFICATION DIVISION./ { iflag = 1 }
/ENVIRONMENT DIVISION./ { eflag = 1 }
/DATA DIVISION./ { dflag = 1 }
/PROCEDURE DIVISION/ { pflag = 1 }
END { if (iflag + eflag + dflag + pflag != 4) [ print 5 > "err1" ] } $1
if (test -f err1)
then exit 1
fi

# checks to see if divisions are present in file two
awk
BEGIN { iflag = 0; eflag = 0; dflag = 0; pflag = 0 }
/IDENTIFICATION DIVISION./ { iflag = 1 }
/ENVIRONMENT DIVISION./ { eflag = 1 }
/DATA DIVISION./ { dflag = 1 }
/PROCEDURE DIVISION/ { pflag = 1 }
END { if (iflag + eflag + dflag + pflag != 4) [ print 5 > "err2" ] } $2
if (test -f err2)
then exit 2
fi

******************************************************************************
# Preprocessing module
******************************************************************************

# removes numbers from tail end of lines
awk '{printf "$s ", $0; printf "##\n"} '$1 |
    sed 's/....##\n" $1.n
awk '{printf "$s ", $0; printf "##\n"} '$2 |
    sed 's/....##\n" $2.n

# removes skips and ejects from file
sed '/ EJECT/ d
    / SKIP/ d $1.n > $1.nnn
sed '/ EJECT/ d
    / SKIP/ d $2.n > $2.nnn
rm $1.n $2.n

# install End of program marker on both files
echo End 0000000 > $1.temp
cat $1.temp >> $1.nnn
cat $1.temp >> $2.nnn
```
# Print out messages regarding the presence or absence of divisions
# for file 1 the file has been preprocessed and appropriate
# divisions inserted without leading blanks
echo "LIST OF MISSING DIVISIONS FOR " $1 >> $1.list
#
awk
BEGIN { iddiv = 0; envdiv = 0; datadiv = 0; prodiv = 0 }
/ IDENTIFICATION DIVISION./ { iddiv++; next }
/ ENVIRONMENT DIVISION./ { envdiv++; next }
/ DATA DIVISION./ { datadiv++; next }
/ PROCEDURE DIVISION/ { prodiv++; next }
END { printf "%d %d %d %d ", iddiv, envdiv, datadiv, prodiv; printf "0, stats";
  if (iddiv == 0) print "*** IDENTIFICATION DIVISION MISSING **** ";
  if (envdiv == 0) print "*** ENVIRONMENT DIVISION MISSING **** ";
  if (datadiv == 0) print "*** DATA DIVISION MISSING **** ";
  if (prodiv==0) print "*** PROCEDURE DIVISION MISSING ****")' $1.nnn>$1.list
}
#
echo >> $1.list
echo "END OF LIST" >> $1.list
echo >> $1.list
#
# Print out messages regarding the presence or absence of divisions
# for file 2 the file has been preprocessed and appropriate divisions
# inserted without leading blanks
echo "LIST OF MISSING DIVISIONS FOR " $2 >> $2.list
#
awk
BEGIN { iddiv = 0; envdiv = 0; datadiv = 0; prodiv = 0 }
/ IDENTIFICATION DIVISION./ { iddiv++; next }
/ ENVIRONMENT DIVISION./ { envdiv++; next }
/ DATA DIVISION./ { datadiv++; next }
/ PROCEDURE DIVISION/ { prodiv++; next }
END { printf "%d %d %d %d ", iddiv, envdiv, datadiv, prodiv; printf "0, stats";
  if (iddiv == 0) print "*** IDENTIFICATION DIVISION MISSING **** ";
  if (envdiv == 0) print "*** ENVIRONMENT DIVISION MISSING **** ";
  if (datadiv == 0) print "*** DATA DIVISION MISSING **** ";
  if (prodiv==0) print "*** PROCEDURE DIVISION MISSING ****")' $2.nnn>$2.list
}
#
echo >> $2.list
echo "END OF LIST" >> $2.list
echo >> $2.list
#
# Distinguishing module
#
# flags statements in the environment division in file one
awk '/ENVIRONMENT DIVISION./,DATA DIVISION./ {
if (($1 == "ENVIRONMENT" || $1 == "DATA") && $2 == "DIVISION.")
  { print $0; next }
else [ printf "%s ", $0; printf "Env\n"; next ]
}./ [ print $0 ]' $1.nnn > $1.nnn
rm $1.nnn

# flags statements in the environment division in file two
awk '/ENVIRONMENT DIVISION./,DATA DIVISION./ {
if (($1 == "ENVIRONMENT" || $1 == "DATA") && $2 == "DIVISION.")
  { print $0; next }
else [ printf "%s ", $0; printf "Env\n"; next ]
}./ [ print $0 ]' $2.nnn > $2.nnn
rm $2.nnn

A-2
flags comment lines in file one
awk '/IDENTIFICATION DIVISION./,ENVIRONMENT DIVISION/ {
  if (($1 == "IDENTIFICATION" || $1 == "ENVIRONMENT") && $2 == "DIVISION.") {
    print $0; next }
else
  [ printf "Comment %s\n", $0; next ]}
}./ [ print $0 ]; $1.nnnn > $1.nnnnn
rm $1.nnnn

flags comment lines in file two
awk '/IDENTIFICATION DIVISION./,ENVIRONMENT DIVISION/ {
  if (($1 == "IDENTIFICATION" || $1 == "ENVIRONMENT") && $2 == "DIVISION.") {
    print $0; next }
else
  [ printf "Comment %s\n", $0; next ]}
}./ [ print $0 ]; $2.nnnn > $2.nnnnn
rm $2.nnnn

separates declarations into declaration part and initialization part
for file one
awk '/DATA DIVISION./,PROCEDURE DIVISION/ {
  if (($1 == "DATA" || $1 == "PROCEDURE") && ($2 == "DIVISION.") || (substr($1,1,1) ==))==""
    [ print $0; next ]
else
  [ if ($1 == "FILE" || $1 == "WORKING-Storage" && $2 == "SECTION.")
    [ printf "Data %s\n", $0; next ]
else
    [ if (substr($1,1,1) "/[0-9]/) [ printf "Dlrtn  " ]
      if (substr($1,1,1) "/[0-9]/) [ printf "Dlrtn  " ]
        [ i = 1
          while (i <= NF )
            [ if (($1 == "REDEFINES") || ($1 == "RENAmes") || ($1 == "VALUE")
              [ printf \nDlrtn  " ]
              printf "%s ", $1; i++ ]
          printf \nDlrtn  " ; next ]
        ]
      else
        [ i = 1
          while ( i <= NF )
            [ if (($1 == "REDEFINES") || ($1 == "RENAmes") || ($1 == "VALUE")
              [ printf \nDlrtn  " ]
              printf "%s ", $1; i++ ]
          ]
        ]
    ]}
  [ print $0 ]; $1.nnnn > $1.nnnnn
rm $1.nnnnn

separates declarations into declaration part and initialization part
for file two
awk '/DATA DIVISION./,PROCEDURE DIVISION/ {
  if (($1 == "DATA" || $1 == "PROCEDURE") && ($2 == "DIVISION.") || (substr($1,1,1) ==))==""
    [ print $0; next ]
else
  [ if ($1 == "FILE" || $1 == "WORKING-Storage" && $2 == "SECTION.")
    [ printf "Data %s\n", $0; next ]
else
    [ if (substr($1,1,1) "/[0-9]/) [ printf "Dlrtn  " ]

A-3
if (substr($NF, length ($NF), 1) == ".")
  { i = 1
   while (i <= NF)
     { if (($1 == "REDEFINES") || ($1 == "Renames") || ($1 == "VALUE"))
       { printf "\n\n Dlrtn
 $s \n", $1; i++ }
       printf "\n\n "
     }
   else
     { i = 1
      while (i <= NF)
        { if (($1 == "REDEFINES") || ($1 == "Renames") || ($1 == "VALUE"))
          { printf "\n\n Dlrtn
 $s \n", $1; i++ }
        next
      }
    }
  }

./ [ print $0 ]' $2.mnnnn > $2.nn
rm $2.mnnnn
# # break procedure division into statements
maintain.auxc $1.nn
mv $1.nn.modules $1.modules
mv $1.nn.calls $1.calls
maintain.auxc $2.nn
# # Printing the heading of the overall number for file one
# echo '----------------------------------------------------------' >> $1.list
echo >> $1.list
echo 'OVERALL ANALYSIS OF STATEMENTS' >> $1.list
echo >> $1.list
# # Calculation module is embedded in maintain.auxl
# compute number(TYPE) of each statement for file one and list the result
maintain.auxc $1.nn >> $1.list
echo -n "$1\"' $2\"' >> 0.stats,totals
echo -n "$1\"' $2\"' >> 0.stats,al ters
echo -n "$1\"' $2\"' >> 0.stats,delet es
echo -n "$1\"' $2\"' >> 0.stats,adds
cat auxtemp >> 0.stats,totals
echo >> 0.stats,totals
rm auxtemp
#
# fix up divisions file one
awk "/IDENTIFICATION DIVISION/ { printf "%s %s\n",$1,$2; next }
/ENVIRONMENT DIVISION/ { printf "%s %s\n",$1,$2; next }
/DATA DIVISION/ { printf "%s %s\n",$1,$2; next }
/PROCEDURE DIVISION/ { printf "%s %s\n",$1,$2; next }
./ [ print $0; next ]' $1.nn > $1.nn
rm $1.nn
mv $1.nn $1.nn
#
# fix up divisions file two
awk "/IDENTIFICATION DIVISION/ { printf "%s %s\n",$1,$2; next }
/ENVIRONMENT DIVISION/ { printf "%s %s\n",$1,$2; next }
/DATA DIVISION/ { printf "%s %s\n",$1,$2; next }
/PROCEDURE DIVISION/ { printf "%s %s\n",$1,$2; next }
./ [ print $0; next ]' $2.nn > $2.nn
rm $2.nn
mv $2.nn $2.nn
A-4
Difference module

flag alters, deletions and addition in file one --> file two
diff -e $1.nn $2.nn 'grep '^[0-9]' |
sed 's/\|/g
s/a/ a /g
s/c/ c /g
s/d/ d /g' |
awk
BEGIN {printf "BEGIN [i=0]\n"
    NF==2 {if ($2 == "a")
        {printf "NR== %d
        printf \"a\",$0
        printf \"b\",$0 ;i=1;J+=1"
        END {print "\n[if (i==0) i=0]"
    awk -f anoth $1.nn > $1.r
rm anoth

flag alters, deletions, and additions in file two --> file one
diff -e $2.nn $1.nn 'grep '^[0-9]' |
sed 's/\|/g
s/a/ a /g
s/c/ c /g
s/d/ d /g' |
awk
BEGIN {printf "BEGIN [i=0]\n"
    NF==2 {if ($2 == "a")
        {printf "NR== %d
        printf \"a\",$0
        printf \"b\",$0 ;i=1;J+=1"
        END {print "\n[if (i==0) i=0]"
    awk -f anoth $2.nn > $1.h
rm anoth

Report module

maintain, aux $1.r >> $1.list
maintain, auxa $1.h >> $1.list

if there were altered, then print out an analysis of those altered
if (test -f $1.r.c)
then
    maintain, auxb $1.nn $2.nn >> $1.list
rm $1.r.c
fi

rm $1.nn $2.nn
if there were deleted, then list the deleted statements
if (test -f delefile)
then
    echo '--------------------------------------------------------------------------------' >> $1.list
    echo >> $1.list
```bash
echo 'LIST THE DELETED STATEMENTS' >> $1.list
echo >> $1.list
cat deletefile >> $1.list
rm deletefile
echo >> $1.list

# if there were added, then list the added statements
if [ -f addfile ]
then
    echo '--------------------------------------------' >> $1.list
echo >> $1.list
    echo 'LIST THE ADDED STATEMENTS' >> $1.list
echo >> $1.list
cat addfile >> $1.list
rm addfile
echo >> $1.list
fi
rm .stats
rm .stats.
r $1.
r $2.nn.
# eof: maintain
```
program: maintain.aux

# tabulating number of addition sections, deletions, altered
# create a new file "ctemp", "alterfile" to store altered statements
# create a new file "dtemp", "delefile" to store deleted statements

BEGIN [ A = 0; C = 0; D = 0 ]
/a/ [ A++ ]
/c/ [ print $0 > "ctemp"; C++ ]
/d/ [ print $0 > "dtemp"; D++ ]

END [ printf \n

TOTAL NUMBER OF ADDED SECTIONS IS: $d \n

TOTAL NUMBER OF ALTERED: $d \n

TOTAL NUMBER OF DELETIONS: $d \n


] $1

if (test -f ctemp) then
sed 's/c/ / ctemp > alterfile
# compute Number(TYPE) of each statement in altered file and list result
maintain.aux1 alterfile
cat auxtemp >> 0.stats.altered
echo "altered" > $1.c
rm ctemp alterfile auxtemp
else
echo "*** NO STATEMENTS ALTERED ***"
fi

if (test -f dtemp) then
sed 's/d/ / dtemp > delefile
# compute Number(TYPE) of each statement for delefile and list result
maintain.aux1 delefile
cat auxtemp >> 0.stats.deletes
rm dtemp auxtemp
else
echo "*** NO STATEMENTS DELETED ***"
fi

echo 'ANALYSIS OF STATEMENTS ADDED'
echo 'ANALYSIS OF STATEMENTS DELETED'
echo 'ANALYSIS OF STATEMENTS ALTERED'
echo 'EOF: maintain.aux'
# program: maintain.auxa
# tabulate the number of added statements via flipping of files
awk 'BEGIN [ D = 0 ]
/d/ { print $0 > "dtemp"; D++ }
END [ printf "TOTAL NUMBER OF ADDED STATEMENTS: \d\n\n", D;
       printf "%d\n", D >> "0.stats"]' $1
# if there were additions
# then remove d from front
# tabulate number of each statement type via maintain.aux1
if (test -f dtemp)
then
    sed 's/d/*dtemp > addsfile
#    # compute Number[TYPE] of each statement in added file
maintain.aux1 addsfile
    cat auxtemp >> 0.stats.adds
    rm dtemp auxtemp
fi
# eof: maintain.auxa

# program: maintain.auxb
# processes altered, listing original statement, and new statement
echo '-----------------------------------------------------------'
echo
echo 'ANALYSIS OF ALTERED STATEMENTS'
echo
echo
diff -1 $1 $2 |
awk ' /[a|d]/ [flag = 0]
/o/ [flag = 1; print $0]
</,/[c|d|a]/ [if (flag == 1) print $0]' |
awk ' /o/ [printf "\n\n"; next]
/</ [printf "ORIGINAL LINE ", $0; next ]
/>/ [ print "NEWLINE ", $0; next ]
/>/* [if (NF == 1) [ printf "\nALTED TO \n\n" ]; next]'
# eof: maintain.auxb
# program: maintain.auxc
#
# process the procedure division splitting statements up
#   awk '/PROCEDURE DIVISION/,.End/ { print $0 >> "last.part"; next }
#     { print $0 >> "first.part" }' $!
rm $!

# place a q in front of all keywords we are looking at
sed 's/ MOVE /qMOVE /g
    s/ ADD /qADD /g
    s/ SUBTRACT /qSUBTRACT /g
    s/ MULTIPLY /qMULTIPLY /g
    s/ DIVIDE /qDIVIDE /g
    s/ COMPUTE /qCOMPUTE /g
    s/ IF /qIF /g
    s/ ELSE /qELSE /g
    s/ ON /qON /g
    s/ AT END /qAT END /g
    s/ CALL /qCALL /g
    s/ PERFORM /qPERFORM /g
    s/ GO /qGO /g
    s/ ALTER /qALTER /g
    s/ NEXT /qNEXT /g
    s/ EXIT /qEXIT /g
    s/ STOP /qSTOP /g
    s/ COPY /qCOPY /g
    s/ DELETE /qDELETE /g
    s/ DISPLAY /qDISPLAY /g
    s/ OPEN /qOPEN /g
    s/ CLOSE /qCLOSE /g
    s/ READ /qREAD /g
    s/ REWRITE /qREWRITE /g
    s/ WRITE /qWRITE /g
    s/ ACCEPT /qACCEPT /g
    s/ SEARCH /qSEARCH /g
    s/ SORT /qSORT /g
    s/ SET /qSET /g
    s/ GOBACK /qGOBACK /g
    s/ EXEC CICS/ qEXEC CICS/g
    s/ TRANSFORM/ qTRANSFORM/g
    s/ EXAMINE /qEXAMINE /g
    s/ INSPECT /qINSPECT /g' last.part >> last.part
rm last.part

# split up into keyword per line
awk ' BEGIN { line = 0;)
/End/ { printf "\n"; next }
[ if (substr($1,1,1) == "#"[ if (line == 1) [ printf "\n" ]
    printf "%s\n",$0; line = 0 }
else

  { i = 1
    while (i <= NF)
      [ if (substr ($1,1,1) == "q")
        [ if (line == 1) [ printf "\n" ]
          if (substr ($1,2,1) == "q")
            [ printf "%s ",substr($1,3,length ($1)-2); line = 0 ]

A-9
else
  \{ printf "%s \n", substr($1,2,length ($1)-1); line = 1 \}
\}
else \{ printf "%s; line = 1 \}
i++
\}
if (substr($NF, length ($NF),1) == ".") \{ printf \"\n\; line = 0 \}
\}

\# last.part > last.part
rm last.part

\# remove q in comment
sed 's/q//g' last.part > last.part.a
rm last.part
mv last.part.a last.part

\# indent for if nesting levels printable version
awk 'BEGIN \{ level = 0 \}
NF = 1 && substr($NF, length ($NF),1) == "." && substr($NF,1,1) "/[0-9]/ \{ level = 0 \}
\/. \{ if (level != 0) \{
i = 1
while (i <= level) \{
printf \"\n\; i++
\}
\}
/IF/ \{ print $0
  level++
  if ((substr ($NF, length ($NF),1) == ".") && (level > 0)) \{ level-- \}
  \}
/./ \{ print $0
  if ((substr ($NF, length ($NF),1) == ".") && (level > 0)) \{ level = 0 \}\} last.part > last.part.n
rm last.part

\# return in original file
cat first.part last.part.n > "$'

\# create files to be used to hand generate the hierarchy diagram
maintain.aux = last.part.n
mv calls $1.calls
mv modules $1.modules
rm first.part last.part.n
\# eof: maintain.aux
BEGIN { cc0_cnt = 0; r_com_cnt = 0; nl_com_cnt = 0;
    other_cnt = 0; cics_cnt = 0; goback_cnt = 0; transform_cnt = 0;
    tdec_cnt = 0; dec_cnt = 0; dsec_cnt = 0; fd_cnt = 0;
    value_cnt = 0; redefinea_cnt = 0; renamea_cnt = 0;
    assign_cnt = 0; movea_cnt = 0; add_cnt = 0; divide_cnt = 0;
    multiply_cnt = 0; subtract_cnt = 0; compute_cnt = 0;
    branch_cnt = 0; next_cnt = 0; exit_cnt = 0; stop_cnt = 0;
    goto_cnt = 0; alter_cnt = 0; perform_cnt = 0; call_cnt = 0;
    l_O_cnt = 0; copy_cnt = 0; delete_cnt = 0; display_cnt = 0;
    open_cnt = 0; close_cnt = 0; read_cnt = 0; write_cnt = 0;
    re_write_cnt = 0; accept_cnt = 0;
    examine_cnt = 0; inspect_cnt = 0;
    search_cnt = 0; sort_cnt = 0; set_cnt = 0;
    cond_cnt = 0; if_cnt = 0; else_cnt = 0; on_cnt = 0; once_cnt = 0;
    env_cnt = 0; configuration_cnt = 0; in_cnt = 0; file_cont_cnt = 0;
    select_cnt = 0; special_cnt = 0; sp_is_cnt = 0; scomp_cnt = 0;
    ocomp_cnt = 0; ibm_cnt = 0;
    label_cnt = 0; at_end_cnt = 0
}

NF == 1 && substr($NF,length($NF),1) == "." && substr($NF,1,1) - /[0-9]/
  [ label_cnt++]
# skip divisions
/ DIVISION/ [ next ]

# comments
substr($1,1,1) == "##" [ com_cnt++
    if (NF == 1) [ nl_com_cnt++ ]
   next ]
/Comment/ [ r_com_cnt++; com_cnt++; next ]

# environment section statements
/Env/ [ if ($1 == "CONFIGURATION") [ configuration_cnt++; env_cnt++; next ]
   else [ if ($1 == "INPUT-OUTPUT") [ in_out_cnt++; env_cnt++; next ]
   else [ if ($1 == "FILE-CONTROL") [ file_cnt_cnt++; env_cnt++; next ]
   else [ if ($1 == "SPECIAL-NAMES") [ special_cnt++; env_cnt++; next ]
   else [ if ($1 == "SOURCE-COMPUTER") [ source_cnt++; env_cnt++; next ]
   else [ if ($1 == "OBJECT-COMPUTER") [ ocomp_cnt++; env_cnt++; next ]
   else [ if ($1 == "IBM-3083") [ ibm_cnt++; env_cnt++; next ]
   else [ if ($1 == "IBM-370") [ ibm_cnt++; env_cnt++; next ]
   else [ if ($1 == "SELECT") [ select_cnt++; env_cnt++; next ]
   else [ if ($2 == "IS") [ sp_is_cnt++; env_cnt++; next ]]
   ]]]]]]

# declarations
/Dirtn/ [ if ($2 == "FD") [ fd_cnt++; tdec_cnt++; next ]
   else [ if ($2 == "VALUE") [ value_cnt++; tdec_cnt++; next ]
   else [ if ($2 == "REDEFINES") [ redefine_cnt++; tdec_cnt++; next ]
   else [ if ($2 == "RENAME") [ renamea_cnt++; tdec_cnt++; next ]
   else [ if ($2 == "VALUE") [ sp_is_cnt++; tdec_cnt++; next ]]
   ] ]]]]

/ ELSE/ [ cond_cnt++; else_cnt++; next ]
# assignment
/MOV/ [ assin_cnt++; mova_cnt++; next ]
/ADD/ [ assin_cnt++; add_cnt++; next ]
/SUBTRACT/ [ assin_cnt++; subtract_cnt++; next ]
/MULTIPLY/ [ assin_cnt++; multiply_cnt++; next ]
/ DIVIDE/ [ assin_cnt++; divide_cnt++; next ]
/COMPUTE/ [ assin_cnt++; compute_cnt++; next ]
I

conditionals

/IF / { cond_cnt++; if_cnt++; next }

/O SIZE ERROR / { cond_cnt++; onsize_cnt++; next }

/O AT END / { cond_cnt++; at_end_cnt++; next; }

# looping -- branching

/EXIT / { branch_cnt++; exit_cnt++; next }

/CALL / { branch_cnt++; call_cnt++; next }

/PERFORM / { branch_cnt++; perform_cnt++; next }

/GO TO / { branch_cnt++; goto_cnt++; next }

/NEXT / { branch_cnt++; next_cnt++; next }

/STOP / { branch_cnt++; stop_cnt++; next }

# input-output

/DELETE / { I_Q_cnt++; delete_cnt++; next }

/DISPLAY / { I_Q_cnt++; display_cnt++; next }

/OPEN / { I_Q_cnt++; open_cnt++; next }

/CLOSE / { I_Q_cnt++; close_cnt++; next }

/READ / { I_Q_cnt++; read_cnt++; next }

/REWRITE / { I_Q_cnt++; rewrite_cnt++; next }

/ACCEPT / { I_Q_cnt++; accept_cnt++; next }

/GOBACK / { goback_cnt++; other_cnt++; next }

END { u_cs = com_cnt - r_com_cnt - nl_com_cnt;
  print "NUMBER OF LINES OF COMMENTS : ",com_cnt;
  print " IDENTIFICATION DIVISION : ",r_com_cnt;
  print " SPACING PURPOSES : ",nl_com_cnt;
  print " USEFUL COMMENTS : ",u_cs;
  printf "%d %d %d %d ",com_cnt,r_com_cnt,nl_com_cnt,u_cs >> "auxtemp"
  print ";
  print "NUMBER OF ENVIRONMENT STATEMENTS : ",env_cnt;
  print " CONFIGURATION SECTION : ",configuration_cnt;
  print " SOURCE-COMPUTER : ",scomp_cnt;
  print " OBJECT-COMPUTER : ",ocomp_cnt;
  print " COMPUTER SPECIFICATION : ",ibm_cnt;
  print " SPECIAL NAMES : ",special_cnt;
  print " SPECIAL NAME ASSIGNMENT : ",sp_iq_cnt;
  print " INPUT-OUTPUT SECTION : ",in_out_cnt;
  print " FILE-CONTROL : ",file_cnt_cnt;
  print " SELECT : ",select_cnt;
  printf "%d ",env_cnt >> "auxtemp";
  printf "%d ",configuration_cnt >> "auxtemp";
  printf "%d ",scomp_cnt >> "auxtemp";
  printf "%d ",ocomp_cnt >> "auxtemp";
  printf "%d ",ibm_cnt >> "auxtemp";
  printf "%d ",special_cnt >> "auxtemp";
  printf "%d ",sp_iq_cnt >> "auxtemp";
  printf "%d ",in_out_cnt >> "auxtemp";
  printf "%d ",file_cnt_cnt >> "auxtemp";
  printf "%d ",select_cnt >> "auxtemp";
  print ";
  print "NUMBER OF DECLARATIONS : ",tdec_cnt;

A-12
print " SECTIONS : " , dseq_cnt ;
print " FD : " , id_cnt ;
print " DECLARATIONS : " , decl_cnt ;
print " VALUE CLAUSES : " , value_cnt ;
print " REDEFINES : " , redefine_cnt ;
print " RENAMES : " , rename_cnt ;
printf " %d " , tdec_cnt >> "auxtemp" ;
printf " %d " , d3eG_cnt >> "auxtemp" ;
printf " %d " , fd_cnt >> "auxtemp" ;
printf " %d " , dec_cnt >> "auxtemp" ;
printf " %d " , value_cnt , redefine_cnt , rename_cnt >> "auxtemp" ;
print " NUMBER OF ASSIGNMENTS : " , assign_cnt ;
print " *** note that the above total includes VALUES CLAUSES ***" ;
print " MOVE : " , move_cnt ;
print " ADD : " , add_cnt ;
print " SUBTRACT : " , subtract_cnt ;
print " MULTIPLY : " , multiply_cnt ;
print " DIVIDE : " , divide_cnt ;
print " COMPUTE : " , compute_cnt ;
printf " %d %d "%d" , assign_cnt , move_cnt , add_cnt , subtract_cnt >> "auxtemp" ;
printf " %d %d "%d" , multiply_cnt , divide_cnt , compute_cnt >> "auxtemp" ;
print " NUMBER OF CONDITIONALS : " , cond_cnt ;
printf " %d "%d" , cond_cnt >> "auxtemp" ;
print " IF : " , if_cnt ;
print " ELSE : " , else_cnt ;
print " ON : " , on_cnt ;
print " ON SIZE ERROR : " , onsize_cnt ;
print " AT END : " , at_end_cnt ;
printf " %d %d "%d" , cond_cnt , if_cnt , else_cnt , on_cnt >> "auxtemp" ;
printf " %d "%d" , onsize_cnt , at_end_cnt >> "auxtemp" ;
print " NUMBER OF BRANCHINGS : " , branch_cnt ;
printf " %d "%d" , branch_cnt , call_cnt , perform_cnt , goto_cnt >> "auxtemp" ;
printf " %d "%d" , next_cnt , exit_cnt , stop_cnt >> "auxtemp" ;
print " NUMBER OF INPUT/OUTPUT : " , i_q_cnt ;
printf " %d "%d" , i_q_cnt >> "auxtemp" ;
print " DELETE : " , delete_cnt ;
print " DISPLAY : " , display_cnt ;
print " OPEN : " , open_cnt ;
print " CLOSE : " , close_cnt ;
print " READ : " , read_cnt ;
print " WRITE : " , write_cnt ;
print " ACCEPT : " , accept_cnt ;
printf " %d "%d" , i_q_cnt >> "auxtemp" ;
printf " %d "%d" , delete_cnt , display_cnt , open_cnt , close_cnt >> "auxtemp" ;
printf " %d "%d" , read_cnt , write_cnt , accept_cnt >> "auxtemp" ;
printf " %d "%d" , i_q_cnt , accept_cnt >> "auxtemp" ;
print " NUMBER OF LABELS : " , label_cnt ;
printf " %d "%d" , label_cnt >> "auxtemp" ;
print " NUMBER OF OTHER STATEMENTS : " , other_cnt ;
print " COPY : " , copy_cnt ;
print " ALTER : " , alter_cnt ;
print " TRANSFORM : " , transform_cnt ;
print " EXAMINE : " , examine_cnt ;
print " INSPECT : " , inspect_cnt ;
printf " %d "%d" , other_cnt , copy_cnt >> "auxtemp" ;
printf " %d "%d" , alter_cnt , transform_cnt >> "auxtemp" ;
printf "%d ", examine_cnt >> "auxtemp";
printf "%d ", inspect_cnt >> "auxtemp";
printf "%d ", search_cnt >> "auxtemp";
printf "%d ", sort_cnt >> "auxtemp";
printf "%d ", set_cnt >> "auxtemp";
printf "%d ", cics_cnt >> "auxtemp";
printf "%d ", gback_cnt >> "auxtemp";

print "SEARCH" : ", search_cnt;
print "SCRT" : ", sort_cnt;
print "SET" : ", set_cnt;
print "CICS EXEC" : ", cics_cnt;
print "GBACK" : ", gback_cnt "$1";

# eof: maintain.aux1
Appendix B. Example COBOL Program Version 1

IDENTIFICATION DIVISION.
  PROGRAM-ID. XXXX
  AUTHOR. XXXX, XXXX
  VERSION 1.
  INSTALLATION. XXXXXXXXXX
  DATE-Written. APRIL 1964
  * Program Was Written From PRG JEX 10-3-64 MVS
  * Conversion, Modified Select Clause

ENVIRONMENT DIVISION.
  CONFIGURATION SECTION.
  SOURCE-COMPUTER. IBM-370.
  OBJECT-COMPUTER. IBM-370.

INPUT-OUTPUT SECTION.
  FILE-CONTROL.
    SELECT PRINT-FILE ASSIGN SYSo03-UR-1403-S.
    SELECT CARD-FILE ASSIGN SYSo04-UR-2520R-S.

DATA DIVISION.
  FILE SECTION.
  FD PRINT-FILE LABEL RECORDS ARE OMITTED.
    REPORT IS REPORT-DETAIL.
  FD CARD-FILE LABEL RECORDS ARE OMITTED.
  01 CARD-REC   PIC X(80).

WORKING-STORAGE SECTION.
  01 WRK-REC.
    02 C-FILL PIC X(5).
    02 FILLER REDEFINES C-FILL.
    03 FILLER PIC X.
       88 DOLLAR-SIGN VALUE '$'.
    03 FILLER PIC X(4).
    02 C-DATE REDEFINES C-FILL PIC 9(5).
    02 C-REF PIC X(6).
    02 FILLER PIC X(5).
    02 C-ACTA PIC X(4).
    02 C-ACTB PIC X(3).
    02 C-ACTC PIC X(3).
    02 FILLER PIC X.
    02 C-AMNTX PIC X(11).
    02 C-AMNT REDEFINES C-AMNTX PIC 9(9) V99.
    02 C-AMNTSIGN PIC X.
    02 FILLER PIC X(8).
    02 C-TYPE PIC X.
    02 FILLER PIC X(12).
    02 C-VEND PIC X(5).
    02 C-DESC PIC X(15).

  01 AMT      PIC S9(9) V99 VALUE ZERO.

REPORT SECTION.
  0D REPORT-DETAIL
    PAGE LIMIT IS 65 LINES
    HEADING 1
    FIRST DETAIL 4.
  01 PAGE-HEADER TYPE IS PAGE HEADING.
    05 LINE NUMBER IS 1.
      10 COLUMN 02 PIC X(6) VALUE 'ED0283'.
      10 COLUMN 30 PIC X(21) VALUE 'DISTRIBUTION KICKOUTS'.
      10 COLUMN 78 PIC X(4) VALUE 'DATE'.

B-1
10 COLUMN 93 PIC X(8) SOURCE CURRENT-DATE.
05 LINE NUMBER IS PLUS 1.
10 COLUMN 78 PIC X(4) VALUE 'PAGE'.
10 COLUMN 92 PIC X(4) SOURCE PAGE-COUNTER.
05 LINE NUMBER IS PLUS 1.
10 COLUMN 03 PIC X(4) VALUE 'DATE'.
10 COLUMN 12 PIC X(4) VALUE 'REF NO'.
10 COLUMN 23 PIC X(4) VALUE 'ACT NO'.
10 COLUMN 57 PIC X(4) VALUE 'TYPE'.
10 COLUMN 74 PIC X(4) VALUE 'VENDOR'.
10 COLUMN 95 PIC X(11) VALUE 'DESCRIPTION'.
05 LINE NUMBER IS PLUS 1.
10 COLUMN 20 PIC X(10) VALUE SPACES.

01 REPORT-LINE TYPE IS DETAIL.
05 LINE NUMBER IS PLUS 2.
10 COLUMN 02 PIC 99999999 SOURCE C-DATE.
10 COLUMN 04 PIC X VALUE '-'.
10 COLUMN 07 PIC X VALUE '-'.
10 COLUMN 12 PIC X(4) SOURCE C-REF.
10 COLUMN 20 PIC X(4) SOURCE C-ACCT.
10 COLUMN 24 PIC X VALUE '-'.
10 COLUMN 25 PIC X(3) SOURCE C-ACTB.
10 COLUMN 28 PIC X VALUE '-'.
10 COLUMN 29 PIC X(3) SOURCE C-ACIC.
10 COLUMN 51 PIC 8228282282288888 SOURCE AMT.
10 COLUMN 62 PIC X SOURCE C-TYPE.
10 COLUMN 74 PIC X(5) SOURCE C-VEND.
10 COLUMN 83 PIC X(15) SOURCE C-DESC.

01 ERROR-DETAIL TYPE IS DETAIL.
05 LINE NUMBER IS PLUS 3.
10 COLUMN 02 PIC X(9) VALUE '***ERROR***'.
10 COLUMN 10 PIC X(80) SOURCE WRK-REC.

PROCEDURE DIVISION.

OPEN OUTPUT PRINT-FILE
INPUT CARD-FILE.
INITIATE REPORT-DETAIL.

1000-LOOP.
READ CARD-FILE INTO WRK-REC AT END GO TO 3000-EOF.
IF DOLLAR-SIGN
  GO TO 1000-LOOP.
EXAMINE C-AMTX REPLACING ALL SPACES BY ZERO.
  IF C-AMTX NOT NUMERIC
    GENERATE ERROR-DETAIL
    MOVE ZERO TO C-AMTX.
    COMPUTE AMT EQUAL C-AMNT * -1
  ELSE
    MOVE C-AMNT TO AMT.
    GENERATE REPORT-LINE.
    MOVE ZERO TO C-AMNT.
    GO TO 1000-LOOP.
3000-EOF.
TERMINATE REPORT-DETAIL.
CLOSE CARD-FILE PRINT-FILE
STOP RUN.
Appendix C. Example COBOL Program Version 2

IDENTIFICATION DIVISION.
PROGRAM-ID. XXXXX
AUTHOR. XXXXX, XXXXX
VERSION 2.
INSTALLATION. XXXXXXXXXXX
DATE-WRITTEN. APRIL 1984

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.

INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT PRINT-FILE ASSIGN UT-P1L.
SELECT CARD-FILE ASSIGN UT-CARDIN.

DATA DIVISION.
FILE SECTION.
FD PRINT-FILE LABEL RECORDS ARE OMITTED.
REPORT IS REPORT-DETAIL.
FD CARD-FILE LABEL RECORDS ARE OMITTED.
01 CARD-REC PIC X(80).

WORKING-Storage SECTION.
01 W RK-REC.
  02 C-FILL PIC X(5).
  02 FILLER REDEFINES C-FILL.
  03 FILLER PIC X.
     88 DOLLAR-SIGN VALUE '-'.
  03 FILLER PIC X(4).
  02 C-DATE REDEFINES C-FILL PIC 9(5).
  02 C-REF PIC X(6).
  02 FILLER PIC X(5).
  02 C-ACTA PIC X(4).
  02 C-ACTB PIC X(3).
  02 C-ACTC PIC X(3).
  02 FILLER PIC X.
  02 C-AMNTX PIC X(11).
  02 C-AMNT REDEFINES C-AMNTX PIC 9(9)V99.
  02 C-AMNTSIGN PIC X.
  02 FILLER PIC X(8).
  02 C-TYPE PIC X.
  02 FILLER PIC X(12).
  02 C-VEND PIC X(5).
  02 C-DESC PIC X(15).

01 AMT.
  PIC S9(9)V99 VALUE ZERO.

REPORT SECTION.
RD REPORT-DETAIL.
  PAGE LIMIT IS 65 LINES
  HEADING 1
  FIRST DETAIL 4.
01 PAGE-HEADER TYPE IS PAGE HEADING.
  05 LINE NUMBER IS 1.
    10 COLUMN 02 PIC X(6) VALUE 'EDO283'.
    10 COLUMN 30 PIC X(21) VALUE 'DISTRIBUTION KICKOUTS'.
    10 COLUMN 78 PIC X(4) VALUE 'DATE'.
    10 COLUMN 83 PIC X(8) SOURCE CURRENT-DATE.
PROCEDURE DIVISION.

GO TO 1000-LOOP.

I0 COLUM 03 PIC X(4) VALUE 'DATE'.
10 COLUM 12 PIC X(6) VALUE 'REF NO'.
10 COLUM 23 PIC X(7) VALUE 'ACCT NO'.
10 COLUM 57 PIC X(3) VALUE 'AMT'.
10 COLUM 67 PIC X(4) VALUE 'TYPE'.
10 COLUM 78 PIC X(5) VALUE 'VENDOR'.
05 LINE NUMBER IS PLUS 1.
10 COLUM 20 PIC X(10) VALUE SPACES.

01 REPORT-LINE TYPE IS DETAIL.
05 LINE NUMBER IS PLUS 2.
10 COLUM 03 PIC X(9) VALUE '****ERROR'.
10 COLUM 10 PIC X(80) SOURCE WRK-REC.

PROCEDURE DIVISION.

OPEN OUTPUT PRINT-FILE
OPEN INPUT CARD-FILE.
INITIATE REPORT-DETAIL.

1000-LOOP.

READ CARD-FILE INTO WRK-REC AT END GO TO 3000-EOF.
IF DOLLAR-SIGN

GO TO 1000-LOOP.

EXAMINE C-AMTX REPLACING ALL SPACES BY ZERO.
IF C-AMTX NOT NUMERIC

GENERATE ERROR-DETAIL.
MOVE ZERO TO C-AMTX.
IF C-AMT$IGN EQUAL ' -'

COMPUTE AMT EQUAL C-AMTX - 1
ELSE

MOVE C-AMTX TO AMT.
GENERATE REPORT-LINE.
MOVE ZERO TO C-AMTX.
GO TO 1000-LOOP.

3000-EOF.

TERMINATE REPORT-DETAIL.
CLOSE CARD-FILE PRINT-FILE
STOP RUN.
Appendix D. Result from Running Maintain

ANALYSIS FOR: COBOL.1 COBOL.2

LIST OF MISSING DIVISIONS FOR COBOL.1

END OF LIST

LIST OF MISSING DIVISIONS FOR COBOL.2

END OF LIST

OVERALL ANALYSIS OF STATEMENTS

NUMBER OF LINES OF COMMENTS : 10
  IDENTIFICATION DIVISION : 7
  SPACING PURPOSES : 0
  USEFUL COMMENTS : 3

NUMBER OF ENVIRONMENT STATEMENTS : 7
  CONFIGURATION SECTION : 1
  SOURCE-COMPUTER : 1
  OBJECT-COMPUTER : 1
  COMPUTER SPECIFICATION : 0
  SPECIAL NAMES : 0
  SPECIAL NAME ASSIGNMENT : 0
  INPUT-OUTPUT SECTION : 1
  FILE-CONTROL : 1
  SELECT : 2

NUMBER OF DECLARATIONS : 95
  SECTIONS : 2
  FD : 0
  DECLARATIONS : 61
  VALUE CLAUSES : 19
  REDEFINES : 3
  RENAMES : 0

NUMBER OF ASSIGNMENTS : 21
  MOVE : 1
  ADD : 0
  SUBTRACT : 0
  MULTIPLY : 0
  DIVIDE : 0
  COMPUTE : 1

NUMBER OF CONDITIONALS : 4
  IF : 2
  ELSE : 1
  ON : 0
  ON SIZE ERROR : 0
  AT END : 1

NUMBER OF BRANCHINGS : 4

D-1
CALL : 0
PERFCRM : 0
GO TO : 3
NEXT : 0
EXIT : 0
STOP : 1

NUMBER OF INPUT/OUTPUT : 3
DELETE : 0
DISPLAY : 0
OPEN : 1
CLOSE : 1
READ : 1
REWRITE : 0
WRITE : 0
ACCEPT : 0

NUMBER OF LABELS : 2

NUMBER OF OTHER STATEMENTS : 1
COPY : 0
ALTER : 0
TRANSFORM : 0
EXAMINE : 1
INSPECT : 0
SEARCH : 0
SORT : 0
SET : 0
CICS EXEC : 0
GOBACK : 0

TOTAL NUMBER OF ADDED SECTIONS IS: 1
TOTAL NUMBER OF ALTERS: 7
TOTAL NUMBER OF DELETIONS: 2

ANALYSIS OF STATEMENTS ALTERED

NUMBER OF LINES OF COMMENTS : 5
IDENTIFICATION DIVISION : 2
SPACING PURPOSES : 0
USEFUL COMMENTS : 3

NUMBER OF ENVIRONMENT STATEMENTS : 2
CONFIGURATION SECTION : 0
SOURCE-COMPUTER : 0
OBJECT-COMPUTER : 0
COMPUTER SPECIFICATION : 0
SPECIAL NAMES : 0
SPECIAL NAME ASSIGNMENT : 0
INPUT-OUTPUT SECTION : 0
FILE-CONTROL : 0
SELECT : 2
<table>
<thead>
<tr>
<th>Statement Type</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Declarations : 0</td>
<td></td>
</tr>
<tr>
<td>Sections : 0</td>
<td></td>
</tr>
<tr>
<td>FD : 0</td>
<td></td>
</tr>
<tr>
<td>Declarations : 0</td>
<td></td>
</tr>
<tr>
<td>Value Clauses : 0</td>
<td></td>
</tr>
<tr>
<td>Redefines : 0</td>
<td></td>
</tr>
<tr>
<td>Renames : 0</td>
<td></td>
</tr>
<tr>
<td>Number of Assignments : 0</td>
<td></td>
</tr>
</tbody>
</table>
| **Note that the above total includes Value Clauses**
| Move : 0                       |       |
| Add : 0                        |       |
| Subtract : 0                   |       |
| Multiply : 0                   |       |
| Divide : 0                     |       |
| Compute : 0                    |       |
| Number of Conditionals : 0     |       |
| If : 0                         |       |
| Else : 0                       |       |
| On : 0                         |       |
| On Size Error : 0              |       |
| At End : 0                     |       |
| Number of Branchings : 0       |       |
| Call : 0                       |       |
| Perform : 0                    |       |
| Go To : 0                      |       |
| Next : 0                       |       |
| Exit : 0                       |       |
| Stop : 0                       |       |
| Number of Input/Output : 0     |       |
| Delete : 0                     |       |
| Display : 0                    |       |
| Open : 0                       |       |
| Close : 0                      |       |
| Read : 0                       |       |
| Rewrite : 0                    |       |
| Write : 0                      |       |
| Accept : 0                     |       |
| Number of Labels : 0           |       |
| Number of Other Statements : 0 |       |
| Copy : 0                       |       |
| Alter : 0                      |       |
| Transform : 0                  |       |
| Examine : 0                    |       |
| Inspect : 0                    |       |
| Search : 0                     |       |
| Sort : 0                       |       |
| Set : 0                        |       |
| CICS Exec : 0                  |       |
| Goback : 0                     |       |

ANALYSIS OF STATEMENTS DELETED
NUMBER OF LINES OF COMMENTS : 2
IDENTIFICATION DIVISION : 2
SPACING PURPOSES : 0
USEFUL COMMENTS : 0

NUMBER OF ENVIRONMENT STATEMENTS : 0
CONFIGURATION SECTION : 0
SOURCE-COMPUTER : 0
OBJECT-COMPUTER : 0
COMPUTER SPECIFICATION : 0
SPECIAL NAMES : 0
SPECIAL NAME ASSIGNMENT : 0
INPUT-OUTPUT SECTION : 0
FILE-CONTROL : 0
SELECT : 0

NUMBER OF DECLARATIONS : 0
SECTIONS : 0
FD : 0
DECLARATIONS : 0
VALUE CLAUSES : 0
REDEFINES : 0
RENAMEs : 0

NUMBER OF ASSIGNMENTS : 0
*** note that the above total includes VALUES CLAUSES ***
MOVE : 0
ADD : 0
SUBTRACT : 0
MULTIPLY : 0
DIVIDE : 0
COMPUTE : 0

NUMBER OF CONDITIONALS : 0
IF : 0
ELSE : 0
ON : 0
ON SIZE ERROR : 0
AT END : 0

NUMBER OF BRANCHINGS : 0
CALL : 0
PERFORM : 0
GO TO : 0
NEXT : 0
EXIT : 0
STOP : 0

NUMBER OF INPUT/OUTPUT : 0
DELETE : 0
DISPLAY : 0
OPEN : 0
CLOSE : 0
READ : 0
REWRITE : 0
WRITE : 0
ACCEPT : 0

NUMBER OF LABELS : 0

NUMBER OF OTHER STATEMENTS : 0
COPY : 0

D-4
**ANALYSIS OF STATEMENTS ADDED**

**TOTAL NUMBER OF ADDED STATEMENTS:** 3

**NUMBER OF LINES OF COMMENTS:** 3
**IDENTIFICATION DIVISION:** 0
**SPACING PURPOSES:** 3
**USEFUL COMMENTS:** 0

**NUMBER OF ENVIRONMENT STATEMENTS:** 0
**CONFIGURATION SECTION:** 0
**SOURCE-COMPUTER:** 0
**OBJECT-COMPUTER:** 0
**COMPUTER SPECIFICATION:** 0
**SPECIAL NAMES:** 0
**SPECIAL NAME ASSIGNMENT:** 0
**INPUT-OUTPUT SECTION:** 0
**FILE-CONTROL:** 0
**SELECT:** 0

**NUMBER OF DECLARATIONS:** 0
**SECTIONS:** 0
**FD:** 0
**DECLARATIONS:** 0
**VALUE CLAUSES:** 0
**REDEFINES:** 0
**RENAAMES:** 0

**NUMBER OF ASSIGNMENTS:** 0

***note that the above total includes VALUES CLAUSES***

**MOVE:** 0
**ADD:** 0
**SUBTRACT:** 0
**MULTIPLY:** 0
**DIVIDE:** 0
**COMPUTE:** 0

**NUMBER OF CONDITIONALS:** 0
**IF:** 0
**ELSE:** 0
**ON:** 0
**ON SIZE ERROR:** 0
**AT END:** 0

**NUMBER OF BRANCHINGS:** 0
**CALL:** 0

---

D-5
ORIGINAL LINE < Comment VERSION 1.
ALTERED TO

NEWLINE > Comment VERSION 2.

ORIGINAL LINE <
ORIGINAL LINE <

ALTERED TO

NEWLINE >
NEWLINE >

ORIGINAL LINE <
ORIGINAL LINE <

ALTERED TO

NEWLINE > IF C-AMNTX NOT NUMERIC GENERATE ERROR-DETAIL
NEWLINE > MOVE ZERO TO C-AMNTX.

LIST THE DELETED STATEMENTS

Comment * PROGRAM WAS WRITTEN FROM PRG JEX 10-6-64 MVS
Comment

• CONVERSION, MODIFIED SELECT CLAUSE

-----------------------------

LIST THE ADDED STATEMENTS

•

•

•
Appendix E. The Shell Program Classify

The input file is the result from executing the shell program Maintain. It generates 4 temporary files.
"Alter1" stores old altered statements and "alter2" stores new altered statements. Deleted statements are put into "delfile" and added statements in "addfile".

The output lists six types of maintenance from files. The six types are Correction, Adaption, Retrenchment, Retrieving, Pretty printing, and Documentation. The plus and minus signs in pretty printing and documentation stand for the increasing or decreasing of the numbers.

echo ANALYZING FOR : $1 > $1.out
echo ------------------------------------------------- > out

# insert special characters to original file
awk '{
BEGIN { line0 = 0 }
/LIST THE ADDED STATEMENTS/ { print "$" >> "copyfile" }
/LIST THE DELETED STATEMENTS/ { print "$" >> "copyfile" }
[ print $0 >> "copyfile" ]
}

END ( print "$$" >> "copyfile" }

# line1: old numbers of altered statement; line2: new numbers of altered
# noline1: old numbers of altered statements in a block
# noblock: numbers of block being altered
# noalter: numbers of block counted as big changed in size
awk '{
BEGIN { line1=0; line2=0; noline1=0; noline2=0; flag =0; noblock=0; noalter=0}
/ANALYSIS OF ALTERED STATEMENTS/, /\#\#\#$/ { 
  if ($3 == "<")
    line++
    i = 1
    while (i <= NF)
      [ print $i >> "alter1"; i++ ]
    #
    if (flag == 0)
      [ noline1++ ]
    # compute the block which changes rapidly in size
    if (flag == 1)
      [ if ((noline1 > 10) && (noline2 > 10))
        [ if (noline1 > noline2)
          [ div1 = noline1 / noline2 ]
        else
          [ div1 = noline2 / noline1 ]
        if (div1 > 2)
          [ noalter++ ]
        ]
      else
        [ if (noline1 > noline2)
          [ div2 = noline1 / noline2
dif2 = noline1 - noline2 ]
        else
          [ div2 = noline2 / noline1
dif2 = noline2 - noline1 ]
        if ((div2 > 5) || (dif2 > 5))
          [ noalter++ ]
        ]
      ]
    ]
  else
    [ if (noline1 > noline2)
      [ div2 = noline1 / noline2
dif2 = noline1 - noline2 ]
    else
      [ div2 = noline2 / noline1
dif2 = noline2 - noline1 ]
    if ((div2 > 5) || (dif2 > 5))
      [ noalter++ ]
  ]
  E-1
# reset to 0 after done a block
flag = 0
noline1 = 1
noline2 = 0
}
if ($2 == ">")
    line2++
    # store new altered statements to alter2 file
    j = 3
    while (j <= NF)
        { print "$j >> "alter2"; j++ }
    if (flag == 0)
        { noline2 = 0; flag = 1 }
    if (flag == 1)
        { noline2++ }
    if (($1 == "ALTERED") && ($2 == "TO") && (NF == 2))
        { noblock++ }
}
    # store deleted statements to delfile
/LIST THE DELETED STATEMENTS/, /\%\%\%/ { 
    if (($3="DELETED") && ($0="$%$%") && (NF==0))
        { }
    else
        { print "$0 >> "delfile" }
    }
    # store added statements to addfile
/LIST THE ADDED STATEMENTS/, /\%\%\%/ { 
    if (($3="ADDED") && ($0="$%$%") && (NF==0))
        { }
    else
        { print "$0 >> "addfile" }
    }
END { if (noblock != 0)
    { print "\n (( Altered ))" >> "out"
        print " number of original line : " line1 >> "out"
        print " number of new line : " line2 >> "out"
        print " number of block altered : " noblock >> "out"
        if (noalter > 5)
            { print " <Adaptive>" >> "out"}
        else
            { print " <Corrective>" >> "out" }
    }
}
if (( test -f alter1) && (test -f alter2))
then
    diff alter1 alter2 > difference
    sed 'a/\%/ /g
        s/a/ a/g
        s/c/ c/g
        s/d/ d/g' difference > result
    awk '{ BEGIN { NoAdd = 0; NoDel = 0; NoRetrench = 0; NoRetrieve = 0;
             alterfrom = 0; alterto = 0;
             DelDocument = 0; AddDocument = 0; DelPrint = 0; AddPrint = 0;
             aflag = 0; cflag = 0; dflag = 0; c1flag =0; c2flag =0 }
E-2
{ if ($2 == "a")
  { aflag = 1; cflag = 0; dflag = 0 }
if (($2 == "a") \|| ($3 == "e"))
  { cflag = 1; cflag = 0; c2flag = 0; c3flag = 0; aflag = 0; dflag = 0 }
if (($2 == "d") \|| ($3 == "d"))
  { dflag = 1; aflag = 0; cflag = 0 }
if ( aflag == 1 )
  { if ($1 == ">")
      [ NoRetrench++ ]
    else
      { NoAdd++ } ]
if ( cflag == 1 )
  { if ($0 == "<")
      [ DelPrint++ ]
    if (cflag == 0 )
      { cflag = 1 } ]
    prestar = 1 ]
if (($1 == ">") \&\& ($2 == ">") \&\& (cflag == 1))
  { cflag = 0; DelDocument++ ; prestar = 0 ]
if (($0 == "<" ) \&\& (cflag == 1) \&\& (prestar == 1))
  { NoRetrieve++ ]
if (($0 == "<") \&\& (cflag == 1) \&\& (prestar == 0))
  { DelPrint++ ]
if ( $0 == ">")
  { if (c2flag == 1)
      [ AddPrint++
        if (c3flag == 0)
          { AddPrint++; c3flag = 1 } ]
    if (c2flag == 0)
      { c2flag = 1 } ]
if (($1 == ">") \&\& ($2 == ">") \&\& (c2flag == 1))
  { c2flag = 0; AddDocument++ ]
if (($1 == ">") \&\& ($2 == ">") \&\& (cflag == 0))
  { alterfroai++ ]
if (($1 == ">") \&\& ($2 == ">") \&\& (c2flag == 0))
  { alterto++ ]
if ( dflag == 1 )
  { if ($1 == ">")
      [ if ($2 == "<")
        [ NoRetrieve++ ]
      else
        { NoDel++ } ] ]
] END { if ( NoRetrench > 0 )
  [ print "<Retrenchment> : number = " NoRetrench >> "out" ]
if ( NoRetrieve > 0 )
  [ print "<Retrieving> : number = " NoRetrieve >> "out" ]
if (AddDocument > 0)
  [ print "<Documentation> : number = " AddDocument >> "out"]
if (DelDocument > 0)
  [ print "<Documentation> : number = " DelDocument >> "out"]
if (AddPrint > 0)
  [ print "<Pretty Printing> : number = " AddPrint >> "out"]
if (DelPrint > 0)
  [ print "<Pretty Printing> : number = " DelPrint >> "out"]
} result
rm alter1 alter2 result difference
} fi
If there exist deleted statements

```bash
if ( test -f delfile )
then

awk 'BEGIN { NoComment = 0; NoDlrtn = 0; DelDoc = 0; DelDoc = 0 }
    if ($1 == "Comment")
        [ NoComment++ ]
    if ($1 == "Dlrtn")
        [ NoDlrtn++ ]
    if ($1 == "*" && (NF == 1))
        [ DelPnt++ ]
    if ($1 == "*" && (NF > 1))
        [ DelDoc++ ]
}
END { print "n ( Deleted ) >> "out" ]
    if (NoComment > 0)
        print "Comment is deleted. number: " NoComment >> "out"
    if (NoDlrtn > 0)
        print "Declaration is deleted. number: " NoDlrtn >> "out"
    if (DelPnt > 0)
        print "<Pretty printing-> : number " DelPnt >> "out"
    if (DelDoc > 0)
        print "<Documentation-> : number " DelDoc >> "out"
    NoOther = NR - NoComment - NoDlrtn - DelPnt - DelDoc
    if (NoOther > 10)
        print "<Adaptive>: number " NoOther >> "out"
    if ((NoOther <= 10) && (NoOther > 0))
        print "<Corrective>: number " NoOther >> "out"
}
'delfile
rm delfile

fi

# if there exist added statements

if ( test -f addfile )
then

awk 'BEGIN { NoComment = 0; NoDlrtn = 0; AddDoc = 0; AddPnt = 0 }
    if ($1 == "Comment")
        [ NoComment++ ]
    if ($1 == "Dlrtn")
        [ NoDlrtn++ ]
    if ($1 == "*" && (NF == 1))
        [ AddPnt++ ]
    if ($1 == "*" && (NF > 1))
        [ AddDoc++ ]
}
END { [ print "n ( Added ) >> "out" ]
    if (NoComment > 0)
        print "Comment is added. number: " NoComment >> "out"
    if (NoDlrtn > 0)
        print "Declaration is added. number: " NoDlrtn >> "out"
    if (AddPnt > 0)
        print "<Pretty Printing-> : number " AddPnt >> "out"
    if (AddDoc > 0)
        print "<Documentation-> : number " AddDoc >> "out"
    NoOther = NR - NoComment - NoDlrtn - AddPnt - AddDoc
    if (NoOther > 10)
        print "<Adaptive>: number " NoOther >> "out"
    if ((NoOther <= 10) && (NoOther > 0))
        print "<Corrective>: number " NoOther >> "out"
}
' addfile
rm addfile

fi
```

E-4
Appendix F. Result from Running Classify

ANALYZING FOR : COBOL.1.listing
----------------------------------------

(( Altered ))
  number of original line : 7
  number of new line : 6
  number of block altered : 3
  <Corrective> : number = 3

(( Deleted ))
  The total number of deleted statements is 2
  Comment is deleted. number: 2
  <Corrective> : number 2

(( Added ))
  The total number of added statements is 3
  <Pretty Printing+> : number 3

F-1
ANALYZING CHANGES IN COBOL PROGRAMS
DURING MAINTENANCE

by

IE-HONG LIN

B.S., NATIONAL CHANG KUNG UNIVERSITY
Tainan, Taiwan 1980

M.S., KANSAS STATE UNIVERSITY
Manhattan, Kansas 1985

-------------------------------

AN ABSTRACT OF A THESIS

submitted in partial fulfillment of the
requirement for the degree

MASTER OF SCIENCE

Department of Computing and Information Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1988
Abstract

Software maintenance has become the most expensive phase. To maintain software, managers need methods to monitor the process in order to predict where changes will occur. Knowing the types of maintenance help managers in managing the maintenance.

The study presents a method to classify types of maintenance. The work focuses on analyzing COBOL programs and classifying different types of maintenance. The shell program Maintain was written as a tool to analyze two sequential versions on a program. Program set A, from a Kansas company, was first introduced to analyze. Six types of maintenance were identified from the results. They are corrective, adaptive, retrenchment, retrieving, pretty printing, and documentation. The classification rules were then converted into the second shell program Classify. Program set B, from data processing environment, was finally verified with the program Maintain and Classify to test the results.

The presented method is successfully in classifying types of maintenance from empirical data that changes between two versions of a program. In particular, the method allows managers to identify types of maintenance that have been done and evaluate the effort by means of the classification rules.