DOCUMENT RESUME

ED 054 705

Price, James D.


BR-9-7722

Jun 71

OEC-0-9-097722-4411

232p.

*Algorithms; Computational Linguistics; *Computer Programs; Deep Structure; Flow Charts; Grammar; *Hebrew; Language Patterns; Language Research; Language Universals; Logic; Phrase Structure; Semitic Languages; Sentence Structure; *Structural Analysis; Syntax; *Transformation Generative Grammar; Transformation Theory (Language)

ABSTRACT

The final part of a four-part report of research on the development of a computerized, phrase-structure grammar of modern Hebrew describes the computerized algorithm for analyzing the sentences generated based on a complex-constituent-phrase structure grammar. The first section here discusses a structural model for modern Hebrew; the second provides a detailed description of the procedure for analyzing sentences; the third gives a detailed description of the computer program for the procedure; the last section describes the tests and the verifications of the algorithm. Appendix A lists the grammar rules of Hebrew syntax for the analysis of sentences. Appendix B gives the source-language listing of the computer program and subprograms used. Appendix C provides a sample output. Appendix D gives examples of the exhaustive syntactic analysis of the computer. For related reports see FL 002 627, FL 002 628, and FL 002 629. (VM)
A COMPUTERIZED PHRASE-STRUCTURE GRAMMAR (MODERN HEBREW)

PART IV

James D. Price
Franklin Institute Research Laboratories
20th and Race Street
Philadelphia, Penn. 19103

June 1971
A COMPUTERIZED PHRASE-STRUCTURE GRAMMAR (MODERN HEBREW)

PART IV

AN ALGORITHM FOR ANALYZING HEBREW SENTENCES

James D. Frice

Franklin Institute Research Laboratories
Philadelphia, Pennsylvania 19103

June 1971

The research reported herein was performed pursuant to a contract with the Office of Education, U. S. Department of Health, Education, Welfare.

U. S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE
Office of Education
Institute of International Studies
ABSTRACT

This is the fourth part of a four-part report of research for the development of a Computerized Phrase-Structure Grammar of Modern Hebrew. This part describes a computerized algorithm for analyzing sentences in modern Hebrew which is based on a generalized complex-constituent phrase-structure grammar (defined in Part I) as it was applied to the syntax of modern Hebrew (described in Part II).

A computer program of the algorithm is described which includes, for the main program and each subprogram, (1) a flow chart, (2) a written description of its operation, and (3) a source language listing in FORTRAN IV.

The algorithm was made operational in a UNIVAC 1108 computer and used to systematically test the grammar of modern Hebrew syntax by analyzing sentences in the language. A total of 26 sentences were analyzed in the process of which 57 of the 17 grammar rules were tested. The tests demonstrate the capability of the algorithm for analyzing sentences. Due to limitations on the predictive logic of the algorithm, some difficulties were experienced in the analysis of complex sentences. However, the algorithm proved to be a valuable tool for testing, validating, and, in numerous cases, correcting the grammar rules.

Three aspects of the algorithm need further attention: (1) several computations must be freed from dependence on the "content" of the source grammar, (2) the input data preparation should be simplified, and (3) the predictive logic should be extended to a greater depth. With these modifications the algorithm can be used to generate sentences in other Semitic languages whenever grammars become available, and to aid in the development of such grammars.
<table>
<thead>
<tr>
<th>Part</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part IV</td>
<td>An Algorithm for Analyzing Hebrew Sentences</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>A Structural Model of Hebrew Sentences</td>
<td>4-1</td>
</tr>
<tr>
<td></td>
<td>4.1.1 The Mapping Functions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.1.1.1 The Transliteration</td>
<td>4-2</td>
</tr>
<tr>
<td></td>
<td>4.1.1.2 The Symbols</td>
<td>4-2</td>
</tr>
<tr>
<td></td>
<td>4.1.1.3 The Rules</td>
<td>4-6</td>
</tr>
<tr>
<td></td>
<td>4.1.1.4 The Restraints</td>
<td>4-6</td>
</tr>
<tr>
<td></td>
<td>4.1.1.5 The Symbol Names</td>
<td>4-8</td>
</tr>
<tr>
<td></td>
<td>4.1.1.6 The Analysis Predicates and Index</td>
<td>4-8</td>
</tr>
<tr>
<td></td>
<td>4.1.1.7 The Feature Values</td>
<td>4-8</td>
</tr>
<tr>
<td>4.1.2</td>
<td>The Variables</td>
<td>4-8</td>
</tr>
<tr>
<td>4.2</td>
<td>The Procedure</td>
<td>4-19</td>
</tr>
<tr>
<td>4.3</td>
<td>Computer Program of the Algorithm</td>
<td>4-25</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Flow-Chart Description of Computer Program</td>
<td>4-25</td>
</tr>
<tr>
<td></td>
<td>4.3.1.1 Main Program ANALYZ</td>
<td>4-25</td>
</tr>
<tr>
<td></td>
<td>4.3.1.2 Subprogram ALPHA.</td>
<td>4-35</td>
</tr>
<tr>
<td></td>
<td>4.3.1.3 Subprogram DIAGRM</td>
<td>4-35</td>
</tr>
<tr>
<td></td>
<td>4.3.1.4 Subprogram LIMIT.</td>
<td>4-47</td>
</tr>
<tr>
<td></td>
<td>4.3.1.5 Subprogram MACHER</td>
<td>4-47</td>
</tr>
<tr>
<td></td>
<td>4.3.1.6 Subprogram OUTPUT</td>
<td>4-50</td>
</tr>
<tr>
<td></td>
<td>4.3.1.7 Subprogram PARSE.</td>
<td>4-50</td>
</tr>
<tr>
<td></td>
<td>4.3.1.8 Subprogram PROPH1</td>
<td>4-52</td>
</tr>
<tr>
<td></td>
<td>4.3.1.9 Subprogram PROPH2</td>
<td>4-52</td>
</tr>
<tr>
<td></td>
<td>4.3.1.10 Subprogram RERITE</td>
<td>4-64</td>
</tr>
<tr>
<td></td>
<td>4.3.1.11 Subprogram RULENO</td>
<td>4-66</td>
</tr>
<tr>
<td></td>
<td>4.3.1.12 Subprogram SYMACH</td>
<td>4-74</td>
</tr>
<tr>
<td></td>
<td>4.3.1.13 Subprogram VARATT</td>
<td>4-81</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Input Map for Computer Program</td>
<td>4-83</td>
</tr>
<tr>
<td>4.3.2.1</td>
<td>Program Option Card</td>
<td>4-83</td>
</tr>
<tr>
<td>4.3.2.2</td>
<td>Program Grammar Cards</td>
<td>4-87</td>
</tr>
<tr>
<td></td>
<td>4.3.2.2.1 Transliteration Table:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Card 2-1.</td>
<td>4-87</td>
</tr>
<tr>
<td></td>
<td>4.3.2.2.2 Grammar Rules: Cards 2-2.</td>
<td>4-87</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (CONT.)

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.2.2.3</td>
<td>Restraint Matrix: Cards 2-3.</td>
<td>4-89</td>
</tr>
<tr>
<td>4.3.2.2.4</td>
<td>The Table of Symbols: Cards 2-4.</td>
<td>4-90</td>
</tr>
<tr>
<td>4.3.2.2.5</td>
<td>The Table of Symbol Names Cards 2-5.</td>
<td>4-90</td>
</tr>
<tr>
<td>4.3.2.2.6</td>
<td>The Predicate Index: Cards 2-6.</td>
<td>4-91</td>
</tr>
<tr>
<td>4.3.2.2.7</td>
<td>The Analysis Predicates: Cards 2-7.</td>
<td>4-91</td>
</tr>
<tr>
<td>4.3.2.2.8</td>
<td>The Feature Values: Cards 2-8.</td>
<td>4-92</td>
</tr>
<tr>
<td>4.3.3</td>
<td>The Sentence Description Cards.</td>
<td>4-92</td>
</tr>
<tr>
<td>4.3.3.1</td>
<td>Hebrew Sentence: Card 3-1</td>
<td>4-92</td>
</tr>
<tr>
<td>4.3.3.2</td>
<td>Word Description: Cards 3-2</td>
<td>4-93</td>
</tr>
<tr>
<td>4.4</td>
<td>Test and Demonstration of Algorithm.</td>
<td>4-95</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Sentences Analyzed by Algorithm</td>
<td>4-95</td>
</tr>
<tr>
<td>4.4.2</td>
<td>The Algorithm Tested by Computer.</td>
<td>4-97</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
<td>4-102</td>
</tr>
</tbody>
</table>

APPENDIX A - Listing of the Rules of the Grammar of Hebrew Syntax for Analysis of Sentences | 4-A |

APPENDIX B - Source Language Listing of Computer Program ANALYZ and Associated Subprograms | 4-B |

APPENDIX C - Sample Output from Subprogram OUTPUT | 4-C |

APPENDIX D - Examples of Exhaustive Syntactic Analysis by Computer (with Associated Tree Diagrams) Tree Diagrams are Produced by Subprogram DIAGRM; Syntax Analysis Statements are Produced by Subprogram PARSE. | 4-D |
<table>
<thead>
<tr>
<th>Figure No.</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Flow Chart of Procedure</td>
<td>4-22</td>
</tr>
<tr>
<td>4.2</td>
<td>Map of Procedure</td>
<td>4-26</td>
</tr>
<tr>
<td>4.3</td>
<td>Program ANALYZ</td>
<td>4-27</td>
</tr>
<tr>
<td>4.4</td>
<td>Subprogram ALPHA</td>
<td>4-36</td>
</tr>
<tr>
<td>4.5</td>
<td>Subprogram DIAGRAM</td>
<td>4-39</td>
</tr>
<tr>
<td>4.6</td>
<td>Subprogram LIMIT</td>
<td>4-48</td>
</tr>
<tr>
<td>4.7</td>
<td>Subprogram MACHER</td>
<td>4-49</td>
</tr>
<tr>
<td>4.8</td>
<td>Subprogram OUTPUT</td>
<td>4-51</td>
</tr>
<tr>
<td>4.9</td>
<td>Subprogram PARSE</td>
<td>4-53</td>
</tr>
<tr>
<td>4.10</td>
<td>Subprogram PROPH1</td>
<td>4-59</td>
</tr>
<tr>
<td>4.11</td>
<td>Subprogram PROPH2</td>
<td>4-62</td>
</tr>
<tr>
<td>4.12</td>
<td>Subprogram RERITE</td>
<td>4-67</td>
</tr>
<tr>
<td>4.13</td>
<td>Subprogram RULENO</td>
<td>4-75</td>
</tr>
<tr>
<td>4.14</td>
<td>Subprogram SYMACH</td>
<td>4-80</td>
</tr>
<tr>
<td>4.15</td>
<td>Subprogram VARATT</td>
<td>4-84</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
<td>Alpha-Numeric Transliteration.</td>
<td>4-3</td>
</tr>
<tr>
<td>4-2</td>
<td>List of Symbols.</td>
<td>4-4</td>
</tr>
<tr>
<td>4-3</td>
<td>List of Restraints.</td>
<td>4-7</td>
</tr>
<tr>
<td>4-4</td>
<td>List of Symbol Names</td>
<td>4-9</td>
</tr>
<tr>
<td>4-5</td>
<td>List of Analysis Predicates.</td>
<td>4-11</td>
</tr>
<tr>
<td>4-6</td>
<td>Index of Analysis Statements</td>
<td>4-16</td>
</tr>
<tr>
<td>4-7</td>
<td>List of Feature Values</td>
<td>4-18</td>
</tr>
<tr>
<td>4-8</td>
<td>List of Variables</td>
<td>4-20</td>
</tr>
<tr>
<td>4-9</td>
<td>Restraint Matrix Data.</td>
<td>4-90</td>
</tr>
<tr>
<td>4-10</td>
<td>Table of Symbols</td>
<td>4-91</td>
</tr>
<tr>
<td>4-11</td>
<td>Sentences Illustrating Grammar Symbols</td>
<td>4-98</td>
</tr>
</tbody>
</table>
PART IV
AN ALGORITHM FOR ANALYZING HEBREW SENTENCES

This part of the report describes a computerized algorithm for analyzing Hebrew sentences, and algorithm essentially complete in that it defines all the procedures and associated computer programs for analyzing sentences in modern Hebrew, but incomplete in that improvements can be made in the grammar upon which it operates. Because the algorithm is dependent only on the "form" of the grammar and not on its "content"\(^1\), it may be used for training research workers in the field of computational linguistics without being limited to the Hebrew language.

The algorithm consists of: (1) a structural model of the language, (2) a procedure for using the model to analyze sentences in the language, and (3) a program for performing the procedure by means of a computer.

The first section discusses the structural model of modern Hebrew. The second section provides a detailed description of the procedure for analyzing sentences. The third section provides a detailed description of the computer program of the procedure. The last section describes the tests and verifications of the algorithm.

4.1 A Structural Model of Hebrew Sentences

The structural model of modern Hebrew sentences consists of a complex-constituent phrase-structure grammar of modern Hebrew the generalized characteristics of which are defined in Part I of this report and the specific details for which are defined in Part II. Those details of the structural model required to explain the algorithm are presented in the appropriate places throughout the text. Reference should be made to Parts I and II for further information.

For the purposes of the algorithm, the structural model is viewed as the following:

(1) a set of mapping functions
(2) a set of variables

\(^1\)A few exceptions are noted throughout the text; these will be corrected in a subsequent revision.
4.1.1 The Mapping Functions

The mapping functions enable the user to define the specific details of the structural model that are peculiar to the given language (in this case Hebrew). The use of mapping functions makes the algorithm independent of the "content" of grammar it processes by enabling the user to define the "content" for a given language.

The mapping functions consist of the following:

1. a table of the transliteration
2. a table of symbols
3. a table of grammar rules
4. a table of restraints
5. a table of symbol names
6. a table of analysis predicates
7. an index of (6)
8. a table of feature values

4.1.1.1 The Transliteration

The table of transliteration provides a list of alphabetic and numeric characters of the given language. These data are used by the algorithm to transform the input characters to numbers for use in computation and to transform to resultant output numbers to their corresponding characters. Table 4-1 is a listing of the transliteration used for Hebrew (see Table 2-1, Part II, for Hebrew equivalents).

4.1.1.2 The Symbols

The table of symbols provides a list of the symbols used in the grammar of the given language. The list is used by the algorithm to transform the input symbol names of the grammar rules to numbers for use in computation and to transform the resultant output numbers to their corresponding symbol names for use in tree diagrams and output listings. Table 4-2 is a listing the symbols used for Hebrew.²

²See Table 2-2 and 2-3 of Part II for further description of the symbols.
<table>
<thead>
<tr>
<th>L</th>
<th>TRANSL(L)</th>
<th>IA</th>
<th>L</th>
<th>TRANSL(L)</th>
<th>IA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>26</td>
<td>O</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>27</td>
<td>P</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>28</td>
<td>&amp;</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>29</td>
<td>Q</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>30</td>
<td>R</td>
<td>29</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>31</td>
<td>S</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>6</td>
<td>32</td>
<td>T</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>7</td>
<td>33</td>
<td>(</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>8</td>
<td>34</td>
<td>)</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>9</td>
<td>35</td>
<td>,</td>
<td>34</td>
</tr>
<tr>
<td>11</td>
<td>A</td>
<td>10</td>
<td>36</td>
<td>:</td>
<td>35</td>
</tr>
<tr>
<td>12</td>
<td>B</td>
<td>11</td>
<td>37</td>
<td>?</td>
<td>36</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>12</td>
<td>38</td>
<td>.</td>
<td>37</td>
</tr>
<tr>
<td>14</td>
<td>G</td>
<td>13</td>
<td>39</td>
<td>!</td>
<td>38</td>
</tr>
<tr>
<td>15</td>
<td>H</td>
<td>14</td>
<td>40</td>
<td>–</td>
<td>39</td>
</tr>
<tr>
<td>16</td>
<td>W</td>
<td>15</td>
<td>41</td>
<td>(space)</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>Z</td>
<td>16</td>
<td>42</td>
<td>F</td>
<td>41</td>
</tr>
<tr>
<td>18</td>
<td>X</td>
<td>17</td>
<td>43</td>
<td>I</td>
<td>42</td>
</tr>
<tr>
<td>19</td>
<td>@</td>
<td>18</td>
<td>44</td>
<td>V</td>
<td>43</td>
</tr>
<tr>
<td>20</td>
<td>Y</td>
<td>19</td>
<td>45</td>
<td>*</td>
<td>44</td>
</tr>
<tr>
<td>21</td>
<td>K</td>
<td>20</td>
<td>46</td>
<td>#</td>
<td>45</td>
</tr>
<tr>
<td>22</td>
<td>L</td>
<td>21</td>
<td>47</td>
<td>Δ</td>
<td>46</td>
</tr>
<tr>
<td>23</td>
<td>M</td>
<td>22</td>
<td>48</td>
<td>'</td>
<td>47</td>
</tr>
<tr>
<td>24</td>
<td>N</td>
<td>23</td>
<td>49</td>
<td>?</td>
<td>48</td>
</tr>
<tr>
<td>25</td>
<td>C</td>
<td>24</td>
<td>50</td>
<td>?</td>
<td>49</td>
</tr>
<tr>
<td>SYMBOL No.</td>
<td>SYMBOL Name</td>
<td>SYMBOL No.</td>
<td>SYMBOL Name</td>
<td>SYMBOL No.</td>
<td>SYMBOL Name</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>------------</td>
<td>-------------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>Z</td>
<td>21</td>
<td>B_p</td>
<td>41</td>
<td>R_sp</td>
</tr>
<tr>
<td>2</td>
<td>V_qo</td>
<td>22</td>
<td>N_pvb</td>
<td>42</td>
<td>N_sp</td>
</tr>
<tr>
<td>3</td>
<td>A_pa</td>
<td>23</td>
<td>N_pa</td>
<td>43</td>
<td>N_op</td>
</tr>
<tr>
<td>4</td>
<td>A_p</td>
<td>24</td>
<td>N_pc</td>
<td>44</td>
<td>N_ip</td>
</tr>
<tr>
<td>5</td>
<td>A_s</td>
<td>25</td>
<td>N_ap</td>
<td>45</td>
<td>N_px</td>
</tr>
<tr>
<td>6</td>
<td>N_a</td>
<td>26</td>
<td>N_p</td>
<td>46</td>
<td>V_ma</td>
</tr>
<tr>
<td>7</td>
<td>S_qo</td>
<td>27</td>
<td>D_p_a</td>
<td>47</td>
<td>V_mb</td>
</tr>
<tr>
<td>8</td>
<td>N_s</td>
<td>28</td>
<td>D_pb</td>
<td>48</td>
<td>V_mc</td>
</tr>
<tr>
<td>9</td>
<td>R_d</td>
<td>29</td>
<td>D_pc</td>
<td>49</td>
<td>V_md</td>
</tr>
<tr>
<td>10</td>
<td>R_o</td>
<td>30</td>
<td>D_p</td>
<td>50</td>
<td>V_m</td>
</tr>
<tr>
<td>11</td>
<td>B_αα</td>
<td>31</td>
<td>E_a</td>
<td>51</td>
<td>V_mr</td>
</tr>
<tr>
<td>12</td>
<td>B_αβ</td>
<td>32</td>
<td>V_b</td>
<td>52</td>
<td>V_mi</td>
</tr>
<tr>
<td>13</td>
<td>B_αc</td>
<td>33</td>
<td>V_bb</td>
<td>53</td>
<td>V_p</td>
</tr>
<tr>
<td>14</td>
<td>B_αd</td>
<td>34</td>
<td>V_c</td>
<td>54</td>
<td>V_rb</td>
</tr>
<tr>
<td>15</td>
<td>B_αε</td>
<td>35</td>
<td>V_αα</td>
<td>55</td>
<td>V_ri</td>
</tr>
<tr>
<td>16</td>
<td>B_αf</td>
<td>36</td>
<td>V_α</td>
<td>56</td>
<td>N_v</td>
</tr>
<tr>
<td>17</td>
<td>B_βα</td>
<td>37</td>
<td>B_c</td>
<td>57</td>
<td>N_w</td>
</tr>
<tr>
<td>18</td>
<td>B_ββ</td>
<td>38</td>
<td>X_p</td>
<td>58</td>
<td>E_pb</td>
</tr>
<tr>
<td>19</td>
<td>B_β</td>
<td>39</td>
<td>N_o</td>
<td>59</td>
<td>E_pa</td>
</tr>
<tr>
<td>20</td>
<td>B_βc</td>
<td>40</td>
<td>D_pd</td>
<td>60</td>
<td>E_p</td>
</tr>
</tbody>
</table>
Table 4-2 (Continued)

LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>SYMBOL No.</th>
<th>SYMBOL Name</th>
<th>SYMBOL No.</th>
<th>SYMBOL Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>61</td>
<td>S&lt;sub&gt;aa&lt;/sub&gt;</td>
<td>81</td>
<td>E</td>
</tr>
<tr>
<td>62</td>
<td>S&lt;sub&gt;ab&lt;/sub&gt;</td>
<td>82</td>
<td>G</td>
</tr>
<tr>
<td>63</td>
<td>S&lt;sub&gt;ac&lt;/sub&gt;</td>
<td>83</td>
<td>H</td>
</tr>
<tr>
<td>64</td>
<td>S&lt;sub&gt;a&lt;/sub&gt;</td>
<td>84</td>
<td>I</td>
</tr>
<tr>
<td>65</td>
<td>S&lt;sub&gt;ro&lt;/sub&gt;</td>
<td>85</td>
<td>J</td>
</tr>
<tr>
<td>66</td>
<td>S&lt;sub&gt;ri&lt;/sub&gt;</td>
<td>86</td>
<td>L</td>
</tr>
<tr>
<td>67</td>
<td>R&lt;sub&gt;g&lt;/sub&gt;</td>
<td>87</td>
<td>N</td>
</tr>
<tr>
<td>68</td>
<td>K&lt;sub&gt;n&lt;/sub&gt;</td>
<td>88</td>
<td>O</td>
</tr>
<tr>
<td>69</td>
<td>K&lt;sub&gt;c&lt;/sub&gt;</td>
<td>89</td>
<td>P</td>
</tr>
<tr>
<td>70</td>
<td>K&lt;sub&gt;k&lt;/sub&gt;</td>
<td>90</td>
<td>Q</td>
</tr>
<tr>
<td>71</td>
<td>K&lt;sub&gt;i&lt;/sub&gt;</td>
<td>91</td>
<td>R</td>
</tr>
<tr>
<td>72</td>
<td>K&lt;sub&gt;d&lt;/sub&gt;</td>
<td>92</td>
<td>T</td>
</tr>
<tr>
<td>73</td>
<td>S&lt;sub&gt;d&lt;/sub&gt;</td>
<td>93</td>
<td>U</td>
</tr>
<tr>
<td>74</td>
<td>S</td>
<td>94</td>
<td>V</td>
</tr>
<tr>
<td>75</td>
<td>S&lt;sub&gt;i&lt;/sub&gt;</td>
<td>95</td>
<td>W</td>
</tr>
<tr>
<td>76</td>
<td>c</td>
<td>96</td>
<td>Y</td>
</tr>
<tr>
<td>77</td>
<td>A</td>
<td>97</td>
<td>F</td>
</tr>
<tr>
<td>78</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.1.1.3 The Rules

The table of grammar rules provides a list of the rules of the complex-constituent phrase-structure grammar of the given language. Appendix A of this part of the report contains a listing of the grammar rules used for Hebrew. The rules of the analysis grammar differ from those of the synthesis grammar in two respects:

(1) the left side of the analysis rules is the right side of the synthesis rules and \textit{visa versa}.

(2) synthesis rules that have an optimal symbol as the first element of the right side must have two corresponding analysis rules, one with the first symbol as mandatory, and one without the first symbol.

4.1.1.4 The Restraints

The table of restraints provides a list of limitations to be used by the rules of the grammar. The grammar specifies a certain horizontal row of the table to limit the value of a given symbol subscript. The restraints are interpreted as follows:

(1) The first number in the row specifies the number of restraint values in the row.

(2) If the numbers in the specified row are positive, the value of the given subscript must be one of the numbers in the row.

(3) If the numbers in the specified row are negative, the value of the given subscript must not be one of the numbers in the row.

Table 4-3 is a list of the restraints used in the grammar of Hebrew.

\footnotesize{Reference should be made to Section 2.2.3 of Part II of this report for a detailed description of the rules.}
Table 4-3

LIST OF RESTRAINTS

<table>
<thead>
<tr>
<th>Row No.</th>
<th>No. of Items</th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>-1</td>
<td>-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
</tr>
</tbody>
</table>
4.1.1.5 The Symbol Names

The table of symbol names provides a list of the names assigned to the symbols of the grammar. The list is used by the algorithm for constructing analysis statements about the sentences being analyzed. The analysis statements are of the general form

\[ N_s + N_a \]

where \( N_s \) stands for a symbol name and its associated derivational history and \( N_a \) stands for an analysis predicate (see next section). For example, the algorithm may construct the statement "The basic past-nom. adjective phrase (APA) expresses the superlative degree," the underlined part of which is the symbol name \( N_s \) the remainder is the analysis predicate. Table 4-4 is a listing of the symbol names used for Hebrew.

4.1.1.6 The Analysis Predicates and Index

The table of analysis predicates provides a list of the syntactic functions of all the categories of the symbols used by the grammar. The list is used by the algorithm to serve as the predicate of analysis statements (see example in the previous section). Table 4-5 is a listing of the analysis predicates for Hebrew. Table 4-6 is an index of the analysis predicates that maps the correspondence between a given class of a symbol and the associated predicate. For example, Symbol 3 class 3 is associated with predicate 5.

4.1.1.7 The Feature Values

The table of feature values provides a list of the various semantic values that a given linguistic feature of the symbols may assume. The table maps the correspondence of the numerical values of the symbol subscripts with the semantic value of the associated linguistic feature. Table 4-7 is the table of feature values for Hebrew. For example, the table specifies for subscript \( n=2 \) that this corresponds to the semantic value dual for the linguistic feature number. The algorithm uses these data to exhaustively define the surface structure elements in analysis statements.

4.1.2 The Variables

The set of variables consist of 29 subscripts on the set of symbols of the grammar that enable the user to define the sentence to be analyzed by the algorithm. The algorithm uses three types of variables:

(a) Fixed variables -- those with values fixed by the rules of the grammar.

(b) Dependent variables -- those the value of which are computed by the grammar.

(c) Independent variables -- those the value of which the user defines in the process of describing the sentence being analyzed.
<table>
<thead>
<tr>
<th>Symbol No.</th>
<th>Symbol Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PREPOSITIONAL PRON. PHRASE(Z)</td>
</tr>
<tr>
<td>2</td>
<td>OBJECTIVE-INTEROG. VERB PHRASE(VQO)</td>
</tr>
<tr>
<td>3</td>
<td>BASIC POST-NOM. ADJECTIVE PHRASE(APA)</td>
</tr>
<tr>
<td>4</td>
<td>POST-NOMINAL ADJECTIVE PHRASE(AP)</td>
</tr>
<tr>
<td>5</td>
<td>ADJECTIVAL POSSESS. PHRASE(AS)</td>
</tr>
<tr>
<td>6</td>
<td>BASIC NOUN PHRASE(NA)</td>
</tr>
<tr>
<td>7</td>
<td>OBJECTIVE INTEROGATIVE PHRASE(SQO)</td>
</tr>
<tr>
<td>8</td>
<td>POSSESS.-PRON. NOUN PHRASE(NS)</td>
</tr>
<tr>
<td>9</td>
<td>BASIC DEMONSTRATIVE PRON. PHRASE(RD)</td>
</tr>
<tr>
<td>10</td>
<td>DIRECT-OBJECT PRON. PHRASE(RO)</td>
</tr>
<tr>
<td>11</td>
<td>UNITS NUMBER PHRASE(BAA)</td>
</tr>
<tr>
<td>12</td>
<td>TENS NUMBER PHRASE(BAB)</td>
</tr>
<tr>
<td>13</td>
<td>TEENS NUMBER PHRASE(BAC)</td>
</tr>
<tr>
<td>14</td>
<td>MULTI-TENS NUMBER PHRASE(BAD)</td>
</tr>
<tr>
<td>15</td>
<td>HUNDREDS NUMBER PHRASE(BAE)</td>
</tr>
<tr>
<td>16</td>
<td>THOUSANDS NUMBER PHRASE(BAF)</td>
</tr>
<tr>
<td>17</td>
<td>1-TO-99 NUMBER PHRASE(BBA)</td>
</tr>
<tr>
<td>18</td>
<td>100-TO-999 NUMBER PHRASE(BBB)</td>
</tr>
<tr>
<td>19</td>
<td>1-TO-999 NUMBER PHRASE(BA)</td>
</tr>
<tr>
<td>20</td>
<td>1000-TO-9999 NUMBER PHRASE(BBC)</td>
</tr>
<tr>
<td>21</td>
<td>1-TO-9999 NUMBER PHRASE(BP)</td>
</tr>
<tr>
<td>22</td>
<td>SIMPLE NOUN PHRASE(NPB)</td>
</tr>
<tr>
<td>23</td>
<td>REGULAR NOUN PHRASE(NPA)</td>
</tr>
<tr>
<td>24</td>
<td>APPOSITIONAL NOUN PHRASE(NPC)</td>
</tr>
<tr>
<td>25</td>
<td>APPOSITIONAL PHRASE(NAP)</td>
</tr>
<tr>
<td>26</td>
<td>GENERAL NOUN PHRASE(NP)</td>
</tr>
<tr>
<td>27</td>
<td>SPECIFIC QUANTITY ADVERB PHRASE(DPA)</td>
</tr>
<tr>
<td>28</td>
<td>GENERAL QUANTITY ADVERB PHRASE(DPB)</td>
</tr>
<tr>
<td>29</td>
<td>QUALITATIVE ADVERB PHRASE(APC)</td>
</tr>
<tr>
<td>30</td>
<td>ADVERB PHRASE(DP)</td>
</tr>
<tr>
<td>31</td>
<td>BASIC PARTICIPLE PHRASE(EA)</td>
</tr>
<tr>
<td>32</td>
<td>VERB MOOD PHRASE(VB)</td>
</tr>
<tr>
<td>33</td>
<td>THREE-TENSE VERB PHRASE(VBB)</td>
</tr>
<tr>
<td>34</td>
<td>EMPHATIC VERB PHRASE(VC)</td>
</tr>
<tr>
<td>35</td>
<td>SEVEN-TENSE VERB PHRASE(VAA)</td>
</tr>
<tr>
<td>36</td>
<td>VERB PHRASE(VA)</td>
</tr>
<tr>
<td>37</td>
<td>DEFINITE NUMBER PHRASE(BC)</td>
</tr>
<tr>
<td>38</td>
<td>PREPOSITION PHRASE(XP)</td>
</tr>
<tr>
<td>39</td>
<td>DIRECT OBJECT PHRASE(NO)</td>
</tr>
<tr>
<td>40</td>
<td>COPULATIVE ADVERB PHRASE(OPD)</td>
</tr>
<tr>
<td>41</td>
<td>SUBJECT PRONOUN PHRASE(RSP)</td>
</tr>
<tr>
<td>42</td>
<td>SUBJECT PHRASE(NSP)</td>
</tr>
<tr>
<td>43</td>
<td>OBJECT PHRASE(NOP)</td>
</tr>
<tr>
<td>44</td>
<td>INDIRECT PHRASE(NIP)</td>
</tr>
<tr>
<td>45</td>
<td>COPULATIVE PHRASE(NPX)</td>
</tr>
<tr>
<td>46</td>
<td>DIRECT-OBJ. VERB MODIF. PHRASE(VMA)</td>
</tr>
<tr>
<td>47</td>
<td>INDIRECT-OBJ. VERB MODIF. PHRASE(VMB)</td>
</tr>
<tr>
<td>48</td>
<td>DISCOURSE VERB MODIF. PHRASE(VMC)</td>
</tr>
</tbody>
</table>
Table 4-4 (Continued)

<table>
<thead>
<tr>
<th>Symbol No.</th>
<th>Symbol Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>DOUBLE-ACCUS.VERB MODIF.PHRASE (VMD)</td>
</tr>
<tr>
<td>50</td>
<td>VERB MODIFYING PHRASE (VM)</td>
</tr>
<tr>
<td>51</td>
<td>OBJECT REL.CLAUSE VERB.MOD.PHRASE (VMR)</td>
</tr>
<tr>
<td>52</td>
<td>INDIR.REL.CLAUSE VERB.MOD.PHRASE (VMI)</td>
</tr>
<tr>
<td>53</td>
<td>PREDICATE PHRASE (VP)</td>
</tr>
<tr>
<td>54</td>
<td>OBJ.REL.CLAUSE VERB PHRSE (VRB)</td>
</tr>
<tr>
<td>55</td>
<td>INDIRECT REL.CLAUSE VERB PHRASE (VRI)</td>
</tr>
<tr>
<td>56</td>
<td>INFINITIVE CONSTRUCT PHRSE (NV)</td>
</tr>
<tr>
<td>57</td>
<td>INFINITIVE ABSOLUTE PHRASE (NW)</td>
</tr>
<tr>
<td>58</td>
<td>CONSTRUCT PARTICIPLE PHRS (EPB)</td>
</tr>
<tr>
<td>59</td>
<td>ABSOLUTE/CONSTRUCT PART.PHRASE (EPA)</td>
</tr>
<tr>
<td>60</td>
<td>PARTICIPLE PHRASE (EP)</td>
</tr>
<tr>
<td>61</td>
<td>POSSESSIVE INDEPENDENT CLAUSE (SAA)</td>
</tr>
<tr>
<td>62</td>
<td>DEFINITE INDEPENDENT CLSE (SAB)</td>
</tr>
<tr>
<td>63</td>
<td>INDEFINITE INDEPENDENT CLAUSE (SAC)</td>
</tr>
<tr>
<td>64</td>
<td>INDEPENDENT CLAUSE (SA)</td>
</tr>
<tr>
<td>65</td>
<td>OBJECTIVE RELATIVE PHRASE (SRO)</td>
</tr>
<tr>
<td>66</td>
<td>INDIRECT RELATIVE PHRASE (SRI)</td>
</tr>
<tr>
<td>67</td>
<td>RELATIVE PRONOUN CLAUSE (RG)</td>
</tr>
<tr>
<td>68</td>
<td>SUBJECT-OBJ.DEPENDENT CLAUSE (KN)</td>
</tr>
<tr>
<td>69</td>
<td>CIRCUMSTANTIAL DEPENDENT CLAUSE (KC)</td>
</tr>
<tr>
<td>70</td>
<td>CONDITIONAL CLUSE (KK)</td>
</tr>
<tr>
<td>71</td>
<td>INTERROGATIVE CLAUSE (KI)</td>
</tr>
<tr>
<td>72</td>
<td>DISCOURSE CLAUSE (KD)</td>
</tr>
<tr>
<td>73</td>
<td>DEPENDENT CLAUSE SENTENCE (SD)</td>
</tr>
<tr>
<td>74</td>
<td>BASIC SENTENCE (S)</td>
</tr>
<tr>
<td>75</td>
<td>INTERROGATIVE SENTENCE (SI)</td>
</tr>
<tr>
<td>76</td>
<td>COMPLETED SENTENCE (SC)</td>
</tr>
<tr>
<td>77</td>
<td>ADJECTIVE (A)</td>
</tr>
<tr>
<td>78</td>
<td>NUMBER ABSOLUTE (B)</td>
</tr>
<tr>
<td>79</td>
<td>CONJUNCTION (C)</td>
</tr>
<tr>
<td>80</td>
<td>ADVERB (D)</td>
</tr>
<tr>
<td>81</td>
<td>PARTICIPLE ABSOLUTE (E)</td>
</tr>
<tr>
<td>82</td>
<td>PARTICIPLE CONSTRUCT (G)</td>
</tr>
<tr>
<td>83</td>
<td>DEFINITE ARTICLE (H)</td>
</tr>
<tr>
<td>84</td>
<td>NUMBER CONSTRUCT (I)</td>
</tr>
<tr>
<td>85</td>
<td>NOUN CONSTRUCT (J)</td>
</tr>
<tr>
<td>86</td>
<td>NEGATIVE (L)</td>
</tr>
<tr>
<td>87</td>
<td>NOUN ABSOLUTE (K)</td>
</tr>
<tr>
<td>88</td>
<td>SIGN OF DIRECT OBJECT (O)</td>
</tr>
<tr>
<td>89</td>
<td>PREPOSITION (P)</td>
</tr>
<tr>
<td>90</td>
<td>INTEROGATIVE (Q)</td>
</tr>
<tr>
<td>91</td>
<td>PRONOUN (R)</td>
</tr>
<tr>
<td>92</td>
<td>PUNCTUATION MRK (T)</td>
</tr>
<tr>
<td>93</td>
<td>PARTICLE (U)</td>
</tr>
<tr>
<td>94</td>
<td>VERB (V)</td>
</tr>
<tr>
<td>95</td>
<td>INFINITIVE ABSOLUTE (W)</td>
</tr>
<tr>
<td>96</td>
<td>INFINITIVE CONSTRUCT (Y)</td>
</tr>
</tbody>
</table>
Table 4-5
LIST OF ANALYSIS PREDICATES

<table>
<thead>
<tr>
<th>Statement No.</th>
<th>Analysis Predicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CONTAINS A PREPOSITION AND A PRON.</td>
</tr>
<tr>
<td>2</td>
<td>IS THE PREDICATE OF OBJ. INTERROG. PHRASE.</td>
</tr>
<tr>
<td>3</td>
<td>EXPRESSES THE NONCOMPARATIVE DEGREE.</td>
</tr>
<tr>
<td>4</td>
<td>EXPRESSES THE COMPARATIVE DEGREE.</td>
</tr>
<tr>
<td>5</td>
<td>EXPRESSES THE SUPERLATIVE DEGREE.</td>
</tr>
<tr>
<td>6</td>
<td>IS A BASIC POST-NOMINAL ADJ. PHRASE.</td>
</tr>
<tr>
<td>7</td>
<td>EXPRESSES POSSESSION BY A PRONOUN.</td>
</tr>
<tr>
<td>8</td>
<td>EXPRESSES POSSESSION BY A NOUN PHRASE.</td>
</tr>
<tr>
<td>9</td>
<td>HAS A NONDETERMINATE NOUN.</td>
</tr>
<tr>
<td>10</td>
<td>HAS A DETERMINATE NOUN.</td>
</tr>
<tr>
<td>11</td>
<td>HAS A (DETERMINATE) PROPER NOUN.</td>
</tr>
<tr>
<td>12</td>
<td>NAMES THE SUBJECT.</td>
</tr>
<tr>
<td>13</td>
<td>DOES NOT NAME THE SUBJECT.</td>
</tr>
<tr>
<td>14</td>
<td>EXPRESSES POSSESSION BY A CONSTRUCT.</td>
</tr>
<tr>
<td>15</td>
<td>STANDS IN PLACE OF A NOUN.</td>
</tr>
<tr>
<td>16</td>
<td>MODIFIES A DETERMINATE NOUN.</td>
</tr>
<tr>
<td>17</td>
<td>CONSISTS OF A PERSONAL PRONOUN (DET).</td>
</tr>
<tr>
<td>18</td>
<td>CONSISTS OF AN OBJECT PRONOUN.</td>
</tr>
<tr>
<td>19</td>
<td>IS THE NUMBER ONE.</td>
</tr>
<tr>
<td>20</td>
<td>IS THE NUMBER TWO.</td>
</tr>
<tr>
<td>21</td>
<td>IS A NUMBER BETWEEN 3 AND 9.</td>
</tr>
<tr>
<td>22</td>
<td>IS THE NUMBER 10.</td>
</tr>
<tr>
<td>23</td>
<td>IS THE NUMBER 11.</td>
</tr>
<tr>
<td>24</td>
<td>IS THE NUMBER 12.</td>
</tr>
<tr>
<td>25</td>
<td>IS A NUMBER BETWEEN 13 AND 19.</td>
</tr>
<tr>
<td>26</td>
<td>IS THE NUMBER 20.</td>
</tr>
<tr>
<td>27</td>
<td>IS EITHER 30, 40, 50, 60, 70, 80, OR 90.</td>
</tr>
<tr>
<td>28</td>
<td>IS THE NUMBER 100.</td>
</tr>
<tr>
<td>29</td>
<td>IS THE NUMBER 200.</td>
</tr>
<tr>
<td>30</td>
<td>IS EITHER 300, 400, ..., 800 OR 900.</td>
</tr>
<tr>
<td>31</td>
<td>IS THE NUMBER 1000.</td>
</tr>
<tr>
<td>32</td>
<td>IS THE NUMBER 2000.</td>
</tr>
<tr>
<td>33</td>
<td>IS EITHER 3000, 4000, ..., OR 9000.</td>
</tr>
<tr>
<td>34</td>
<td>IS A UNITS NUMBER PHRASE.</td>
</tr>
<tr>
<td>35</td>
<td>IS A TENS NUMBER PHRASE.</td>
</tr>
<tr>
<td>36</td>
<td>IS A TEENS NUMBER PHRASE.</td>
</tr>
<tr>
<td>37</td>
<td>IS A MULTI-TENS NUMBER PHRASE.</td>
</tr>
<tr>
<td>38</td>
<td>CONSISTS OF A MULTI-TEN AND UNITS.</td>
</tr>
<tr>
<td>39</td>
<td>IS THE NUMBER 100 OR MULTIPLE OF IT.</td>
</tr>
<tr>
<td>40</td>
<td>IS A NUMBER BETWEEN 100 AND 999.</td>
</tr>
<tr>
<td>41</td>
<td>IS A NUMBER BETWEEN 1 AND 99.</td>
</tr>
<tr>
<td>42</td>
<td>IS A NUMBER BETWEEN 100 AND 999.</td>
</tr>
<tr>
<td>43</td>
<td>IS THE NUMBER 1000 OR ITS MULTIPLE.</td>
</tr>
<tr>
<td>Statement No.</td>
<td>Analysis Predicate</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>44</td>
<td>IS A NUMBER BETWEEN 1000 AND 9999.</td>
</tr>
<tr>
<td>45</td>
<td>IS A NUMBER BETWEEN 1 AND 999.</td>
</tr>
<tr>
<td>46</td>
<td>HAS A BASIC NOUN PHRASE AS NUCLEUS.</td>
</tr>
<tr>
<td>47</td>
<td>HAS A POSSESS. NOUN PHRASE AS NUCLEUS.</td>
</tr>
<tr>
<td>48</td>
<td>HAS A PARTICIPLE PHRASE AS NUCLEUS.</td>
</tr>
<tr>
<td>49</td>
<td>CONTAINS NO CONSTRUCT NOUNS.</td>
</tr>
<tr>
<td>50</td>
<td>CONTAINS CONSTRUCT NOUNS.</td>
</tr>
<tr>
<td>51</td>
<td>IS A DEFINITE NUMBER PHRASE.</td>
</tr>
<tr>
<td>52</td>
<td>IS A REGULAR NOUN PHRASE.</td>
</tr>
<tr>
<td>53</td>
<td>IS AN APPPOSITIONAL NOUN PHRASE.</td>
</tr>
<tr>
<td>54</td>
<td>IS A PREPOSITIONAL PHRASE.</td>
</tr>
<tr>
<td>55</td>
<td>IS A RELATIVE CLAUSE.</td>
</tr>
<tr>
<td>56</td>
<td>IS A REG. NOUN PHRASE.</td>
</tr>
<tr>
<td>57</td>
<td>IS A SPEC. QUANT. ADVERB.</td>
</tr>
<tr>
<td>58</td>
<td>IS A NUMBER + A SPEC. QUANT. ADVERB.</td>
</tr>
<tr>
<td>59</td>
<td>IS A CLASS 4 ADVERB + (MODIFIER).</td>
</tr>
<tr>
<td>60</td>
<td>IS A CLASS 5 ADVERB + (MODIFIER).</td>
</tr>
<tr>
<td>61</td>
<td>IS A TEMPORAL ADVERB PHRASE.</td>
</tr>
<tr>
<td>62</td>
<td>IS A LOCATIVE ADVERB PHRASE.</td>
</tr>
<tr>
<td>63</td>
<td>IS A SPECIFIC QUANTITY ADVERB.</td>
</tr>
<tr>
<td>64</td>
<td>IS A GENERAL QUANTITY ADVERB.</td>
</tr>
<tr>
<td>65</td>
<td>IS A QUALITATIVE ADVERB PHRASE.</td>
</tr>
<tr>
<td>66</td>
<td>IS AN INTENSITY ADVERB PHRASE.</td>
</tr>
<tr>
<td>67</td>
<td>IS A PREPOSITIONAL PHRASE.</td>
</tr>
<tr>
<td>68</td>
<td>IS AN UNDERTERMINATED PARTICIPLE.</td>
</tr>
<tr>
<td>69</td>
<td>IS A DETERMINATED PARTICIPLE.</td>
</tr>
<tr>
<td>70</td>
<td>IS A VERB OR INFINITIVE ABSOLUTE.</td>
</tr>
<tr>
<td>71</td>
<td>IS A VERB OR PARTICIPLE.</td>
</tr>
<tr>
<td>72</td>
<td>OMMITS VERB FOR PRES. ACT. INDIC. COPUL.</td>
</tr>
<tr>
<td>73</td>
<td>EXPRESSES EMPHASIS OF CERTAINTY (BH).</td>
</tr>
<tr>
<td>74</td>
<td>EXPRESSES EMPHASIS OF DURATION (BH).</td>
</tr>
<tr>
<td>75</td>
<td>EXPRESSES NO SPECIAL EMPHASIS.</td>
</tr>
<tr>
<td>76</td>
<td>IS AN EMPHATIC VERB PHRASE.</td>
</tr>
<tr>
<td>77</td>
<td>IS A SEVEN-TENSE VERB PHRASE.</td>
</tr>
<tr>
<td>78</td>
<td>IS A NONDETERMINATE NUMBER PHRASE.</td>
</tr>
<tr>
<td>79</td>
<td>IS A DETERMINATE NUMBER PHRASE.</td>
</tr>
<tr>
<td>80</td>
<td>GOVERNS A NOUN PHRASE.</td>
</tr>
<tr>
<td>81</td>
<td>IS A PREPOSITIONAL PRONOUN PHRASE.</td>
</tr>
<tr>
<td>82</td>
<td>GOVERNS A RELATIVE PRONOUN CLAUSE.</td>
</tr>
<tr>
<td>83</td>
<td>IS A NOUN PHRASE.</td>
</tr>
<tr>
<td>84</td>
<td>IS A RELATIVE PRONOUN CLAUSE.</td>
</tr>
<tr>
<td>85</td>
<td>IS A RELATIVE PRONOUN CLAUSE.</td>
</tr>
<tr>
<td>Statement No.</td>
<td>Analysis Predicate</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>86</td>
<td>IS A TEMPORAL ADVERB.</td>
</tr>
<tr>
<td>87</td>
<td>IS A LOCATIVE ADVERB.</td>
</tr>
<tr>
<td>88</td>
<td>IS A SUBJECT PRONOUN + (APPOS. N. PH.)</td>
</tr>
<tr>
<td>89</td>
<td>IS A DEMONSTRATIVE PRONOUN.</td>
</tr>
<tr>
<td>90</td>
<td>IS A NOUN PHRASE.</td>
</tr>
<tr>
<td>91</td>
<td>IS A SUBJECT PRONOUN PHRASE.</td>
</tr>
<tr>
<td>92</td>
<td>IS A SUBJECT-OBJECT DEPENDENT CLAUSE</td>
</tr>
<tr>
<td>93</td>
<td>IS AN INFINITIVE CONSTRUCT PHRASE.</td>
</tr>
<tr>
<td>94</td>
<td>IS A NONTDETERMINATE NOUN PHRASE.</td>
</tr>
<tr>
<td>95</td>
<td>IS A DETERMINATE DIRECT OBJECT PHRASE.</td>
</tr>
<tr>
<td>96</td>
<td>IS A PREPOSITIONAL PHRASE.</td>
</tr>
<tr>
<td>97</td>
<td>IS AN INFINITIVE PHRASE.</td>
</tr>
<tr>
<td>98</td>
<td>IS AN ADJECTIVE PHRASE.</td>
</tr>
<tr>
<td>99</td>
<td>IS A COPULATIVE ADVERB PHRASE.</td>
</tr>
<tr>
<td>100</td>
<td>IS A NOUN PHRASE.</td>
</tr>
<tr>
<td>101</td>
<td>IS A SUBJECT PRONOUN PHRASE.</td>
</tr>
<tr>
<td>102</td>
<td>IS A PREPOSITIONAL PHRASE.</td>
</tr>
<tr>
<td>103</td>
<td>IS A CIRCUMSTANTIAL DEPENDENT CLAUSE.</td>
</tr>
<tr>
<td>104</td>
<td>IS A PRONOUN + (ADVERB PHRASE).</td>
</tr>
<tr>
<td>105</td>
<td>IS A DIRECT OBJ. PRON. + (ADV. PHRASE).</td>
</tr>
<tr>
<td>106</td>
<td>IS A DIRECT OBJECT PHRASE.</td>
</tr>
<tr>
<td>107</td>
<td>HAS AN OBJ. PRON. AND AN IND. PHRASE.</td>
</tr>
<tr>
<td>108</td>
<td>HAS A DIR. OBJ. PRON. AND AN IND. PHRASE.</td>
</tr>
<tr>
<td>109</td>
<td>HAS AN OBJ. PHRASE. AND AN IND. PHRASE.</td>
</tr>
<tr>
<td>110</td>
<td>HAS AN OBJ. PRON. AND A DISC. CLAUSE.</td>
</tr>
<tr>
<td>111</td>
<td>HAS A DIR. OBJ. PRON. AND A DISC. CLAUSE.</td>
</tr>
<tr>
<td>112</td>
<td>HAS A DISC. CLAUSE + (OTHER MODIFS).</td>
</tr>
<tr>
<td>113</td>
<td>HAS AN OBJ. PRON. AND A NOUN PHRASE.</td>
</tr>
<tr>
<td>114</td>
<td>HAS A DIR. OBJ. PRON. AND A NOUN PHRASE.</td>
</tr>
<tr>
<td>115</td>
<td>HAS AN OBJ. PHRASE AND A NOUN PHRASE.</td>
</tr>
<tr>
<td>116</td>
<td>IS A COPULATIVE PHRASE.</td>
</tr>
<tr>
<td>117</td>
<td>IS AN ADVERB PHRASE.</td>
</tr>
<tr>
<td>118</td>
<td>IS A DIRECT OBJECT VERB MOD. PHRASE.</td>
</tr>
<tr>
<td>119</td>
<td>IS AN INDIRECT OBJECT VERB MOD. PHRASE.</td>
</tr>
<tr>
<td>120</td>
<td>IS A PREPOSITIONAL PHRASE.</td>
</tr>
<tr>
<td>121</td>
<td>IS AN INFINITIVE PHRASE.</td>
</tr>
<tr>
<td>122</td>
<td>IS A DISCOURSE VERB MODIF. PHRASE.</td>
</tr>
<tr>
<td>123</td>
<td>IS A DOUBLE ACCUS. VERB MODIF. PHRASE.</td>
</tr>
<tr>
<td>124</td>
<td>IS AN ADVERB PHRASE.</td>
</tr>
<tr>
<td>125</td>
<td>IS A DIR. OBJ. VERB MODIF. PHRASE.</td>
</tr>
<tr>
<td>126</td>
<td>IS AN INDIRECT OBJ. VERB MODIF. PHRASE.</td>
</tr>
<tr>
<td>127</td>
<td>IS A PREPOSITION-PRON. + (ADV. MODIF.).</td>
</tr>
<tr>
<td>Statement No.</td>
<td>Analysis Predicate</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>128</td>
<td>IS A DOUBLE-ACC. VERB MODIF. PHRASE.</td>
</tr>
<tr>
<td>129</td>
<td>IS AN OBJ. PRN. AND A PREP. PRON. PHRASE.</td>
</tr>
<tr>
<td>130</td>
<td>IS A DIR-OBJ. PRON. + PREP. PRON. PHRASE.</td>
</tr>
<tr>
<td>131</td>
<td>IS A VERB PHRASE + VRB. MODIF. PHRASE.</td>
</tr>
<tr>
<td>132</td>
<td>IS A NP. PHRASE + IND. REL. CLS. VB. M. PHRASE.</td>
</tr>
<tr>
<td>134</td>
<td>HAS PREVIOUSLY NAMED SUBJECT.</td>
</tr>
<tr>
<td>135</td>
<td>NAMES SUBJECT BY MEANS OF PRON.</td>
</tr>
<tr>
<td>136</td>
<td>HAS AN INFIN. ABS. AND VRB. MOD. PHRASE.</td>
</tr>
<tr>
<td>138</td>
<td>HAS A NOUN PHRASE AS OBJECT.</td>
</tr>
<tr>
<td>144</td>
<td>EMPHASIZES THE THING POSSESSED.</td>
</tr>
<tr>
<td>145</td>
<td>HAS NO SPECIAL EMPHASIS.</td>
</tr>
<tr>
<td>146</td>
<td>HAS SPECIAL SYNTAX FOR PRES. ACT. IND.</td>
</tr>
<tr>
<td>147</td>
<td>HAS A NAMED SUBJECT.</td>
</tr>
<tr>
<td>148</td>
<td>NAMES SUBJECT + SPECIAL PRES. ACT. IND.</td>
</tr>
<tr>
<td>150</td>
<td>NAMES SUBJECT + SPECIAL NEG. PR. ACT. IND.</td>
</tr>
<tr>
<td>152</td>
<td>EMPHASIZES VERBAL IDEA, SPEC. PR. A. I.</td>
</tr>
<tr>
<td>154</td>
<td>REFERS TO SUBJECT PREVIOUSLY NAMED.</td>
</tr>
<tr>
<td>156</td>
<td>NAMES THE SUBJECT OF THE VERB.</td>
</tr>
<tr>
<td>158</td>
<td>HAS REL. PRON. AS SUBJECT OF VERB.</td>
</tr>
<tr>
<td>160</td>
<td>HAS REL. PRON. AS OBJECT OF VERB.</td>
</tr>
<tr>
<td>161</td>
<td>HAS REL. PRON. RELATED TO VERB BY PRP.</td>
</tr>
<tr>
<td>162</td>
<td>USES MODERN HEBREW CONJUNCTION(S).</td>
</tr>
<tr>
<td>163</td>
<td>IS A TIME DEPENDENT CLAUSE.</td>
</tr>
<tr>
<td>164</td>
<td>IS A PURPOSE-RESULT DEPENDENT CLAUSE.</td>
</tr>
<tr>
<td>165</td>
<td>IS A CAUSE-REASON DEPENDENT CLAUSE.</td>
</tr>
<tr>
<td>166</td>
<td>IS A CIRCUMSTANTIAL PREP. PHRASE.</td>
</tr>
<tr>
<td>167</td>
<td>IS THE PROTASIS OF THE COND. SENTENCE.</td>
</tr>
<tr>
<td>168</td>
<td>QUESTIONS TRUTH/CIRCUMSTANCES OF SN.</td>
</tr>
<tr>
<td>169</td>
<td>ASKS WHO/WHAT IS SUBJECT OF VERB.</td>
</tr>
<tr>
<td>Statement No.</td>
<td>Analysis Predicate</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>171</td>
<td>ASKS WHO/WHAT IS OBJECT OF VERB.</td>
</tr>
<tr>
<td>172</td>
<td>ASKS WHO/WHAT IS IND.OBJ.OF VERB.</td>
</tr>
<tr>
<td>173</td>
<td>QUOTES THE DISCOURSE INDIRECTLY.</td>
</tr>
<tr>
<td>174</td>
<td>QUOTES THE DISCOURSE DIRECTLY.</td>
</tr>
<tr>
<td>175</td>
<td>HAS NO EMPHASIS ON DEPENDENT CLAUSE.</td>
</tr>
<tr>
<td>176</td>
<td>HAS SOME EMPHASIS ON DEPENDENT CLAUSE.</td>
</tr>
<tr>
<td>177</td>
<td>IS A SIMPLE SENTENCE.</td>
</tr>
<tr>
<td>178</td>
<td>IS A DEPENDENT CLAUSE SENTENCE.</td>
</tr>
<tr>
<td>179</td>
<td>IS A CONDITIONAL SENTENCE.</td>
</tr>
<tr>
<td>180</td>
<td>HAS NO DEPENDENT CLAUSE.</td>
</tr>
<tr>
<td>181</td>
<td>HAS A DEPENDENT CLAUSE (NO EMPHASIS).</td>
</tr>
<tr>
<td>182</td>
<td>HAS A DEPENDENT CLAUSE (EMPHASIS).</td>
</tr>
<tr>
<td>183</td>
<td>IS A DECLARATIVE SENTENCE.</td>
</tr>
<tr>
<td>184</td>
<td>IS AN INTERROGATIVE SENTENCE.</td>
</tr>
<tr>
<td>185</td>
<td>IS AN IMPERATIVE SENTENCE.</td>
</tr>
</tbody>
</table>
Table 4-6
INDEX OF ANALYSIS STATEMENTS

<table>
<thead>
<tr>
<th>Symbol No.</th>
<th>Symbol Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>13</td>
<td>23</td>
</tr>
<tr>
<td>14</td>
<td>26</td>
</tr>
<tr>
<td>15</td>
<td>28</td>
</tr>
<tr>
<td>16</td>
<td>31</td>
</tr>
<tr>
<td>17</td>
<td>34</td>
</tr>
<tr>
<td>18</td>
<td>39</td>
</tr>
<tr>
<td>19</td>
<td>41</td>
</tr>
<tr>
<td>20</td>
<td>43</td>
</tr>
<tr>
<td>21</td>
<td>45</td>
</tr>
<tr>
<td>22</td>
<td>47</td>
</tr>
<tr>
<td>23</td>
<td>50</td>
</tr>
<tr>
<td>24</td>
<td>53</td>
</tr>
<tr>
<td>25</td>
<td>54</td>
</tr>
<tr>
<td>26</td>
<td>57</td>
</tr>
<tr>
<td>27</td>
<td>58</td>
</tr>
<tr>
<td>28</td>
<td>60</td>
</tr>
<tr>
<td>29</td>
<td>61</td>
</tr>
<tr>
<td>30</td>
<td>62</td>
</tr>
<tr>
<td>31</td>
<td>69</td>
</tr>
<tr>
<td>32</td>
<td>71</td>
</tr>
<tr>
<td>33</td>
<td>72</td>
</tr>
<tr>
<td>34</td>
<td>74</td>
</tr>
<tr>
<td>35</td>
<td>77</td>
</tr>
<tr>
<td>36</td>
<td>78</td>
</tr>
<tr>
<td>37</td>
<td>79</td>
</tr>
<tr>
<td>Symbol No.</td>
<td>1</td>
</tr>
<tr>
<td>-----------</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>81</td>
</tr>
<tr>
<td>2</td>
<td>84</td>
</tr>
<tr>
<td>3</td>
<td>86</td>
</tr>
<tr>
<td>4</td>
<td>88</td>
</tr>
<tr>
<td>5</td>
<td>90</td>
</tr>
<tr>
<td>6</td>
<td>94</td>
</tr>
<tr>
<td>7</td>
<td>96</td>
</tr>
<tr>
<td>8</td>
<td>98</td>
</tr>
<tr>
<td>9</td>
<td>104</td>
</tr>
<tr>
<td>10</td>
<td>107</td>
</tr>
<tr>
<td>11</td>
<td>110</td>
</tr>
<tr>
<td>12</td>
<td>113</td>
</tr>
<tr>
<td>13</td>
<td>116</td>
</tr>
<tr>
<td>14</td>
<td>124</td>
</tr>
<tr>
<td>15</td>
<td>129</td>
</tr>
<tr>
<td>16</td>
<td>132</td>
</tr>
<tr>
<td>17</td>
<td>132</td>
</tr>
<tr>
<td>18</td>
<td>133</td>
</tr>
<tr>
<td>19</td>
<td>134</td>
</tr>
<tr>
<td>20</td>
<td>137</td>
</tr>
<tr>
<td>21</td>
<td>138</td>
</tr>
<tr>
<td>22</td>
<td>140</td>
</tr>
<tr>
<td>23</td>
<td>142</td>
</tr>
<tr>
<td>24</td>
<td>143</td>
</tr>
<tr>
<td>25</td>
<td>147</td>
</tr>
<tr>
<td>26</td>
<td>153</td>
</tr>
<tr>
<td>27</td>
<td>154</td>
</tr>
<tr>
<td>28</td>
<td>157</td>
</tr>
<tr>
<td>29</td>
<td>157</td>
</tr>
<tr>
<td>30</td>
<td>159</td>
</tr>
<tr>
<td>31</td>
<td>162</td>
</tr>
<tr>
<td>32</td>
<td>164</td>
</tr>
<tr>
<td>33</td>
<td>168</td>
</tr>
<tr>
<td>34</td>
<td>169</td>
</tr>
<tr>
<td>35</td>
<td>173</td>
</tr>
<tr>
<td>36</td>
<td>175</td>
</tr>
<tr>
<td>37</td>
<td>177</td>
</tr>
<tr>
<td>38</td>
<td>180</td>
</tr>
<tr>
<td>39</td>
<td>183</td>
</tr>
</tbody>
</table>
### Table 4-7

**LIST OF FEATURE VALUES**

<table>
<thead>
<tr>
<th>Feature Subscript</th>
<th>Feature Name</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>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>compounding</td>
<td>f=1</td>
<td>f=2</td>
<td>f=3</td>
<td>f=4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>k</td>
<td>com.</td>
<td>once</td>
<td>twice</td>
<td>three</td>
<td>four</td>
<td>five</td>
<td>six</td>
<td>seven</td>
<td>eight</td>
</tr>
<tr>
<td>b</td>
<td>compounding class</td>
<td>conj.</td>
<td>disj.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>c</td>
<td>Symbol class</td>
<td>one</td>
<td>two</td>
<td>three</td>
<td>four</td>
<td>five</td>
<td>six</td>
<td>seven</td>
<td>eight</td>
</tr>
<tr>
<td>e</td>
<td>negation type</td>
<td>one</td>
<td>two</td>
<td>three</td>
<td>four</td>
<td>five</td>
<td>six</td>
<td>seven</td>
<td>eight</td>
</tr>
<tr>
<td>y</td>
<td>negation</td>
<td>neg.</td>
<td>neg.</td>
<td>neg.</td>
<td>neg.</td>
<td>neg.</td>
<td>neg.</td>
<td>neg.</td>
<td>neg.</td>
</tr>
<tr>
<td>d</td>
<td>determination</td>
<td>indef.</td>
<td>def.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>n</td>
<td>number</td>
<td>sing.</td>
<td>dual.</td>
<td>pl.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>g</td>
<td>gender</td>
<td>masc.</td>
<td>fem.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>p</td>
<td>pers.</td>
<td>first.</td>
<td>sec.</td>
<td>third</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>r</td>
<td>preposition class</td>
<td>r=1</td>
<td>r=2</td>
<td>r=3</td>
<td>r=4</td>
<td>r=5</td>
<td>r=6</td>
<td>r=7</td>
<td>r=8</td>
</tr>
<tr>
<td>a</td>
<td>verb class</td>
<td>a=1</td>
<td>a=2</td>
<td>a=3</td>
<td>a=4</td>
<td>a=5</td>
<td>a=6</td>
<td>a=7</td>
<td>a=8</td>
</tr>
<tr>
<td>v</td>
<td>voice</td>
<td>act.</td>
<td>pass.</td>
<td>refl.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>i</td>
<td>mood</td>
<td>ind.</td>
<td>impv.</td>
<td>subj.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>t</td>
<td>tense</td>
<td>past</td>
<td>futr.</td>
<td>pres.</td>
<td>pst. C. fut. C. plpf.</td>
<td>f. prf.</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 4-8 is a list of the variables used by the algorithm together with a description of each.

4.2 The Procedure

This section describes the procedure for using the structural model to analyze Hebrew sentences. It consists of the logical interaction of a set of operational functions that manipulate the data of the mapping functions in accordance with the specified values of the independent variables supplied by the user. Basically the procedure begins with a string of initial symbols of the Grammar of Hebrew Syntax and applies the replacement rules of the grammar until only one terminal symbol remain. The major operational functions required to perform this task are listed below:

(1) Read in the initial symbols
(2) Write the initial symbols on the "A" list of symbols
(3) Initialize symbol counting indices
(4) Compare computed maximum value with 1
(5) Increment symbol counting index
(6) Compare symbol counting index with the computed maximum value
(7) Read a symbol from the "A" list of symbols
(8) Write a symbol on the "B" list from "A" list
(9) Locate a grammar rule that applies to a given symbol
(10) Determine that a rule exists
(11) Determine that the rule matches the given symbol with respect to all subscripts
(12) Initialize rule element counting indices
(13) Increment rule element counting indices
(14) Compare rule element counting index with the computed maximum value
(15) Determine that an element of the rule matches an element on the "A" list with respect to all subscripts
(16) Compute values for dependent-variable subscripts

---

4See Section 2.2.1 of Part II for a detailed description of the variables as they apply to Hebrew.

5Defined in Part II of this report.
Table 4-4

LIST OF VARIABLES

<table>
<thead>
<tr>
<th>Var. No.</th>
<th>Var. Name</th>
<th>Description</th>
<th>Var. No.</th>
<th>Var. Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SN</td>
<td>Symbol No.</td>
<td>16</td>
<td>i</td>
<td>mood</td>
</tr>
<tr>
<td>2</td>
<td>m</td>
<td>opt./mand.</td>
<td>17</td>
<td>t</td>
<td>tense</td>
</tr>
<tr>
<td>3</td>
<td>f</td>
<td>comp. class</td>
<td>18</td>
<td>s</td>
<td>stem</td>
</tr>
<tr>
<td>4</td>
<td>k</td>
<td>comp. no.</td>
<td>19</td>
<td>w₁</td>
<td>Root 1</td>
</tr>
<tr>
<td>5</td>
<td>b</td>
<td>comp. type</td>
<td>20</td>
<td>w₂</td>
<td>Root 2</td>
</tr>
<tr>
<td>6</td>
<td>c</td>
<td>class</td>
<td>21</td>
<td>w₃</td>
<td>Root 3</td>
</tr>
<tr>
<td>7</td>
<td>z</td>
<td>neg. class</td>
<td>22</td>
<td>w₄</td>
<td>Root 4</td>
</tr>
<tr>
<td>8</td>
<td>y</td>
<td>neg./pos</td>
<td>23</td>
<td>ST</td>
<td>Symbol Type</td>
</tr>
<tr>
<td>9</td>
<td>d</td>
<td>definiteness</td>
<td>24</td>
<td>RN</td>
<td>Rule No.</td>
</tr>
<tr>
<td>10</td>
<td>n</td>
<td>number</td>
<td>25</td>
<td>EN</td>
<td>Element No.</td>
</tr>
<tr>
<td>11</td>
<td>g</td>
<td>gender</td>
<td>26</td>
<td>NRH</td>
<td>No. Rt. Elements</td>
</tr>
<tr>
<td>12</td>
<td>p</td>
<td>person</td>
<td>27</td>
<td>RT</td>
<td>Restraint type</td>
</tr>
<tr>
<td>13</td>
<td>r</td>
<td>prep. class</td>
<td>28</td>
<td>RS</td>
<td>Restraint Subs.</td>
</tr>
<tr>
<td>14</td>
<td>a</td>
<td>verb. class</td>
<td>29</td>
<td>x</td>
<td>No./gend. Trans.</td>
</tr>
<tr>
<td>15</td>
<td>v</td>
<td>voice</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(17) Predict success of a rule after one or two passes.  
(18) Write a rule element on the "B" list  
(19) Reset value of symbol counting index  
(20) Erase the "A" list of symbols and transfer the symbols on the "B" list to the "A" list  
(21) Write sequence of analysis statements  
(22) Construct a tree diagram of the generated sentence.

Many minor operational functions that are associated with these are not listed. They are defined in Section 4.3 that describes the computer program of the algorithm. The above are sufficient for explaining the procedure of the algorithm.

The procedure consists of a logical manipulation of the operational functions so as to analyze a sentence. The logical interrelationship of the functions is defined in flow chart form in Figures 4.1a and 4.1b. The actual program of the procedure is more complex than this flow chart, but this is sufficient for explaining the procedure.

The initial symbol is read in (Block 1) together with the value of all independent variable subscripts, and it is written on the "A" list of symbols (LIST1) as the first symbol (Block 2). The number of symbols on LIST1 (Jmax) is computed and counting register (J) is initialized (Block 3). The value of Jmax is compared with 1 (Block 4), if Jmax < 1 computation proceeds to Block 21, otherwise to Block 5. In Block 5 the counting register (J) is indexed by one and the value of J is compared with Jmax to determine whether or not all the symbols on list LIST1 have been processed (Block 6); if not, computation proceeds to Block 7, otherwise to Block 20.

In Block 7, the J-th symbol of LIST1 is read, and an applicable grammar rule is located (Block 9). If there is no applicable rule (Block 10) computation goes to Block 8, otherwise to Block 11. In Block 8 the J-th symbol of LIST1 is written on LIST2 and computation returns to Block 5.

In Block 11, a test is made to determine that the first left hand element of the rule matches the J-th symbol of LIST1 in accordance with the following criteria:

(1) The symbol number must be the same for both symbols, or if not, the grammar rule symbol must be a variable symbol; and

---

6 The "Block" number corresponds to operation function number previously listed.

7 Predictive logic is required to synchronize the production of related constituents because the rules of the grammar are unordered (by original definition). Prediction to a depth of two passes enables the algorithm to analyze most simple Hebrew sentences. Greater predictive power is required for more complex sentences.
Figure 4.1a: Flow Chart of Procedure
$K_{max} = B$
$K = 0$
$J_1 = J$

$J_1 = J_1 + 1$
$K = K + 1$

$K > K_{max}$

Compute Value of Dependent Variables

To p.a.

12

13

14

15

16

17

18

19

B: No. of elements in Rule

Write Right Hand Element of Rule on LIST2

J = J_1 - 1

to p.a.

C: Does J_1-th Symbol of LIST1 Match K-th element of Rule?

D: Predict Success in 1 or 2 Future Passes?

Figure 4-1b: Flow Chart of Procedure

C: Does J_1-th Symbol of LIST1 Match K-th element of Rule?

D: Predict Success in 1 or 2 Future Passes?
(2) the symbol class must be the same for both symbols; and

(3) every fixed-valued subscript of the rule symbol must be the same as the corresponding subscript of the J-th symbol; and

(4) the restraints on the rule symbol must be met.

If the criteria are not met (Block 11) computation returns to Block 9 for location of the next applicable grammar rule. If the criteria are met, computation proceeds to Block 12.

In Block 12, the number of symbols on the right side of the rule \(K_{\text{max}}\) is computed and counting register \(K\) is initialized to zero and register \(J_1\) is set equal to \(J\). Then registers \(K\) and \(J_1\) are indexed by one (Block 13) and the value of \(K\) is compared with \(K_{\text{max}}\) (Block 14) to determine whether or not all the right left elements of the rule been processed; if not, computation proceeds to Block 15, otherwise to Block 18. In Block 15, a test is made to determine that the \(J_1\)-th symbol of LIST1 matches the \(K\)-th element of the given rule in accordance with the criteria listed for Block 11. If the symbols match computation proceeds to Block 16, otherwise to Block 17. In Block 16, the values of the variable subscripts of the \(K\)-th element of the rule are computed and computation returns to Block 13.

In Block 17, a test is made to predict whether the present rule would be satisfied after one or two more passes, that is, whether the \(J_1\)-th symbol of LIST1 will develop in one or two passes into the symbol required by the present rule. If success is predicted computation returns to Block 8 and the \(J\)-th symbol is held over so that the predicted development can take place; if success is not predicted, computation returns to Block 9 for a new rule. In Block 18, the right hand element of the rule is written on LIST2 with the computed values assigned to its subscripts, and computation proceeds to Block 19.

In Block 19, the value of the symbol counting index is changed to the value \(J_1-1\), and computation returns to Block 5 to begin with the next symbol on USTI.

Referring back to Block 6, if all symbols on LIST1 have been processed, computation proceeds to Block 20 where the symbols on LIST1 are erased, those on LIST2 are transferred to LIST1, LIST2 is erased, and computation returns to Block 3 for processing the new symbols on LIST1.

Referring back to Block 4, if only one symbol remains on LIST1, the analysis is complete, so a list of analysis statements is assembled (Block 21), a tree diagram of the analyzed sentence is constructed (Block 22), and computation stops.
4.3 Computer Program of the Algorithm

This section describes a computer program of the Algorithm for Analyzing Hebrew Sentences. First a flow-chart description of the program is given, then an input map is provided with instructions on how to use the program. Appendix B of this part of the report contains a source language listing of the program in FORTRAN IV.

4.3.1 Flow-Chart Description of Computer Program

This section describes the computer program of the Algorithm for Analyzing Hebrew Sentences in terms of flow diagrams. The program consists of a main program ANALYZ and the following subprograms:

(1) ALPHA
(2) DIAGRM
(3) LIMIT
(4) MACHER
(5) OUTPUT
(6) PARSE
(7) PROPH1
(8) PROPH2
(9) RERITE
(10) RULENO
(11) SYMACH
(12) VARATT

Figure 4-2 is a map of the program showing the hierarchy of the calling sequences. The following sections contain descriptions of each portion of the program.

4.3.1.1 Main Program ANALYZ

This section describes the operation of the main program of the Algorithm for Analyzing Hebrew Sentences. This program manages the overall operation of the algorithm and calls the various subprograms at the appropriate times.

Figures 4.3 is a flow diagram of main program ANALYZ. The main program performs the following operations:

(1) Reads the program options (Fig. 4.3a)
(2) Reads the Transliteration Table (Fig. 4.3a)
(3) Reads the Matrix of Grammar Rules, transforms the alpha-numeric data to integers using Subprogram ALPHA, and stores the transformed rules in Matrix RULE (Fig. 4.3b).
Figure 4.2: Map of Procedure

1. ANALYZ
2. PARSE
3. DIAGRM
4. RERITE
5. OUTPUT
6. ALPHA
7. PROPH1
8. PROPH2
9. VARATT
10. SYMACH
11. LIMIT
12. MACHE
(1) Read Program Options

Read rules etc. from Mag. Tape if not MODE 0

(2) Read in transliteration

Figure 4.3a: Program ANALYZ
(3) Read in Grammar Rules

(4) Compute Storage address of Rules

(5) Read Restraint Matrix

(6) Read Symbol Names

(7) Write on Mag. Tape

Figure 4.3b: Program ANALYZ
(8) Read in Equivalent Hebrew Sentence

(9) Read in Initial Symbols

(10) Read in Initial Symbols

(11) Initialize Tree Matrix

(12) Initialize Indices

Figure 4.3c: Program ANALYZ
Make Rewrite Pass

Examine Limiting Indices

If Incomplete Pass

Completed Pass, Reset Indices

If Last Pass

Figure 4.3.d: Program ANALYZ
Figure 4.3e: Program ANALYZ

(18) Hunt Back to first Node with Alternate Rule

(19) Write Error Message

(20) Retrieve rule nos.

(18a) Skip if governed Node is same as compound governing Node

A: NODE(4,NODE1) ≠ NODE(4,NG) or NODE(5,NODE1) ≠ NODE(5,NG)

B: ITABLE(NODE1,4) = 0
And
ITABLE(NG,4) > 0

38
Cancel any Previous Prediction

Figure 4.3f: Program ANALYZ
Figure 4.3g: Program ANALYZ
Computes a catalog of the Rules (Fig. 4.3b)

Reads the Restraint Table (Fig. 4.3c)

Reads the table of Symbol Names that are associated with a symbol number (Fig. 4.3c), and other mapping functions

 Writes these data on a magnetic tape for permanent storage (Fig. 4.3c)

Reads the equivalent Hebrew sentence and certain indices (Fig. 4.3c)

Stops the program after the analysis of the last sentence (Fig. 4.3c)

Reads in initial symbols and their English equivalents (Fig. 4.3c)

Initializes tree diagram matrix data

Initializes certain bookkeeping indices

Makes a rewrite pass on the symbols in the string using Subroutine RERITE

Examines the value of monitoring indices which limit the maximum range of computation; if the range is exceeded, error messages are written and computation ceases.

Examines the value of indices MON and I, if no rules were applied on the last pass (MON=0) or if the pass did not process all symbols in the string (I≠IMAXI), computation skips to step (18), otherwise to (16).

If the pass was complete, reset certain indices

If not last pass, that is, there is still more than 1 symbol in the string, repeat from step (13), otherwise skip to step (22).

Hunts back in derivation to first symbol with an alternate rule yet available, but (18a) skips the symbol if it is identical with its governing symbol.

If no symbols with an alternate rule are found, writes a diagnostic message and terminates computation.

When the first symbol with an alternate rule is found, the rule numbers are retrieved and certain indices are reset.

Previous predictions about the symbol are cancelled, other indices are reset and computation returns to step (13).

When analysis is completed, print full analysis (upon request) consisting of complete detailed listing of every symbol in every rewrite pass, using subprogram OUTPUT.

Prints the Hebrew sentence being analyzed.
(24) Prints a tree diagram using subprogram DIAGRAM (upon request).
(25) Prints a complete set of analysis statements using subprogram PARSE, prints a completion message and returns to step (8) for another sentence.

4.3.1.2 Subprogram ALPHA

Subprogram ALPHA (Fig. 4.4) is used to transform alpha-numeric symbols into integers for use in the computations of the program. This subprogram permits input data to be submitted in a form more meaningful to the user. The subprogram functions in the following manner:

1. A given alpha-numeric symbol (input argument A) is compared with each element in the table of Transliteration (array TRANSL).
2. When a match is found, an integer value for output argument IA is computed such that IA is one less than the index L.
3. However, if this value is 40, the value of IA is made zero. This converts all spaces to zero.  

The transformation of the data is shown in Table 4-1. Other transformations can be obtained by changing the data in the Table of Transliteration. Subprogram ALPHA is called by the Main Program ANALYZ.

4.3.1.3 Subprogram DIAGRAM

This subprogram (Figure 4.5) is called by the Main Program ANALYZ (step 24) to construct a tree diagram of an analyzed Hebrew sentence. At this stage the analysis of the sentence is complete and data defining the nodal structure of the tree diagram has been computed and stored in Matrix NODE, and Array ITREE. Matrix NODE contains a row corresponding to each node in the tree diagram. The data in Matrix NODE is as follows:

\[
\text{NODE}(I,J):
\]
- \(J = \text{Node number}\)
- \(I = 1, \text{Number of nodes the } J\text{-th node governs}\)
- \(I = 2, \text{Line position of } J\text{-th node}\)
- \(I = 3, \text{node number of node governing } J\text{-th node}\)

8 This computation is presently dependent on the "content" of the grammar.
Figure 4.4: Subprogram ALPHA
$I = 4$, Symbol type at $J$-th node

$I = 5$, Symbol class at $J$-th node

Array ITREE defines the number of nodes at each level of the tree as follows:

ITREE ($I$):  
I = level number  
ITREE($I$) = number of nodes in $I$-th level.

Figure 4.5a outlines the four main operations of this subprogram and refers to the corresponding figure for the detailed flow diagram of that operation.

These operations are:

1. **Compute the line position of each node.** (Fig. 4.5b&c). This operation consists of filling in the data for Matrix NODE(2,J). Computation starts with the first level of nodes each of which is assigned a line position beginning with position 1 for the first, position 2 for the second, and so forth up to a maximum of 20. Computation then goes to the next level of nodes and the position of each node is computed to be midway between the position of the nodes it governs. Successively lower levels are computed until the last node is reached.

Steps 2 through 4 (following) are repeated in sequence for each level of nodes in the tree diagram. These steps cause the computer to print out the tree diagram of the analyzed sentence a level at a time beginning with the first level. Three parts are required for each level: (a) the upper connectors that show the relationship of the nodes in a given level to the governing nodes at the next highest level; (b) the symbol name and class of each node; and (c) the lower connectors that show the governing relationship of a given node to the nodes at the next lowest level.

2. **Write upper connectors** (Figure 4.5d). The upper connectors occupy two lines on the printed output. The first one connects together (with a horizontal line) nodes governed by a common higher level node. The second line provides a vertical bar above each nodes position. This is accomplished by using various combinations of the following 3-character words:

   BLANK = AAA

   DASHES = ---

   BAR = IAA

4-37
Where Δ represents a space. The logic for the first line is as follows:

(a) If the governing node governs only one node, BLANK + BAR is written above the node position, that is ΔΔΔΔΔ.

(b) If the governing node governs more than one node, BLANK + DASHES (ΔΔΔ---) is written above the first of the governed nodes, DASHES + DASHES (-----) is written above intermediate governed nodes, and DASHES + BLANK (---ΔΔΔ) is written above the last of the governed nodes. This provides a continuous line of dashes from the center of the position of the first node to the center of the position of the last node that are governed by one common higher level node. For the second line of the upper connectors, BLANK + BAR (ΔΔΔΔΔΔ) is written above each node that governs at least 1 node and BLANK + BLANK (AAAAAAA) is written above each node that governs zero nodes.

Upper connectors are omitted for the first level of nodes.

(3) Write line of nodes (Fig. 4.5e). This operation consists of printing the symbol name and class at each nodal position in the line. The symbol type is specified by NS=NODE (4,J), and the corresponding symbol name is obtained from SYWNS). The symbol class is specified by NT=NODE(5,J) which is transformed to alpha-numeric characters by CLS(NT). However, if the symbol and class of a given node are the same as the symbol and class of the governing node, BLANK + BAR (ΔΔΔΔΔΔ) is written in the position of the node; this eliminates redundant data from the tree.

(4) Write lower connectors (Figure 4.5f). This operation consists of writing BLANK + BAR (ΔΔΔΔΔΔ) under the position of a given node if it governs one or more nodes and BLANK + BLANK (AAAAAAA) if it governs zero nodes. This is the same as line 2 of the upper connectors, so the same coding is used for both under control of an index (IUFLW). The lower connectors are omitted for the last level of nodes.
Figure 4.5a: Subprogram DIAGRM
Compute position of nodes

Figure 4.5b: Subprogram DIAGRM
Governing node put midway between first and last governed node.

Continue If Last Node, Otherwise Repeat For Next.

Produce IPASS Lines of Nodes

Omit Upper Connectors For First Line

Figure 4.5c: Subprogram DIAGRM
(2) Write Upper/Lower Connectors (Vert.)

from p.c.

DO 30 L = 1, 20

If
G

If

L1 = L + L - 1
L2 = L + L

L = NODE(2, NO)

ALINE(L1) = BLANK
ALINE(L2) = BAR
NO = NO + 1

end of DO Loop

NO = NO - FREE(M)

PRINT ALINE

IUPLOW = 2

Figure 4.5d: Subprogram DIAGRM
(3) Write Line Of Nodes

from p.d.

DO 40 L = 1,20

If G

L1 = L+L-1
L2 = L+L

ALINE(L1)=SYWNS
ALINE(L2)=CLS(NT)

End of Do Loop 40
to p.f.

see at 21, p.c.

H: [NS=NODE(4,NG)) and
NT=NODE(5,NG)]

31 NO = NO + 1

32 NS = NODE(4,NO)
NT = NODE(5,NO)
NG = NODE(3,NO)
NO = NO + 1

33 If M = 1

34 ALINE(L1)=BLANK
ALINE(L2)=BAR

Figure 4.5e: Subprogram DIAGRM
(4) Write Lower Connectors (vertical)

Omit if last line

Figure 4.5f: Subprogram DIAGRM
ALINE(L1) = BLANK
ALINE(L2) = BLANK
72
L1 = L + L - 1
L2 = L + L
L = IENDPO
T
F

10 -
L - NODE(2, NO) +
12
0

NG = NODE(3, NO)
NO1 = NODE(1, NG)
NO2 = NO + NO1 - 1
IENDPO = NODE(2, NO2)
ALINE(L1) = BLANK

20

11

14

ALINE(L1) = DASHES
ALINE(L2) = BLANK
NO = NO + NO1
IEND = 0
IENDPO = 0

13

End of Do Loop
See Page f

L = IENDPO
T
F

10

14

ALINE(L1) = DASHES
ALINE(L2) = DASHES

Figure 4.5g: Subprogram DIAGRM

4-45
Figure 4.5h: Subprogram DIAGRM

20 from p.g.

NO=NO-ITREE(M)
IUPLOW = 1

PRINT ALINE
to p.d.

51 from p.f.

NO=NO+ITREE(M)

50 end of Do Loop see p.c.

RETURN
4.3.1.4 Subprogram LIMIT

This subprogram (Figure 4.6) is called by Subprogram RERITE and by Subprogram SYMACH to test symbols for certain specified limitations on the values of their subscripts. The limitations are specified by Subscripts 27 and 28. Subscript 27 specifies the restraint type and Subscript 28 specifies the subscript number to which the restraint applies.

The restraint type refers to a row number IR in Restraint Matrix RESTRT(L,IR). The first number in this row, RESTRT(1, IR), tells how many restraint numbers are on the list; the remaining numbers in this row are limitations on the specified subscript. The logic of the limitations is as follows: the value of the specified subscript must be equal to one of the positive numbers on the list, or it must not be equal to the absolute value of the negative numbers on the list. If these restraints are not met the subprogram sets the logic monitor MATCH to "false."

4.3.1.5 Subprogram MACHER

This subprogram (Figure 4.7) is called by Subprogram PROP.H1 and PROPH2 to compare two symbols in accordance with the criteria listed below. Subprograms PROPH1 and PROPH2 make predictions of the possible successful satisfaction of a given rule after one or two more passes on the present string of symbols. In order to do this, the subprograms select likely rules and put their appropriate symbols in registers (IS1 and IS2) which are used by MACHER to determine whether a symbol of a given rule (IS2) could be applied to a given symbol under consideration (IS1) in the process of predictions.

The criteria, which apply to the first 17 subscripts (excluding 3) are as follows:

1. The symbol types must be the same and
2. one of the following must be true for each subscript
   (a) the subscripts are equal, or
   (b) the subscript in IS1 is not negated (test A), or
   (c) the subscript of one or the other of the symbols is an independent variable (test B), or
   (d) the subscript of the symbol in IS1 is a dependent variable and the subscript of the symbol in IS2 is a negative number (test C), or
   (e) the subscript of the symbol in IS1 is a fixed variable and the subscript of the symbol in IS2 is a dependent variable.
Figure 4.6: Subprogram LIMIT
Figure 4.7: Subprogram MACHER

A: \((L=7)\) and \(IS1(8)=0\)

B: \((IS1(L)=9)\) OR \((IS2(1)=9)\)

C: \([((IS2(L)<0)\) and \((IS1(L)<9)]\) OR \([((IS2(L)>9)\) and \((IS1(L)<9)]\)
If the above criteria are met, the logical variable MATCH is set to "true", otherwise "false".

4.3.1.6 Subprogram OUTPUT

This subprogram (Figure 4.8) is called by the Main Program ANALYZ (step 22) to print out the data associated with each symbol in the string after each rewrite pass of the grammar on a string of symbols. See Appendix C for an example of the resultant output from this subprogram.

4.3.1.7 Subprogram PARSE

This subprogram (Figure 4.9) is called by the main program ANALYZ (step 25) in order to print out a list of analysis statements about the sentence being analyzed. At this stage of the computation the analysis is complete and data defining the deep structure of the analysis is stored in matrix NODE, and array ITREE (see Section 4.3.1.3 for more detail). These data include the symbol number, symbol type and the index number of the governing symbol for each constituent of the analysis. For each unique constituent of the analysis, this subprogram assembles an analysis statement of the form

\[ N_s + N_a \]

where \( N_s \) stands for a sequence of one or more symbol names in the form

\[ \text{The } N_1 \text{ of the } N_2 \ldots \text{ of the } N_j \]

where \( N_1 \) is the name of the symbol about which the statement is made, \( N_2 \) is the name the symbol immediately governing \( N_1 \), and \( N_j \) is the name of the symbol immediately governing \( N_{j-1} \). \( N_a \) is the analysis predicate that applies to \( N_1 \). Examples of the output of this subprogram are contained in Appendix D. The data for a given analysis statement is accumulated in array ACCUM. When the statement is complete, it is printed out and the array is cleared for the next statement.

The operations of the subprogram are as follows:

1. For each pass produced in the analysis the following steps are performed:
2. For each symbol recorded in the given pass, the following steps are performed:
3. Retrieve the symbol number of the J-th symbol
4. Accumulate the name of the J-th symbol from the table of symbol names SYML.
Print J Symbols

Compute Transliteration of Root

Print J Symbols

DO 1 JJ = 1, J

J = No. of Symbols in Present String

IR1 = ITABLE(ILIST1, JJ, 19) + 1
IR2 = ITABLE(ILIST1, JJ, 20) + 1
IR3 = ITABLE(ILIST1, JJ, 21) + 1
IR4 = ITABLE(ILIST1, JJ, 22) + 1

PRINT ITABLE(ILIST1, JJ, L)

End of Do Loop

RETURN

Figure 4.8: Subprogram OUTPUT
(5) Compute the index number of the governing node.

(6) If the symbol is not governed, skip to step (13).

(7) Compute the symbol number of the governing symbol.

(8) If this symbol has previously been treated omit this analysis statement and return back to step (3) for the next symbol on the list.

(9) If the governing symbol is the same as the governed symbol, the information is redundant, skip to step (5).

(10) Accumulate the words "of the" in ACCUM followed by the name of the governing symbol.

(11) Compute the index number of the symbol that governs the present governing symbol.

(12) If the symbol is not governed skip to step (13), otherwise return to step (7).

(13) If the first symbol was a terminal symbol, skip to step (16).

(14) Compute predicate index number.

(15) Print out the content of ACCUM followed by the given predicate from AMSG, and return to step (3) for the next symbol.

(16) Print out the content of ACCUM followed by the word "IS" followed by the English equivalent of the first symbol.

(17) For the subscripts $f$, $k$, $b$, $c$, $l$, $y$, $d$, $r$, $g$, $p$, $r$, $a$, $v$, $i$, and $t$, accumulate (in ACCUM) the semantic values of those subscripts that pertain to the first symbol.

(18) Print out the semantic values accumulated in ACCUM and return to step (3) for the next symbol.

4.3.1.8 Subprogram PROPH1

This subprogram (Figure 4.10) is called by Subprogram RERITE (step 12) in order to predict the possibility of satisfying a given grammar rule if its application were delayed one or two passes. This procedure is used in order to avoid the premature abandonment of a rule which is not being satisfied. Thus in the process of analysis, when a given symbol does not satisfy the requirements of the rule under consideration, before a new rule is requested, this subprogram is called to answer the question "will the requirements of this rule be met if one (or two) other rules are applied first, and if so what is the first rule that should be applied?" If a favorable prediction is made, the analysis is required to take the predicted route before abandoning the rule under consideration. The following operations are performed by this subprogram:
\[ A: (N1 > N2) \text{or} (N1 \leq 0) \]

\[ B: (IS > 76) \text{and} (J > NTERM) \]

Figure 4.9a: Subprogram PARSE
IC = NODE(5,J)
ISS = IS
ICC = IC
M2 = M1 + ISYMB(IS)-1
M3 = 0

Do 1 M = M1, M2
M3 = M3 + 1
ACCUM(M) = SYML(IS,M3)

End of Do Loop

NG = NODE(3,J)

C: (NG ≤ 0) or (NG > NODE2)

(4) Accumulate name of J-th Node

(5) Compute Govern. node

(6) if not governed

(7) Compute gov. symbol type

Figure 4.9b: Subprogram PARSE
D: (ISS=NSG) AND (IC=NSC)
E: (ISS=NSG) AND (ICC=NSC)

M1 = M2 + 1
M2 = M1 + 1
ACCUM(M1) = "OF"
ACCUM(M2) = "THE"
M1 = M2 + 1
M2 = M2 + ISYMB(NSG)
M3 = 0

Do 2 M = M1, M2

M3 = M3 + 1
ACCUM(M) = SYML(NSG, M3)

End of Do Loop

F: (NG = 0) or (NG ≥ NODE2)

Figure 4.9c: Subprogram PARSE
NAN = NAN + 1

Print NAN, ACCUM(M) M=1,M2 "IS".ENGLISH(J,L) L=1,4

M1 = 0

Do 85 M = 3, 17

IMSG = INDEX (IS,IC)

IMSG ≥ 0

T

End of Do Loop

See Page a

End of Do Loop

See Page a

Print NAN, ACCUM(M) M=1,M2 AMSG(IMSG,L)L=1,6

End of Do Loop

See Page a

End of Do Loop

See Page a

RETURN

Figure 4.9d: Subprogram PARSE
(1) A test is made to determine whether the present symbol under consideration (stored in register SYMIN) has failed to meet the requirements of the given rule for other reasons. This is true if present symbol matches the rule element as to symbol type and symbol class. If this is the case, skip to step (11).

(2) Compute the address of those prediction rules that apply to the symbol in SYMIN, by computing
   (a) the starting rule number (IR1) and
   (b) the ending rule number (IR3)

   If there are no prediction rules, skip to step (12).

(3) Compute the address (IR2) of the right hand member of the present prediction rule on the list.

(4) Put the current symbol under consideration in register IS1, the IR1-th rule element in register IS2, the present rule element (defined by IRULE1) in register IS3, and the right hand element of the prediction rule in IS4.

(5) Using Subprogram Macher, determine that the symbols in IS1 and IS2 match, if not skip to step (7).

(6) Using subprogram Macher, determine that the symbols in IS3 and IS4 match, if not skip to step (9).

(7) Advance the index (IR1) of the prediction rule element and of the present symbol (IN)

(8) If IR1 \geq IR2, the symbols in the string will satisfy the prediction rule, skip to step (14), otherwise return to step (4) and repeat from there for the new set of indices.

(9) Using subprogram Proph2, predict satisfaction of the present "prediction" rule at the next pass, if prediction is favorable (ID=4) skip to step (14).

(10) Reset ID to zero and compute the address (IR1) of the next available prediction rule; if there are no more (IR1>IR3) skip to step (13), otherwise return to step (3) for processing the next prediction rule.

(11) No future satisfaction is predicted because the present rule failed to meet the requirements of subprogram SYMACH or LIMIT (previously applied); set ID=1 and return.

(12) There are no rules that apply to the present symbol under consideration, therefore no future satisfaction is predicted; set ID=2 and return.
(13) The prediction rules were exhausted and none predicted future satisfaction, set ID=3 and return.

(14) Satisfaction is predicted if the rule at address IR4 is applied first; set ID=4, IR=IR4, and IM=IR3, and return.

4.3.1.9 Subprogram PROPH2

This subprogram (Figure 4.11) is called by subprogram PROPH1 (step 9) in order to predict the possibility of satisfying a given grammar rule if its application were delayed two passes (see Section 4.3.1.8 for further discussion). The following operations are performed by this subprogram:

(1) Compute the address of those prediction rules that apply to rule element defined by input argument IR, by computing

   (a) the starting rule number (IR1) and
   (b) the ending rule number (IR3)

   If there are no rules, skip to step (7).

(2) Compute the address (IR2) of the right hand member of the present prediction rule on the list.

(3) Put the IR-th rule element in register IS1, the IR1-th element in register IR2, the present rule element (defined by IRULE1) in register IS3, and the IR2-th rule element in register IS4.

(4) Using subprogram MACHER determine that the symbols in IS1 and IS2 match, if not skip to step (6).

(5) Using subprogram MACHER determine that the symbols in IS3 and IS4 match, if not skip to step (6), otherwise to step (9).

(6) Compute the address (IR1) of the next available prediction rule; if there are no more (IR1>IR3), skip to step (8) otherwise return to step (2) for processing the next prediction rule.

(7) There are no rules that apply to the symbol under consideration, therefore, no future satisfaction is predicted; set ID=2 and return.

(8) The prediction rules were exhausted and none predicted future satisfaction; set ID=3 and return.

(9) Satisfaction is predicted by the rule just tested; set ID=4 and return.
Figure 4.10a: Subprogram PROPH1
Figure 4.10b: Subprogram PROPH1

IR2 = IR1 + RULE(IR1,26)
IR4 = IR1

IS1(L) = ITABLE(IN,L)
IS2(L) = RULE(IR1,L)
IS3(L) = RULE(IRULE1,L)
IS4(L) = RULE(IR2,L)

CALL MACHER(IS1,IS2)
MATCH
F
S1
T
CALL MACHER(IS3,IS4)
IS2(2) = 1
F
7
T
CALL PROPH2(ID,IR2)
ID = 4
F
5
T
To Page c

Is there satisfaction at next level?

Does present Rule Match required symbol?

Enter critical elements into registers

Does present Rule Match Input Symbol?

Address of right element

End of Do Loop
Figure 4.10c: Subprogram PROPH1
Figure 4.11a: Subprogram PROPH2

(1) Compute Rule No.

(1a) Starting No.

(1b) Ending No.
Figure 4.11b: Subprogram PROPH2
3.1.10 Subprogram RERITE

This subprogram (Fig. 4.12) is called by the Main Program ANALYZ step 13) to apply the appropriate replacement rule to given symbols and write the replacement symbol on the replace symbol list.

The subprogram performs the following operations:

1. Initialize certain indices
2. Examine the present value of symbol counting index I, if it is greater than the number of symbols in the present pass (IMAX) return, otherwise continue.
3. Increment indices I, LIM, and NODE1, and place the NODE1-th symbol in register SYMTN.
4. Examine the value of index J, if J≠0 skip to step (9), otherwise continue.
5. Using subprogram RULENO, compute the indices of the rules that apply to the present symbol under consideration; store these values in matrix NODE for the symbol under consideration; and compute the index of the right hand element of the first rule.
6. If there is only one rule, clear any rule numbers stored in matrix NODE for the present symbol.
7. If there is no applicable rule skip to step (24), otherwise continue.
8. Initialize certain indices in preparation for checking symbols in the string against the left hand elements of the present rule, place the right hand element of the rule in register SYMBOL, and skip to step (10).
9. Clear any rule numbers stored in matrix NODE for the present symbol.
10. Increment index J and IRULE1, and compare J with JMAX; if J>JMAX skip to step (18), otherwise continue.
11. Using subprogram SYMACH, test to see if the symbol in register SYMIN matches the J-th symbol of the given rule; if they match skip to step (16), otherwise continue.
12. At this point, the symbol under consideration has been found not to match the present rule; however before abandoning the rule, subprogram PROPH1 is used to predict possible future satisfaction of this rule if its application is delayed one or two passes; if the prediction is favorable (ID=4) skip to step (27), otherwise continue.
(13) If the present element of the rule is optional (test A), make certain bookkeeping adjustments and skip back to step (10) to compare the present symbol with the next element of the rule, otherwise continue.

(14) If the present rule involves compounding (J3≠0) skip to step (15), otherwise make certain bookkeeping adjustments associated with abandoning the present rule and return to step (2) for consideration of any other applicable rules.

(15) Make adjustments to certain indices associated with compounding and return to step (4).

(16) At this point, the symbol under consideration has been found to match the J-th element of the given rule, so using subprogram VARATT, the values of the dependent variables associated with the symbol are computed, and the governing mode for the symbol is recorded.

(17) If all the elements of the rule have been satisfied (J=JMAX) skip to step (19), otherwise return to step (2) for further consideration of the remaining elements of the rule.

(18) Reduce the value of indices NODE1 and I by one and continue.

(19) Using subprogram LIMIT check the right hand element of the present rule for any imposed restraints. If the restraints are not met return to step (14) to select a new rule, otherwise continue.

(20) If the present rule involves compounding (J2=1), Increment J3 by 1, transfer the symbol in register SYMBOL to register SYMIN, make bookkeeping adjustments and return to step (4), otherwise continue.

(21) Make adjustments to certain bookkeeping indices, and increment index NODE2 by one.

(22) If the rule involves compounding, make further bookkeeping adjustments.

(23) Write the right hand element of the rule with the computed values of its subscripts (as contained in register SYMBOL) as the NODE2-th symbol of the analysis, and skip to step (30).

(24) If the present rule involves compounding that is not yet complete (test C) return to step (1a) otherwise continue.

(25) If the present rule involves compounding which is now complete (J3≠0) return to step (21), otherwise continue.
If the present rule does not involve compounding (J2=2) skip to step (27), otherwise set J1=0 and J2=2 and return to step (14a).

(27) Make bookkeeping adjustments to certain indices and continue.

(28) If a prediction of future satisfaction of the present rule has been made (ID=4), store the rule number (negated) as a prediction for the next symbol of the derivation and store the number of the prediction rule as a prediction for the present symbol, and in either case continue.

(29) Make bookkeeping adjustments and copy the present symbol (index NODE1) on the next list (index NODE2) and continue.

(30) Initialize certain indices in preparation for the next rule and return to step (1a).

4.3.1.11 Subprogram RULENO

This subprogram (Figure 4.13) is called by subprogram RERITE (step 5) to compute the grammar rule number that applies to a given symbol. The grammar rules are stored in Matrix RULE, and the computed rule numbers are as follows:

IRULE = the number of the first element of the first rule that applies to the given symbol
IMAX = the number of the last element of the last rule that applies to a given symbol.

For simple (non-compounded) symbols the applicable rule numbers initially were computed in Main Program ANALYZ (step 4) and stored in rule catalogue IPSI. The data is stored so that for Symbol IS, Class IC, IRULE = IPSI(IS,IC,1) and IMAX = IPSI(IS,IC,2).

The subprogram computes the rule numbers according to the following hierarchy:

(a) Compound symbols
(b) Simple, non-negated symbols.

The following operations are performed:

(1) If the symbol under consideration is being presented for the first time (J1=0) continue, but if at least one rule has been previously tried (J1≠0) skip to step (18).
Figure 4.12a: Subprogram RERITE
Figure 4.12b: Subprogram RERITE
Figure 4.12c: Subprogram RERITE
Figure 4.12d: Subprogram RERITE
Figure 4.12e: Subprogram RERITE

(29)

\[
\begin{align*}
\text{NODE}(3, \text{NODE1}) &= \text{NODE2} \\
\text{NODE}(1, \text{NODE2}) &= 1 \\
\text{NODE}(4, \text{NODE2}) &= \text{ITABLE(\text{NODE1}, 1)} \\
\text{NODE}(5, \text{NODE2}) &= \text{ITABLE(\text{NODE1}, 6)} \\
\text{ITABLE(\text{NODE2}, L)} &= \text{ITABLE(\text{NODE1}, L)}, L = 1, 29 \\
\end{align*}
\]

(30)

\[
\begin{align*}
J &= 0 \\
J_1 &= 0 \\
J_2 &= 1 \\
J_3 &= 0 \\
\text{NSTART} &= \text{NODE1} \\
\text{MSTART} &= \text{NODE2} \\
\end{align*}
\]

\{Initialize Indices For Next Rule\}
(2) If universal rules should be considered \((J2=1)\) continue, but if phrase rules should be considered \((J2=2)\) skip to step (10).

(3) At this phase the symbol under consideration is being considered for possible application of universal rules, so certain indices are initialized.

(4) If this symbol and the next one are both of the same type (i.e., possible compounding pattern FF), set index IC=4 and skip to step (14), otherwise continue.

(5) If this symbol is followed by a conjunction and another symbol of the same type (i.e., possible compounding pattern F+C+F), skip to step (7) otherwise continue.

(6) If this symbol is followed by a comma, another symbol of the same type, and either a comma or a conjunction (i.e., possible compounding pattern F+, +F+, /C), continue, otherwise skip to step (8).

(7) At this point, it has been determined that the rules governing compounding apply; set IRULE=IPS(97,1,1) the index of the first element of the first compounding rule for class 1; set IMAX=IPS(97,3,2) the index of the last element of the last compounding rule for class 3; set IC=1 and return.\(^9\)

(8) If this point it has been determined that no compounding rules apply to this symbol; if, however, previous compounding rules have been successfully executed \((J3\neq0)\) on this symbol, set IRULE=0, IMAX=0 and return, otherwise continue.

(9) At this point it has been determined that compounding rules are not applicable to this symbol either now or in the past, set \(J2=2\) and continue.

(10) At this point consideration is given to phrase rules (non-universal) that may apply to the symbol; set the index IS=SYMIN(1) the symbol type, and index IC=SYMIN(6) the symbol class and continue.

(11) If the present symbol has a prediction rule recorded for it \([\text{NODE } (6, \text{NODE}) \neq 0]\), assign the indices of that rule to IRULE and IMAX, and return, otherwise continue.

(12) If the present symbol has an undefined class (IC=9) skip to step (17), otherwise continue.

(13) At this point, a test is made for the presence of rules that apply to all classes of the symbol, these rules

---

\(^9\)The computation of this step is dependent on the "content" of the grammar.
are filed under class 1, so here the rules in class 1 are examined; IRULE is set to the index of the first class [IPSI (IS,1,1)], if no rule exists (IRULE=0) skip to step (14); if a rule exists and its class is undefined [RULE (IRULE,6)=9] skip to step (15), otherwise continue.

(14) At this point it has been determined that the grammar has no rules on the given symbol that apply to all classes; set IRULE to the index of the rule for the given class IC [IRULE=IPSI (IS, IC,1)] and continue.

(15) Set IMAX to the index of the last element of the last rule for the given class IC [IMAX=IPSICIS,IC,2)], and continue.

(16) If no rule exists (IRULE=0) skip to step (17); however, if a rule exists, and if IMAX=0, it means that rules exist that govern all classes but none exists for the given class only, set IMAX=IPSI(IS,1,2), and return.

(17) At this point it has been determined that the given symbol has an undefined class, so all available rules on the symbol must be supplied; set IRULE=0 and (17a) find the index of the first element of the first rule that applies to lowest numbered class of the given symbol and if one exists, set IRULE equal to that index; and (17b) find the index of the last element of the last rule that applies to the highest numbered class of the given symbol, and if one exists, set IMAX equal to that index, and return.

(18) At this point it has been determined that a set of rules has already been located for the given symbol and that the most recent one examined was not satisfied; the following sequence of operations computes the index of the next available rule in the set; if the rules of the set are "universal" rules [IRULE > IPSI (97,1,1)], set index J2=2, and in either case continue.10

(19) Set the index IS (symbol type) equal to the type for the most recent rule [IS=RULE(IRULE,1)]; and set the index IC (symbol class) equal to the class of the most recent rule [IC=RULE(IRULE,6)], with the exception that if the rules are universal (IS=97), IC is set equal to the value of subscript f [IC=RULE(IRULE,3)], then continue.10

10The computation of this step is dependent on the "content" of the grammar.
(20) Set the value of IRULE to its present value plus the number of elements in the specified rule, plus 1 (the index of the next rule in the set), with the exception that if the most recent rule was the last rule (IRULE > IMAX), set both IRULE and IMAX to zero, and return.

4.3.1.12 Subprogram SYMACH

This subprogram (Figure 4.14) is called by Subprogram RERITE (step 11) to check that the "present rule symbol" matches the "present constituent" with respect to the criteria of the grammar. The subprogram performs the following functions:

1. Initializes the logic monitor MATCH to "true".
2. If the symbol types do not match or if the "present rule symbol" is a variable symbol, skip to step 14.
3. For each subscript up to 17 perform steps 4 through 15.
4. If subscript 3, and the "present rule symbol" is a variable symbol, skip to step 15.
5. If subscript 7, and the "present rule symbol" is not a negative, skip to step 15.
6. If the value of the given subscript of the "present rule symbol" is 9, that is, it is an independent variable, skip to step 15.
7. If the value of the given subscript of the "present rule symbol" equals the value of the subscript of the "present constituent," skip to step 15.
8. If the given subscript of the "present constituent" is an independent variable skip to step (15).
9. If the given subscript is a negative number for the "present rule symbol" and it is a fixed value (<9) for the "present constituent", skip to step (15).
10. If the given subscript is not a dependent variable for the "present rule symbol", skip to step (14).
11. If the given subscript of register SYMBOL is equal to the value specified by the rule, skip to step (15).
12. If the given subscript of register SYMBOL is less than 9 and equal to the value of the corresponding subscript of the "present constituent", skip to step (15).
First Check for Compounding

A: Is pattern FF?
B: Is pattern F+C+F
C: Is pattern F+g+F+{c}*

Figure 4.13a: Subprogram RULENO
(8) No Compounding Rules apply; were Compounding Rules in Operation?

(9) Go to "Phrase" rules

(10) Assign "Phrase" Rules

(11) Is There a prev. prediction?
D: NODE(6,NODE1) = 0

(12) Is Class undefined?

Figure 4.13b: Subprogram RULENO
(13) Try for rule on undefined class

If none, use rule for class IC

E: RULE(IRULE,6) = 9

(14)

IRULE = IPSI(IS,IC,1)

(15)

IMAX = IPSI(IS,IC,2)

(16) If no rule try for any rule on IS.

IRULE = 0

IMAX = 0

Figure 4.13c: Subprogram RULENO
Find rules for all classes

\[ G: \text{IPSI(IS,L,1)} = 0 \]
\[ H: \text{IPSI(IS,L,2)} = 0 \]

Figure 4.13d: Subprogram RULENO
(18) If Compounding
F: IRULE>IPSI(97,1,1)

(19) If Compounding,
Class is governed by
Subscript f.

(20) Compute next
Available Rule

If no more,
Initialize

Figure 4.13.e: Subprogram RULENO
Subprogram SMYACH

A: SYMIN(1) = RULE(IRULE1),
B: (RULE(IRULE1,1) = 97) and (SYMBOL(1) = 97)
C: (RULE(IRULES,1) = 97) and (SYMIN(1) = SYMBOL(1))
D: (L=3) and (RULE(IRULE1,1) ≠ 97)
E: (L=7) and (SYMIN(8) = 0)
F: (RULE(IRULE1,L) = 97)
G: SYMIN(L) = RULE(IRULE1,L))
H: SYMIN(L) = 9
I: (RULE(IRULE1,L) <0) and (SYMIN(L) <9)
J: (RULE(IRULE1,L) >9) and (RULE(IRULE1,L) = RULE(JJ,L))
K: SYMBOL(L) = RULE(IRULE1,L)
L: (SYMBOL(L) <9) and (SYMBOL(L) = SYMIN(L))
M: (SYMBOL(L) = 9)
N: SYMBOL(L) = 9

Figure 4.14: Subprogram SYMACH
(13) If the given subscript of register SYMBOL is 9, skip to step (15).

(14) Set the logic monitor MATCH to "false" and return.

(15) If the last subscript (L=17), skip to step 16, otherwise increment L and return to step (3).

(16) Using subprogram LIMIT, test the symbol for any specified restraints, and return.

4.4.1.13 Subprogram VARATT

This subprogram (Fig. 4.15) is called by subprogram RERITE (step 16) to compute the value of the various dependent variable subscripts for the "present rule symbol." For a given rule, certain of the subscripts may be designated as dependent variables—that is, the values of these subscripts depend on the values of the corresponding subscripts in the "present constituent." These dependent variable subscripts are written as lower case alphabetic characters in the rules; these were transformed to integer numbers when the rules were read in by the main program (step 3). Reference to Table 4-1 indicates that all alphabetic characters have been transformed to an integer value greater than 9, so that this test is used to identify dependent variable subscripts.

This subprogram performs the following operations:

(1) If the "present rule symbol" is a variable symbol (used in rules for negating or compounding), set the symbol type subscript of the "present rule symbol" to the same value as that of the "present constituent" and store this value in register SYMBOL.

(2) For subscripts 2 through 22, perform steps 3 through 5.

(3) If the given subscript of the "present rule symbol" is -1, set the corresponding subscript of the register SYMBOL to 1 greater than the value of the subscript of the "present constituent," and in either case continue.

(4) If the given subscript of the "present rule symbol" is a dependent variable (>9), set the corresponding subscript of register SYMBOL equal to the value of the subscript of the "present constituent," and in either case continue.
(5) If not the 22-nd subscript, advance index L by one and return to step (3), otherwise continue.

(6) If the symbol type of the "present constituent" is not the same as the symbol stored in register SYMBOL, return, otherwise continue.

(7) If the symbol stored in register SYMBOL indicates that compounding after pattern 2 is being performed, continue, otherwise return.

(8) For symbols being compounded after pattern 2, the following operations are performed:

(9) For phrases with conjunctive compounding (b=1), the number feature of the phrase is set to plural (n=3).

(10) For phrases with undefined number (n=0), the number of the phrase is set to that of the "present constituent."

(11) For phrases with undefined gender (g=0), the gender of the phrase is set to that of the "present constituent."

(12) For phrases with undefined person (p=0), the person of the phrase is set to that of the "present constituent."

(13) For the feature number, the phrase is assigned the largest value found in itself or any of its like constituent elements.

(14) For the feature gender, the phrase is assigned the smallest value found in any of its like constituent elements.

(15) For the feature person, the phrase is assigned the smallest value found in any of its like constituent elements.

(16) Return to calling program.

These computations may be found to be dependent on the "content" of the grammar in that they may not be universally true for all Semitic languages.
4.3.2 Input Map for Computer Program

This section describes the input data required to use the computer program of the Algorithm for Analyzing Hebrew Sentences. Three types of data cards are required:

1. Program option card,
2. Data cards describing the grammar and its rules.
3. Data cards describing the sentences to be analyzed.

The following sections describe each type of card.

4.3.2.1 Program Option Card

This card is used to control the various options available in the program. The data and the associated options are as follows:

<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1-5       | 15     | MODE     | = 0, grammar rule are to be read in from cards  
          |        |          | = 1, grammar rules are stored on tape ITAPE. |
| 6-10      | 15     | ITAPE    | logical unit number of magnetic tape for storage of grammar rules. |
| 11-15     | 15     | ITRACE   | = 0, no diagnostic messages to be printed out.  
          |        |          | = 1, print out diagnostic messages of program. |
| 16-20     | 15     | IOUTPT   | = 0, do not print out full description of nodes at every level of synthesis.  
          |        |          | = 1, print out full description of nodes at every level of synthesis. |
| 21-25     | 15     | IOUTRE   | = 0, do not print out tree diagram of synthesis,  
          |        |          | = 1, print out tree diagram. |
| 26-30     | 15     | LIMAX    | Maximum number of symbol tests. |

4-83
A: \((\text{RULE}(\text{IRULE1},1)=97)\) (1) 
And \((\text{SYMBOL}(1)=97)\)

B: \(\text{RULE}(\text{IRULE1},L)=-1\)

C: \((\text{RULE}(\text{IRULE1},L)>9)\) 
And \((\text{SYMBOL}(L)=\text{RULE}(\text{IRULE1},L))\)

D: \(\text{SYMBOL}(1)\neq\text{SYMIN}(1)\)

E: \((\text{SYMBOL}(3)=2)\) 
And \((\text{SYMBOL}(4)\neq0)\)

Figure 4.15a: Subprogram VARATT
Figure 4.15b: Subprogram VARATT
The variable MODE is used to control the source of the grammar rules used by the program. Initially the rules are read in from cards (MODE=0) and stored on the magnetic tape mounted on logical unit ITAPE. On subsequent runs the rules are read from the tape (MODE=1) unless changes in the rules are to be made.

Variable ITAPE defines the logical unit number containing the magnetic tape on which the grammar rules are stored (MODE=1) or are to be stored (MODE=0).

Variable ITRACE is used for tracing the progress of the program throughout its operation. If ITRACE=1, the program prints out numerous messages together with the values of certain critical variables at key steps of the program. The use of this variable is for program debugging only, it should normally be set to zero.

Variable IOUTPT is used to control the output of the computer. The computer will list a complete description of each nodal point at every level of the synthesis if IOUTPT = 1 (see Appendix C for a sample of this output). This output is useful when the user wants to examine the output results in fine detail, otherwise IOUTPT should be zero.

Variable IOUTRE is used to control the output of the computer. The computer will construct and print out a tree diagram of the analyzed sentence. Each nodal point is identified as to symbol type and class only (see Section 2.3.2 of Part II of this report for a sample of this output). If more detailed data are required the IOUTPUT option should be used also. Note: either IOUTPT or IOUTRE (or both) must be 1.

Variable LIMAX is used to control the maximum number of symbol test permitted in a given analysis. If an analysis is not found in less than LIMAX symbol tests, the program terminates the analysis and goes to the next sentence. This prevents the program from continuing to search for analyses beyond a reasonable limit.

4.3.2.2 Program Grammar Cards

The program grammar cards consists of four types:

Type 2-1: Transliteration Table
Type 2-2: Grammar Rules
Type 2-3: Restraint Matrix
The program grammar cards are not required if MODE=1; in this case these data are read from the magnetic tape (ITAPE). The following sections define the program grammar cards.

4.3.2.2.1 Transliteration Table: Card 2-1

This card defines the transliteration Table (see Table 4-1). The data are as follows:

<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>16</td>
<td>NL</td>
<td>= number of elements in Table (50 max.).</td>
</tr>
<tr>
<td>7-56</td>
<td>50A1</td>
<td>TRANS(L)</td>
<td>Transliteration Table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L=1,NL</td>
<td></td>
</tr>
</tbody>
</table>

4.3.2.2.2 Grammar Rules: Cards 2-2

These cards define the rules of the grammar. Each card defines one constituent of a rule. Thus the rule

\[ B + C = A \]

requires three cards: the first defines B, the second C, and the third A. A maximum of 800 cards is permitted. A card with 999 in columns 1-3 should follow the last grammar rule card. The rules should be placed in numeric order by rule number. See Appendix A for a complete listing of the rules used for the present research. The following is the data format for each card:
<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>A3</td>
<td>SN</td>
<td>Symbol number (See Table 4-2)</td>
</tr>
<tr>
<td>6</td>
<td>A1</td>
<td>M</td>
<td>Subscript m - optional/mandatory</td>
</tr>
<tr>
<td>9</td>
<td>A1</td>
<td>F</td>
<td>Subscript f - compound class</td>
</tr>
<tr>
<td>11-12</td>
<td>A2</td>
<td>K</td>
<td>Subscript k - compound number</td>
</tr>
<tr>
<td>15</td>
<td>A1</td>
<td>B</td>
<td>Subscript b - compound type</td>
</tr>
<tr>
<td>18</td>
<td>A1</td>
<td>C</td>
<td>Subscript c - symbol class</td>
</tr>
<tr>
<td>21</td>
<td>A1</td>
<td>&amp;</td>
<td>Subscript &amp; - negative class</td>
</tr>
<tr>
<td>24</td>
<td>A1</td>
<td>Y</td>
<td>Subscript y - negative/positive</td>
</tr>
<tr>
<td>27</td>
<td>A1</td>
<td>D</td>
<td>Subscript d - definiteness</td>
</tr>
<tr>
<td>30</td>
<td>A1</td>
<td>N</td>
<td>Subscript n - number</td>
</tr>
<tr>
<td>33</td>
<td>A1</td>
<td>G</td>
<td>Subscript g - gender</td>
</tr>
<tr>
<td>36</td>
<td>A1</td>
<td>P</td>
<td>Subscript p - person</td>
</tr>
<tr>
<td>39</td>
<td>A1</td>
<td>R</td>
<td>Subscript r - preposition class</td>
</tr>
<tr>
<td>42</td>
<td>A1</td>
<td>A</td>
<td>Subscript a - verb class</td>
</tr>
<tr>
<td>45</td>
<td>A1</td>
<td>V</td>
<td>Subscript v - voice</td>
</tr>
<tr>
<td>48</td>
<td>A1</td>
<td>I</td>
<td>Subscript i - mood</td>
</tr>
<tr>
<td>51</td>
<td>A1</td>
<td>T</td>
<td>Subscript t - tense</td>
</tr>
<tr>
<td>53-54</td>
<td>A2</td>
<td>S</td>
<td>Subscript s - stem</td>
</tr>
<tr>
<td>57</td>
<td>A1</td>
<td>W1</td>
<td>Subscript w₁ - root letter 1</td>
</tr>
<tr>
<td>60</td>
<td>A1</td>
<td>W2</td>
<td>Subscript w₂ - root letter 2</td>
</tr>
<tr>
<td>63</td>
<td>A1</td>
<td>W3</td>
<td>Subscript w₃ - root letter 3</td>
</tr>
<tr>
<td>66</td>
<td>A1</td>
<td>W4</td>
<td>Subscript w₄ - root letter 4</td>
</tr>
<tr>
<td>69</td>
<td>A1</td>
<td>ST</td>
<td>Symbol Type</td>
</tr>
<tr>
<td>70-72</td>
<td>I3</td>
<td>RN</td>
<td>Rule Number</td>
</tr>
<tr>
<td>73</td>
<td>I1</td>
<td>EN</td>
<td>Element Number</td>
</tr>
<tr>
<td>74</td>
<td>I1</td>
<td>NRH</td>
<td>No. of Rt. Hand Elements</td>
</tr>
<tr>
<td>75-76</td>
<td>I2</td>
<td>RT</td>
<td>Restraint Type</td>
</tr>
<tr>
<td>77-78</td>
<td>I2</td>
<td>RS</td>
<td>Restraint Subscript</td>
</tr>
</tbody>
</table>

Variable SN is the symbol number of the given rule element as specified on Table 4-2. A card with SN=999 should follow the last grammar rule.
Variables $m$, $f$, $k$, $b$, $c$, $l$, $y$, $d$, $g$, $p$, $x$, $v$, $i$, $r$, $s$, $w_1$, $w_2$, $w_3$, and $w_4$ are the subscripts of the symbols are defined in Part II, Section 2.2.1.

Variable $ST$ identifies the symbol as follows:

$$ST = 0 \text{ for non-terminal symbols}$$
$$ST = 1 \text{ for terminal symbols}$$

Variable $RN$ is the rule number associated with the given card. For example: in the rule

$$B + C = A \quad (52.1)$$

Variable $RN = 521$ for all cards of the rule. Variable $EN$ is the element number of the symbol defined by the card. In the above example:

- on the card defining $B$, $EN=1$
- on the card defining $C$, $EN=2$
- on the card defining $A$, $EN=0$

Variable $NRH$ defines the number of left hand elements in the rule. In the above example, $NRH=2$ for all cards of the rule.

Variable $RT$ defines the restraint type that applies to the symbol. The value of $RT$ refers to a row of restraints in the Restraint Matrix which is defined by Cards 2-3. Variable $RS$ defines the subscript of the symbol to which the restraint applies. For example,

$$RT = 3, RS = 10, \text{ means that restraint type 3 applies to the 10-th subscript n}.$$  

4.3.2.2.3 Restraint Matrix: Cards 2-3

These cards define the restraints that are placed on a given symbol in a grammar rule. See Section 4.3.1.4 for a discussion and explanation of the data in Matrix $RESTRT$. The following is format of the data:

<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>I5</td>
<td>$IM$</td>
<td>Number of restraints on I-th list (Max.=4)</td>
</tr>
<tr>
<td>6-25</td>
<td>4I5</td>
<td>$RESTRT(J,I)$</td>
<td>J=2, IM + 1</td>
</tr>
</tbody>
</table>

Number of cards 15.

The data on these cards are listed in Table 4-9.
### Table 4-9

**RESTRAINT MATRIX DATA**

<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Card No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>6-10</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-2</td>
</tr>
<tr>
<td>1</td>
<td>-3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### 4.3.2.2.4 The Table of Symbols: Cards 2-4

These cards define the symbols used in the grammar. See Section 4.1.1.2 for further discussion. Ten cards are required for these data which are in 10(2X,A3) format. The data for the Hebrew grammar are given in Table 4-10.

#### 4.3.2.2.5 The Table of Symbol Names: Cards 2-5

These cards define the English name of each symbol used in the grammar. See Section 4.1.1.5 for further discussion. Ninety-six cards are required for these data (one for each symbol) which are in the following format:
Table 4-4 contains the symbol names for the Hebrew grammar.

### Table 4-10

<table>
<thead>
<tr>
<th>Card Column</th>
<th>Card No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-5</td>
<td>6-10</td>
</tr>
<tr>
<td>Z-</td>
<td>VQO</td>
</tr>
<tr>
<td>BAA</td>
<td>BAB</td>
</tr>
<tr>
<td>BP</td>
<td>NPR</td>
</tr>
<tr>
<td>EA</td>
<td>VB</td>
</tr>
<tr>
<td>RSP</td>
<td>NSP</td>
</tr>
<tr>
<td>VMR</td>
<td>VMI</td>
</tr>
<tr>
<td>SAA</td>
<td>SAB</td>
</tr>
<tr>
<td>KI</td>
<td>KD</td>
</tr>
<tr>
<td>E-</td>
<td>G-</td>
</tr>
<tr>
<td>R-</td>
<td>T-</td>
</tr>
</tbody>
</table>

4.3.2.2.6 The Predicate Index: Card 2-6

These cards provide an index for cataloging the analysis predicates. See Section 4.1.1.6 for further discussion. One-hundred cards are required for these data which are in the following format:

<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-40</td>
<td>815</td>
<td>INDEX(I,J)</td>
<td>Index number of the analysis predicate for the I-th symbol, J-th class.</td>
</tr>
</tbody>
</table>

Table 4-6 contains the data for the Hebrew grammar.

4.3.2.2.7 The Analysis Predicates: Card 2-7

These cards provide the list of analysis predicates used in the algorithm. See Section 4.1.1.6 for further discussion. One-hundred eighty-five cards are required for these data (one for each predicate) which are in the following format:
Table 4-5 contains the list of analysis predicates for the Hebrew grammar.

### 4.3.2.2.8 The Feature Values: Cards 2-8

These cards provide the table of feature values used in the grammar. See Section 4.1.1.7 for further discussion. Seventeen (17) cards are required for these data (one for each subscript involved) which are in the following format:

<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
</table>

Table 4-7 contains a list of the Feature Values for the Hebrew grammar.

### 4.3.3 The Sentence Description Cards

The sentence analysis algorithm uses a set of data cards to describe the Hebrew sentence being analyzed. These cards consist of two types:

- **Type 3-1**: Hebrew Sentence Card
- **Type 3-2**: Word Description Cards

A set of sentence description cards is required for each sentence to be analyzed. A blank card should follow the last set of sentence description cards.

#### 4.3.3.1 Hebrew Sentence: Card 3-1

This card provides a listing (in transliterated characters) of the Hebrew sentence to be analyzed, a sentence number assigned by the user, and the number of words in the sentence. The transliterated Hebrew sentence is printed on the tree diagram analysis computed by the algorithm. The sentence number is used to identify the sentence in the analysis statements assembled by the algorithm. One card type 3-1 is required for each sentence to be analyzed. The following is the data format:
4.3.3.2 Word Description: Cards 3-2

These cards provide a complete grammatical description of each word in the sentence being analyzed, one card for each word, prefix (including definite article), pronominal suffix, and punctuation mark. These data provide the equivalent of the output of an automatic word analysis algorithm which presently is not incorporated in the program. Whenever the word analysis algorithm is included, these no longer will be required. The following is the format of the data:

<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>I6</td>
<td>SN(I)</td>
<td>Symbol Number of the I-th word (see Table 4-2). Only values between 77 and 96 are used.</td>
</tr>
<tr>
<td>11-12</td>
<td>I2</td>
<td>M(I)</td>
<td>Subscript ( m ) of the I-th word. Always = 1.</td>
</tr>
<tr>
<td>19-20</td>
<td>I2</td>
<td>C(I)</td>
<td>Symbol Class of the I-th word</td>
</tr>
<tr>
<td>21-22</td>
<td>I2</td>
<td>L(I)</td>
<td>Negative class of the I-th word</td>
</tr>
</tbody>
</table>
| 27-28     | I2     | N(I)     | Number attribute of I-th word  
|           |        |          | = 0 - does not apply 
|           |        |          | = 1 - singular 
|           |        |          | = 2 - dual 
|           |        |          | = 3 - plural |
| 29-30     | I2     | G(I)     | Gender attribute of I-th word.  
|           |        |          | = 0 - does not apply 
|           |        |          | = 1 - masculine 
|           |        |          | = 2 - feminine |
| 31-32     | I2     | P(I)     | Personal attribute of I-th word  
|           |        |          | = 0 - does not apply 
|           |        |          | = 1 - first person 
|           |        |          | = 2 - second person 
|           |        |          | = 3 - third person 

Most nouns are third person.
<table>
<thead>
<tr>
<th>Card Col.</th>
<th>Format</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>33-34</td>
<td>I2</td>
<td>R(I)</td>
<td>Subscript r for I-th word.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 0 for all words except prepositions and verbs</td>
</tr>
<tr>
<td>35-36</td>
<td>I2</td>
<td>A(I)</td>
<td>Subscript a for I-th word.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 0 for all words except verbs.</td>
</tr>
<tr>
<td>37-38</td>
<td>I2</td>
<td>V(I)</td>
<td>Voice attribute of I-th word.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 0 for all words except verbs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1 - active voice</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 2 - passive voice</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 3 - reflexive voice</td>
</tr>
<tr>
<td>39-40</td>
<td>I2</td>
<td>IM(I)</td>
<td>Mood attribute of I-th word.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 0 for all words except verbs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1 - indicative mood</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 2 - imperative mood</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 3 - subjunctive mood</td>
</tr>
<tr>
<td>41-42</td>
<td>I2</td>
<td>T(I)</td>
<td>Tense attribute of I-th word</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 0 for all words except verbs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 1 - past tense</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 2 - future tense</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This attribute applies to the tense inflection of the specific word itself, not to modifications of tense due to auxiliaries or adverbs. Present tense verb is classified as a participle.</td>
</tr>
<tr>
<td>43-44</td>
<td>I2</td>
<td>S(I)</td>
<td>Suffix s (stem) for the I-th word.</td>
</tr>
<tr>
<td>45-48</td>
<td>A4</td>
<td>W(I)</td>
<td>The four-letter root of the I-th word. Roots are in English transliteration. For non-Hebrew words, omit data.</td>
</tr>
<tr>
<td>49-72</td>
<td>4A6</td>
<td>ENGLISH</td>
<td>The English equivalent of the Hebrew word.</td>
</tr>
</tbody>
</table>

Variables SN, C, L, R, A, V, S, W, and ENGLISH are obtained from the dictionary (see Part II of this report, Appendix A). Variables N, G, P, IM, and T, are defined by the inflection of the given word as found in the sentence.
4.4 Test and Demonstration of Algorithm

In order to test the computerized algorithm and to verify the grammar of modern Hebrew syntax, a total of 26 sentences were analyzed on the computer.

The sentences correspond to those generated by the synthesis algorithm (see Part III), except that not all are included and some are in simplified form.

Computations were performed on a UNIVAC 1108. No record was kept of the computing time required to generate a sentence, however, a conservative estimate would be about 2.0 seconds of computer time per sentence. Computing time is roughly proportional to the length and complexity of the sentence.

4.4.1 Sentences Analyzed by Algorithm

This section contains a list of the Hebrew sentences analyzed by the computerized algorithm preceded by their English equivalent. The number accompanying each pair of sentences refers to the corresponding tree diagrams generated by the algorithm which are contained in Section 2.3.2 of Part II of this report.

1 Chaim Nuchman Bialik was a great poet in the land of Israel. XYYM NXMN BYALYQ HYH MSWRR GDWL BAR& YSRAL.

101 Was Chaim Nuchman Bialik a great poet in the land of Israel? HAM XYYM NXMN BYALYQ HYH MSWRR GDWL BAR& YSRAL?

102 Chaim Nuchman Bialik be a great poet in the land of Israel! XYYM NXMN BYALYQ HYH MSWRR GDWL BAR& YSRAL!

103 Chaim Nuchman Bialik will be a great poet in the land of Israel! XYYM NXMN BYALYQ YHYH MSWRR GDWL BAR& YSRAL.

2 He was greater than all the poets. HWA HYH GDWL MKL HMSWRRYLM.

201 Who was greater than all the poets? MY HYH GDWL MKL HMSWRRYLM?

4 He lived in a small village. HWA YSB BKPR Q@N.

401 He will live in a small village. HWA YYSB BKPR Q@N.

This illustrates the imperative sentence. The HYH is the imperative ח"ש.
(402) He is living in a small village.
HWA YWSB BKPR Q@N.

(403) He used to sit on a small chair.
HWA HYH YWSB OL KCA Q@N.

(404) He will continually live in a small village.
HWA YHYH YWSB BKPR Q@N.

(5) He loved to study law.
HWA AHB LLMWD TWRH

(6) But he also loved the fields and the forests.
ABL HWA AHB GM AT ESDWT WAT HYORYM

(7) He went to study in a large academy.
HLK LLMWD BSYBH GDWLH.

(701) Did he go to study in a large academy?
HHLK13 LLMWD BSYBH GDWLH?

(702) Did he go to study in a large academy?
HAM14 HLK LLMWD BSYBH GDWLH?

(703) Did Bialik go to study in a large academy?
HAM BYALYQ HLK LLMWD BSYBH GDWLH?

(8) He wrote the first poems.
HWA KTB AT HSYRYM HRASWNYM

(9) He wrote about the sun.
HWA KTB OL HSM.

(12) The students studies these poems.
HTLMYDYM LMDW AT HSYRYM HALH.

(121) They knew them by heart.
YDOW AWTM OL PH.

(13) Bialik traveled to the land of Israel.
BYALYQ NCO LAR& YSRAL.

(23) All the Jews wept over his death.
KL HYHWDYM BKW OL MWTW.

(26) Do you want to see the synagogue?
HAM ATH RW&H LRAWT AT BYT HKNCT?

(11) Joseph does not have a room.
AYN LYWCP XDR.

(112) Joseph has a room
YS LYWCP XDR.

13 This illustrates the classical Hebrew option.
14 This illustrates the modern Hebrew option.
The 26 sentences analyzed by the computer were selected to test the algorithm. In analyzing these sentences, 57 of the 179 rules on the intermediate symbols were used. Of the 73 intermediate symbols, 40 were tested for at least one class, and 15 were tested for all class variants. Of the 20 terminal symbols, 16 were tested for at least one class, and 10 were tested for all class variants. The purpose of this test was to verify the analytic capability of the algorithm, not to further verify the grammar rules. Time limitation prevented more extensive tests. However, sufficient tests were performed to demonstrate the analytic capability of the algorithm and to reveal the areas where improvement is required.

Table 4-11 is a listing of the sentence numbers of the test sentences that demonstrate the symbols of the grammar and thus, the corresponding rules. For example, for symbol No. 6, Na, test Sentences 1 and 4 (plus others) demonstrate the use of class 1 (Rule 9.1), test Sentences 2 and 6 (plus others) demonstrate class 2 (Rule 9.3), test sentences 1 and 701 (plus others) demonstrate class 3 (Rule 9.4); the asterisk (*) indicates that the symbol does not have classes 4 through 8. Tree diagrams of the test sentences are contained in Section 2.3.2 of Part II of this report which show the specified symbol as it relates to the structure of the sentence.

Because the rules of the grammar are unordered (by original definition), the analysis algorithm requires predictive logic to synchronize the production of related constituents. This logic computes the possibility of satisfying a grammar rule if its application is postponed one or more passes on the string of symbols. This feature causes the algorithm to delay execution of high level rules until applicable lower level rules have been satisfied.

The analysis algorithm is presently equipped with prediction logic to a depth of two passes by the use of Subprograms PROPH1 and PROPH2. With this depth of predictive logic, the algorithm is capable of analyzing most simple sentences. Difficulties are experienced with complex sentences that contain compound verb modifying phrases, compound predicates, predicates with verbs of class 4, 7, and 8, and with sentences containing subordinate clauses or relative clauses. The analysis of these sentences requires predictive logic that is more powerful. This logic must delay the execution of a rule until a deep structure relationship with the preceding symbol is predicted for the next pass on the string. Time has not permitted the incorporation of this feature into the algorithm, but there is nothing to prevent this and other refinements from being added in the future.
Table 4-11
SENTENCES ILLUSTRATING GRAMMAR SYMBOLS

<table>
<thead>
<tr>
<th>Symbol No.</th>
<th>Symbol Name</th>
<th>Symbol Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Z</td>
<td>*</td>
</tr>
<tr>
<td>2</td>
<td>Vgφ</td>
<td>*</td>
</tr>
<tr>
<td>3</td>
<td>Apa</td>
<td>4,7,8 1,2</td>
</tr>
<tr>
<td>4</td>
<td>Ap</td>
<td>1,2,4,7</td>
</tr>
<tr>
<td>5</td>
<td>As</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Na</td>
<td>1,4+ 2,6+ 1,701+</td>
</tr>
<tr>
<td>7</td>
<td>Sqφ</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Ns</td>
<td>23</td>
</tr>
<tr>
<td>9</td>
<td>Rd</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>Ro</td>
<td>121</td>
</tr>
<tr>
<td>11</td>
<td>Baa</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Bab</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Bac</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Bad</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Bae</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Baf</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Bba</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Bbb</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Ba</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Bbc</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Bp</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Npb</td>
<td>1,2,5+ 23 403</td>
</tr>
<tr>
<td>23</td>
<td>Npa</td>
<td>1,4+ 1,2,13+</td>
</tr>
<tr>
<td>24</td>
<td>Npc</td>
<td>*</td>
</tr>
</tbody>
</table>
Table 4-11 (continued)

SENTENCES ILLUSTRATING GRAMMAR SYMBOLS

<table>
<thead>
<tr>
<th>Symbol No.</th>
<th>Symbol Name</th>
<th>Symbol Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>Nap</td>
<td>1,101+</td>
</tr>
<tr>
<td>26</td>
<td>Np</td>
<td>1,2,5+</td>
</tr>
<tr>
<td>27</td>
<td>(Dpa)¹⁵</td>
<td>*</td>
</tr>
<tr>
<td>28</td>
<td>(Dpb)¹⁵</td>
<td>*</td>
</tr>
<tr>
<td>29</td>
<td>(Dpc)¹⁵</td>
<td>*</td>
</tr>
<tr>
<td>30</td>
<td>Dp</td>
<td>6</td>
</tr>
<tr>
<td>31</td>
<td>Ea</td>
<td>402,26+</td>
</tr>
<tr>
<td>32</td>
<td>Vb</td>
<td>1,4,5+</td>
</tr>
<tr>
<td>33</td>
<td>Vbb</td>
<td>1,2,4+</td>
</tr>
<tr>
<td>34</td>
<td>Vc</td>
<td>1,2,5+</td>
</tr>
<tr>
<td>35</td>
<td>Vaa</td>
<td>1,2,+</td>
</tr>
<tr>
<td>36</td>
<td>Va</td>
<td>1,2,7,+</td>
</tr>
<tr>
<td>37</td>
<td>Bc</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Xp</td>
<td>1,2,4+</td>
</tr>
<tr>
<td>39</td>
<td>No</td>
<td>6,8+</td>
</tr>
<tr>
<td>40</td>
<td>Dpd</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Rsp</td>
<td>2,4,5+</td>
</tr>
<tr>
<td>42</td>
<td>Nsp</td>
<td>A,1,23+</td>
</tr>
<tr>
<td>43</td>
<td>Nop</td>
<td>5</td>
</tr>
<tr>
<td>44</td>
<td>Nip</td>
<td>121</td>
</tr>
<tr>
<td>45</td>
<td>Npx</td>
<td>201</td>
</tr>
<tr>
<td>46</td>
<td>Vma</td>
<td>5,6+</td>
</tr>
<tr>
<td>47</td>
<td>Vmb</td>
<td>121</td>
</tr>
<tr>
<td>48</td>
<td>Vmc</td>
<td></td>
</tr>
</tbody>
</table>

¹⁵Symbols Dpa, Dpb, and Dpc have been removed from the grammar as a result of modifications made on this project.
Table 4-11 (continued)
SENTENCES ILLUSTRATING GRAMMAR SYMBOLS

<table>
<thead>
<tr>
<th>Symbol No.</th>
<th>Symbol Name</th>
<th>Symbol Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>Vmd</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>50</td>
<td>Vm</td>
<td>1,2, 5 6+ 121 4,7+ 5,7+</td>
</tr>
<tr>
<td>51</td>
<td>Vmr</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>52</td>
<td>Vmi</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>53</td>
<td>Vp</td>
<td>1,2, 5+ 1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>54</td>
<td>Vrb</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>55</td>
<td>Vri</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>56</td>
<td>Nv</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>57</td>
<td>Nw</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>58</td>
<td>Epb</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>59</td>
<td>Epa</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>60</td>
<td>Ep</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>61</td>
<td>Saa</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>62</td>
<td>Sab</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>63</td>
<td>Sac</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>64</td>
<td>Sa</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>65</td>
<td>Sro</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>66</td>
<td>Sri</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>67</td>
<td>Rg</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>68</td>
<td>Km</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>69</td>
<td>Kc</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>70</td>
<td>Kk</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>71</td>
<td>Ki</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>72</td>
<td>Kd</td>
<td>1 2 3 4 5 6 7 8</td>
</tr>
</tbody>
</table>
Table 4-11 (continued)
SENTENCES ILLUSTRATING GRAMMAR SYMBOLS

<table>
<thead>
<tr>
<th>Symbol No.</th>
<th>Symbol Name</th>
<th>Symbol Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>Sd</td>
<td>1,2,7+</td>
</tr>
<tr>
<td>74</td>
<td>S</td>
<td>1,2,7+</td>
</tr>
<tr>
<td>75</td>
<td>S_i</td>
<td>701,201</td>
</tr>
<tr>
<td>76</td>
<td>S_c</td>
<td>1,2,4+, 201,701+, 101</td>
</tr>
<tr>
<td>77</td>
<td>A</td>
<td>1,2,4+,</td>
</tr>
<tr>
<td>78</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>D</td>
<td>6</td>
</tr>
<tr>
<td>81</td>
<td>E</td>
<td>26,402+,</td>
</tr>
<tr>
<td>82</td>
<td>G</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>H</td>
<td>2,6+</td>
</tr>
<tr>
<td>84</td>
<td>I</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>J</td>
<td>1,2,13+</td>
</tr>
<tr>
<td>86</td>
<td>L</td>
<td>111</td>
</tr>
<tr>
<td>87</td>
<td>N</td>
<td>1,2,4+, 111,112, 1,701+</td>
</tr>
<tr>
<td>88</td>
<td>O</td>
<td>6,8,12</td>
</tr>
<tr>
<td>89</td>
<td>P</td>
<td>1,2,5,8,13+</td>
</tr>
<tr>
<td>90</td>
<td>Q</td>
<td>701+, 26,702</td>
</tr>
<tr>
<td>91</td>
<td>R</td>
<td>12, 2,4,5+, 121,23+, 201,701, 1,2,4+</td>
</tr>
<tr>
<td>92</td>
<td>T</td>
<td>201,701, 1,2,4+</td>
</tr>
<tr>
<td>93</td>
<td>U</td>
<td>112</td>
</tr>
<tr>
<td>94</td>
<td>V</td>
<td>1,2,+</td>
</tr>
<tr>
<td>95</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>Y</td>
<td>5,7+</td>
</tr>
</tbody>
</table>
4.5 Conclusion

Tests of the algorithm have demonstrated its capability for analyzing most simple Hebrew sentences. Present limitations on the depth of its predictive logic have prevented the successful analysis of more complex sentences. However, the general requirements for extending its capability are known, and there is nothing to prevent the incorporation of these refinements in future versions.

Two additional aspects of the algorithm need further attention. First, there are several computations of the algorithm that are dependent upon the "content" of the grammar upon which it operates. If the algorithm were modified to free these computations from such dependence, a modification which is feasible, the algorithm could be used to generate sentences in other Semitic languages, such as Arabic, whenever grammars of these languages become available.

The second aspect of the algorithm needing further attention is the format of the input data prepared by the user. The sentence description input to this algorithm is much less complex than that of the synthesis algorithm. However, it still requires a complete grammatical description of each word in the sentence. This requirement could be eliminated by the incorporation of an automatic word analysis algorithm. Such an algorithm exists for Hebrew but it has not been made a part of this present program.
Appendix

PART IV

APPENDIX A
LISTING OF THE
GRAMMAR OF
HEBREW SYNTAX FOR
ANALYSIS OF SENTENCES
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF, BCLYDNGPRAVIT/S/- /X/Z) RULE NO.

Z-(10 001000000R0000 00000 0 0) + ( 55. 1)
NOP(11 99909000000000 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
VMI(10 003000000R4000 00000 0 0)

Z-(11 99909000000000 00000 0 0) + ( 52. 1)
NOP(11 99909000000000 00000 0 0) +
NP(11 99109999900000 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
VMD(10 00300000008000 00000 0 0)

Z-(11 99909000000000 00000 0 0) + ( 51. 1)
NOP(91 99909000000000 00000 0 0) +
DP(94 99909000000000 00000 0 0) +
KD(11 99909000000000 00000 0 0) =
VMC(10 00300000007000 00000 0 0)

Z-(11 99909000000000 00000 0 0) + ( 49. 1)
NOP(11 99909000000000 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
VMA(10 00300000003000 00000 0 0)

Z-(11 9910900000R0000 00000 0 0) + ( 54. 1)
DP(94 99909000000000 00000 0 0) =
VMR(10 004000000R5000 00000 0 0)

Z-(10 0010Y0000R0000 00000 0 0) = ( 41. 1)
XP(10 0020Y0000R0000 00000 0 0)
### Listing of Modern Hebrew Syntax Analysis Rules 7/21/70—James D. Price

**Symbol (MF/KBCLYDNGPRAVIT/S/-W/-X/Z)**

<table>
<thead>
<tr>
<th>Rule No.</th>
<th>Rule Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1</td>
<td>( VQO(11 \ 9910Y0NGPR999 \ 00000 \ 0 \ 0) = S00(10 \ 0010Y0NGPR0000 \ 00000 \ 0 \ 0) )</td>
</tr>
<tr>
<td>7.1</td>
<td>( APA(10 \ 0090YDNG0000000 \ 00000 \ 210) = AP(10 \ 0010YDNG000000 \ 00000 \ 210) )</td>
</tr>
<tr>
<td>7.2</td>
<td>( APA(10 \ 0090YD3G0000000 \ 00000 \ 0 \ 0) = AP(10 \ 0010YD2G000000 \ 00000 \ 0 \ 0) )</td>
</tr>
<tr>
<td>48.1</td>
<td>( AP(11 \ 9910Y0NG0000000 \ 00000 \ 0 \ 0) = NPX(10 \ 0010Y9NG900000 \ 00000 \ 0 \ 0) )</td>
</tr>
<tr>
<td>41.1</td>
<td>( NA(10 \ 0091Y9999000000 \ 00000 \ 0 \ 0) + U-(10 \ 00700000000000 \ 00000 \ 0 \ 0) = XP(10 \ 0010Y0000100000 \ 00000 \ 0 \ 0) )</td>
</tr>
<tr>
<td>25.1</td>
<td>( NA(12 \ 9990YDNGP000000 \ 00000 \ 0 \ 0) + AP(91 \ 99909DNG000000 \ 00000 \ 0 \ 0) + RD(90 \ 00900DNGP000000 \ 00000 \ 0 \ 0) + AS(91 \ 9990000000000 \ 00000 \ 0 \ 0) = NPB(10 \ 0010YDNGP000000 \ 00000 \ 0 \ 0) )</td>
</tr>
<tr>
<td>25.2</td>
<td>( NS(10 \ 0010Y2NGP000000 \ 00000 \ 0 \ 0) + AP(91 \ 999092NG000000 \ 00000 \ 0 \ 0) = NPB(10 \ 0020Y2NGP000000 \ 00000 \ 0 \ 0) )</td>
</tr>
<tr>
<td>52.1</td>
<td>( RO(11 \ 991092999900000 \ 00000 \ 0 \ 0) + Z-(91 \ 999090000900000 \ 00000 \ 0 \ 0) + NP(11 \ 991099999900000 \ 00000 \ 0 \ 0) + DP(94 \ 99909000000000 \ 00000 \ 0 \ 0) = VMD(10 \ 00200000008000 \ 00000 \ 0 \ 0) )</td>
</tr>
<tr>
<td>SYMBOL (MF/KBCLYDNGPRAVIT/S/-W/-X/Z)</td>
<td>RULE NO.</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>RO(19 99109299900000 00000 0 0) +</td>
<td>(51.1)</td>
</tr>
<tr>
<td>Z-(91 99909000090000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>DP(94 99909000000000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>KD(11 99909000000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>VMC(10 00200000000700 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>RO(11 99109299900000 00000 0 0) +</td>
<td>(55.1)</td>
</tr>
<tr>
<td>Z-(10 00100000000000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>DP(94 99909000000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>VMI(10 00200000000400 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>RO(11 99109299900000 00000 0 0) +</td>
<td>(50.1)</td>
</tr>
<tr>
<td>NIP(11 99909000090000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>DP(94 99909000000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>VMB(10 00200000000400 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>RO(10 00109299900000 00000 0 0) +</td>
<td>(49.1)</td>
</tr>
<tr>
<td>Z-(91 99909000090000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>DP(94 99909000000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>VMA(10 00200000000300 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>BAA(10 00900060000000 00000 0 0) =</td>
<td>(20.1)</td>
</tr>
<tr>
<td>BBA(10 00100060000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>BAB(10 00100036000000 00000 0 0) =</td>
<td>(20.2)</td>
</tr>
<tr>
<td>BBA(10 00200036000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>BAC(10 00900036000000 00000 0 0) =</td>
<td>(20.3)</td>
</tr>
<tr>
<td>BBA(10 00300036000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>Symbol (MF/KBLYDNGPRAVIT/S/-W/-X/Z)</td>
<td>Rule No.</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>BAD(10 00900036000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>BBA(10 00400036000000 00000 0 0)</td>
<td>(20.4)</td>
</tr>
<tr>
<td>BAD(10 00900036000000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>C-(10 0010000000000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>BAA(10 00900096000000 99999 0 0) =</td>
<td></td>
</tr>
<tr>
<td>BBA(10 00500036000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>BAE(10 00900036000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>BBB(10 00100036000000 00000 0 0)</td>
<td>(21.1)</td>
</tr>
<tr>
<td>BAE(10 00900036000000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>C-(10 0010000000000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>BBA(10 00900096000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>BBB(10 00200036000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>BAF(10 00900036000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>BBC(10 00100036000000 00000 0 0)</td>
<td>(23.1)</td>
</tr>
<tr>
<td>BAF(10 00900036000000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>C-(10 0010000000000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>BA(10 00900036000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>BBC(10 00200036000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>BBA(10 00900096000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>BA(10 00100036000000 00000 0 0)</td>
<td>(22.1)</td>
</tr>
</tbody>
</table>
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES D. PRICE

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z) RULE NO.

BBB(10 00900036000000 00000 0 0) = (22. 2)
BA(10 00200036000000 00000 0 0)

BA(10 009000NG000000 00000 0 0) = (24. 1)
BP(10 001000NG000000 00000 0 0)

BBC(10 00900036000000 00000 0 0) = (24. 2)
BP(10 00200036000000 00000 0 0)

BP(10 009000NG000000 00000 0 0) +
J-(10 001000NGP00000 99999 0 0) +
J-(94 99100099900000 99999 0 0) +
NPA(10 002000DN6P00000 00000 0 0)

BP(10 005000NG000000 00000 0 0) + (26. 1)
NPA(10 001000NGP00000 00000 0 0) +
D-(94 99200000000000 99999 0 0) =
NPA(10 001000DN6P00000 00000 0 0)

BP(10 00900099000000 00000 0 0) + (26. 2)
NPA(10 009000DN6P00000 00000 0 0) +
D-(94 99200000000000 99999 0 0) =
NPB(10 001000DN6P00000 00000 0 0)

BP(10 00900099000000 00000 0 0) + (30. 1)
D-(10 00319000000000 99999 0 0) =
DPA(10 00200000000000 00000 0 0)

BP(10 009000NG000000 00000 0 0) = (40. 1)
BC(10 001000DN6G00000 00000 4 9)
SYMBOL (MF/KBCLYDNGPRAVIT/S/-W/-X/Z)                   RULE NO.

NPB(10 00900DNP00000 00000 0 0) + D-(94 9920000000000 99999 0 0) =
NPA(10 00100DNP00000 00000 0 0)                      (26.1)

NPA(10 00900DNP00000 00000 0 0) + NAP(90 009002NGP00000 00000 0 0) =
NP(10 00100DNP00000 00000 0 0)                        (29.1)

NPA(12 99919DNP00000 00000 0 0) =
NPC(10 00100DNP00000 00000 0 0)                      (27.1)

NP(10 001009NGP00000 00000 0 0) + L-(10 0012000000000 00000 0 0) +
R-(10 003000NGP00000 00000 0 0) +
VP(11 99900NGP99113 00000 0 0) =
SAB(10 006010NGP00113 00000 0 0)                      (65.1)

NP(10 001009NGP00000 00000 0 0) +
U-(10 0030000000000 00000 0 0) +
R-(10 003000NGP00000 00000 0 0) +
VP(11 99900NGP99113 00000 0 0) =
SAB(10 006000NGP00113 00000 0 0)                      (65.2)

NP(12 99100DNP00000 00000 0 0) =
NSP(10 00100DNP00000 00000 0 0)                       (45.1)

NP(11 9910YD0000000 00000 2 9) =
NOP(10 0010YD00000000 00000 2 9)                      (46.1)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL (MF/KBC/LYDNGPRAVIT/S/-W-/X/Z) RULE NO.

NP(12 9910YDNGP00000 00000 0 0) = (48. 1)

NPX(10 0030YDNGP00000 00000 0 0)

DPA(11 9911900000000 00000 0 0) = (33. 1)

DP(10 0030000000000 00000 0 0)

DPB(11 9911900000000 00000 0 0) = (33. 2)

DP(10 0040000000000 00000 0 0)

DPB(14 9910Y000000000 00000 0 0) = (48. 1)

NPX(10 0020Y999900000 00000 0 0)

DPC(11 9911900000000 00000 0 0) = (33. 1)

DP(10 0050000000000 00000 0 0)

DP(14 9990900000000 00000 0 0) +

RO(19 9910Y929990000 00000 0 0) +

Z(91 9990900009000 00000 0 0) +

DP(94 9990900000000 00000 0 0) +

KD(11 9990900000000 00000 0 0) =

VMC(10 00200000000700 00000 0 0)

DP(14 9990900000000 00000 0 0) +

Z(91 9990900009000 00000 0 0) +

NOP(91 9990990000000 00000 0 0) +

DP(94 9990900000000 00000 0 0) +

KD(11 9990900000000 00000 0 0) =

VMC(10 0030000000700 00000 0 0)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL (MF/KBC/P/YDNG/PRAVIT/S/W/-/X/Z)  RULE NO.

DP(14 99909000000000 00000 0 0) +
R0(11 99109299900000 00000 0 0) +
Z-(91 99909000090000 00000 0 0) +
NP(11 99109999999999 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
VMD(10 00200000008000 00000 0 0)

DP(14 99909000000000 00000 0 0) +
Z-(91 99909000090000 00000 0 0) +
NOP(11 99909999999999 00000 0 0) +
NP(11 99109999999999 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
VMD(10 00300000008000 00000 0 0)

DP(14 99909000000000 00000 0 0) +
R0(11 00109299900000 00000 0 0) +
Z-(91 99909000090000 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
VMA(10 00200000030000 00000 0 0)

DP(14 99909000000000 00000 0 0) +
Z-(91 99909000090000 00000 0 0) +
NOP(11 99909999999999 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
VMA(10 00300000030000 00000 0 0)

DP(14 99909000000000 00000 0 0) +
R0(11 99109299900000 00000 0 0) +
NIP(11 99909000090000 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
VMB(10 002000000R40000 00000 0 0)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>RULE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP(11) 99990000000000 00000 0 0</td>
<td>(50. 2)</td>
</tr>
<tr>
<td>NIP(11) 9999000000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(94) 99990000000000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>VMB(10) 003000000R4000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(14) 99990000000000 00000 0 0</td>
<td>(53. 1)</td>
</tr>
<tr>
<td>P-(10) 00100000010000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>NV(10) 00900000099000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(94) 99990000000000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>VM(10) 00600000060000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(14) 99990000000000 00000 0 0</td>
<td>(55. 1)</td>
</tr>
<tr>
<td>RO(11) 99109299900000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>Z-(10) 001000000R0000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(94) 99990000000000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>VMI(10) 002000000R4000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(14) 99990000000000 00000 0 0</td>
<td>(55. 2)</td>
</tr>
<tr>
<td>Z-(10) 001000000R0000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>NOP(11) 99999000000000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(94) 99990000000000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>VMI(10) 003000000R4000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(14) 99990000000000 00000 0 0</td>
<td>(56. 1)</td>
</tr>
<tr>
<td>VA(10) 0010YONGP01VIT 99999 0 0</td>
<td></td>
</tr>
<tr>
<td>VM(12) 99990000000000 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>VP(10) 0010YONGP01VIT 00000 0 0</td>
<td></td>
</tr>
<tr>
<td>DP(14) 99990000000000 00000 0 0</td>
<td>(56. 2)</td>
</tr>
<tr>
<td>VA(10) 0010YONGPRAVIT 99999 14</td>
<td></td>
</tr>
<tr>
<td>VM(11) 999900099RA000 00000 14</td>
<td></td>
</tr>
<tr>
<td>VP(10) 0010YONGPRAVIT 00000 14</td>
<td></td>
</tr>
</tbody>
</table>

4-A-9
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W/-X/Z)

<table>
<thead>
<tr>
<th>RULE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(54. 1)</td>
</tr>
<tr>
<td>(53. 1)</td>
</tr>
<tr>
<td>(53. 2)</td>
</tr>
<tr>
<td>(51. 1)</td>
</tr>
<tr>
<td>(54. 1)</td>
</tr>
<tr>
<td>(53. 1)</td>
</tr>
<tr>
<td>(62. 1)</td>
</tr>
</tbody>
</table>

120

4-A-10
<table>
<thead>
<tr>
<th>SYMBOL (MF/KDCLKDYDNGPRAVIT/S/-W/-X/Z)</th>
<th>RULE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA(10 001000NGPR1100 SWWW 0 0) =</td>
<td>(36.1)</td>
</tr>
<tr>
<td>VBB(10 002000NGPR1113 SWWW 0 0)</td>
<td></td>
</tr>
<tr>
<td>EA(10 0010Y1NGPRAV00 SWWW 0 0) =</td>
<td>(36.2)</td>
</tr>
<tr>
<td>VBB(10 0010YONGPRAV13 SWWW 0 0)</td>
<td></td>
</tr>
<tr>
<td>VB(10 0010YONGPRAVIT SWWW 517) =</td>
<td>(36.3)</td>
</tr>
<tr>
<td>VBB(10 0010YONGPRAVIT SWWW 517)</td>
<td></td>
</tr>
<tr>
<td>VBB(10 0090YONGPRAVIT SWWW1117) +</td>
<td>(37.1)</td>
</tr>
<tr>
<td>W-(10 00100000NGPR00 SWWW 0 0) =</td>
<td></td>
</tr>
<tr>
<td>VC(10 0020YONGPRAVIT SWWW1117)</td>
<td></td>
</tr>
<tr>
<td>VBB(10 0090YONGPRAVIT SWWW1117) =</td>
<td>(37.2)</td>
</tr>
<tr>
<td>VC(10 0030YONGPRAVIT SWWW1117)</td>
<td></td>
</tr>
<tr>
<td>VC(10 0090YONGPRAVIT SWWW1117) =</td>
<td>(38.1)</td>
</tr>
<tr>
<td>VAA(10 0010YONGPRAVIT SWWW1117)</td>
<td></td>
</tr>
<tr>
<td>VC(10 0030YONGP01VI121HYH* 0 0) +</td>
<td>(38.2)</td>
</tr>
<tr>
<td>VC(10 009000NGPRAV13 SWWW 0 0) =</td>
<td></td>
</tr>
<tr>
<td>VAA(10 0010YONGPRAV15 SWWW 0 0)</td>
<td></td>
</tr>
<tr>
<td>VC(10 0030YONGP01VI112HYH* 0 0) +</td>
<td>(38.3)</td>
</tr>
<tr>
<td>VC(10 009000NGPRAV13 SWWW 0 0) =</td>
<td></td>
</tr>
<tr>
<td>VAA(10 0010YONGPRAV14 SWWW 0 0)</td>
<td></td>
</tr>
</tbody>
</table>
### Listing of Modern Hebrew Syntax Analysis Rules 7/21/70—James D. Price

<table>
<thead>
<tr>
<th>Symbol (MF/KBC/LY/NG/PRA/V/ST/-W-/X/-Z)</th>
<th>Rule No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAA(10 0010YONGPRAVIT SWWWw 210) = VA(10 0010YONGPRAVIT SWWWw 210)</td>
<td>(39.1)</td>
</tr>
<tr>
<td>VAA(10 0010Y03GPR4VIT SWWWw 0 0) = VA(10 0010Y02GPR4VIT SWWWw 0 0)</td>
<td>(39.2)</td>
</tr>
<tr>
<td>V(10 0010YONGP01VIT 99999 0 0) + XP(11 99900000010000 00000 0 0) + NSP(12 999099NP00000 00000 0 0) + DP(94 99900000000000 00000 0 0) = SAA(10 0030YONGP00VIT 00000 317)</td>
<td>(64.1)</td>
</tr>
<tr>
<td>V. (10 0010YONGPRAVIT 99999 0 0) + VMR(11 999090999RA000 00000 0 0) = VRB(10 0010YONGPRAVIT 00000 0 0)</td>
<td>(57.1)</td>
</tr>
<tr>
<td>VA(10 0010YONGPRAVIT 99999 0 0) + VMI(11 999090000R4000 00000 0 0) = VR(10 0010YONGPRAVIT 00000 0 0)</td>
<td>(58.1)</td>
</tr>
<tr>
<td>VA(10 0010YONGPRAVIT 99999 114) + VM(11 999000999RA000 00000 114) = VP(10 0010YONGPRAVIT 00000 114)</td>
<td>(56.1)</td>
</tr>
<tr>
<td>VA(10 0010YONGP01VIT 99999 0 0) + VM(12 999000NGP01000 00000 0 0) = VP(10 0010YONGP01VIT 00000 0 0)</td>
<td>(56.2)</td>
</tr>
<tr>
<td>SYM/RUL (MF/KLCVYDNGPRAVIT/S/-W/-X/2)</td>
<td>RULE NO.</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>BC(10 00900DNG00000 00000 0 0) =</td>
<td>(26.1)</td>
</tr>
<tr>
<td>NPA(10 00300DNG900000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>XP(11 99900000010000 00000 0 0) +</td>
<td>(64.1)</td>
</tr>
<tr>
<td>VA(10 010YONGP01VIT 99999 0 0) +</td>
<td></td>
</tr>
<tr>
<td>NSP(12 99909NGP00000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>JP94 99909000000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>SAA(10 0010YONGP0UVIT 00000 317)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>XP(11 99900000010000 00000 0 0) +</td>
<td>(64.2)</td>
</tr>
<tr>
<td>NSP(12 99909NGP00000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>JP94 99909000000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>SAA(10 04000NGP00113 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>XP(11 999090000R0000 00000 0 0) +</td>
<td>(53.1)</td>
</tr>
<tr>
<td>DP(94 99909000000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>VM(10 00500000GR5000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>XP(11 9990Y0000090000 00000 0 0) =</td>
<td>(72.1)</td>
</tr>
<tr>
<td>KC(10 0040Y00000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>XP(11 9990Y0000090000 00000 0 0) =</td>
<td>(48.1)</td>
</tr>
<tr>
<td>NPX(10 0050Y9999000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>XP(11 9990Y0000090000 00000 0 0) =</td>
<td>(47.1)</td>
</tr>
<tr>
<td>NIP(10 0010Y00000R0000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td>SYMBOL(MF/KBCLYDNGPRAVIT/S/-W/-X/Z)</td>
<td>RULE NO.</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>XP(14 9990000090000 00000 0 0) =</td>
<td>(28.1)</td>
</tr>
<tr>
<td>NAP(10 00200999900000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>XP(11 9990Y000090000 00000 0 0) =</td>
<td>(33.1)</td>
</tr>
<tr>
<td>DP(10 00700000000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>NO(11 9990Y200000000 00000 0 0) =</td>
<td>(46.1)</td>
</tr>
<tr>
<td>NOP(10 0020Y200000000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RSP(12 99902NGP00000 00000 0 0) =</td>
<td>(45.1)</td>
</tr>
<tr>
<td>NSP(10 00200DNGP00000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>RSP(12 9990Y2NGP00000 00000 0 0) =</td>
<td>(48.1)</td>
</tr>
<tr>
<td>NPX(10 0040YDNGP00000 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>NSP(12 999099NGP00000 00000 0 0) +</td>
<td>(64.1)</td>
</tr>
<tr>
<td>VA(10 0010Y0NGP01VIT 99999 0 0) +</td>
<td></td>
</tr>
<tr>
<td>XP(11 99900000010000 00000 0 0) +</td>
<td></td>
</tr>
<tr>
<td>DP(94 99909000000000 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>SAA(10 0020Y0NGP00VIT 00000 317)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>NSP(12 999099NGP00000 00000 0 0) +</td>
<td>(65.1)</td>
</tr>
<tr>
<td>VP(11 9990Y0NGP99VIT 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>SAB(10 0010Y0NGP00VIT 00000 0 0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>NSP(12 999099NGP00000 00000 0 0) +</td>
<td>(68.1)</td>
</tr>
<tr>
<td>VRB(11 9990Y0NGP99999 00000 0 0) =</td>
<td></td>
</tr>
<tr>
<td>SRO(10 0020Y0NGP00000 00000 0 0)</td>
<td></td>
</tr>
</tbody>
</table>
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/KBCLYUNGPRAVIT/S/-w/-X/Z)

RULE NO.

NSP(12 999099NGP0000 00000 0 0) +
VRI(11 9990YONGPR9999 00000 0 0) =
SRI(10 0020YONGPR0000 00000 0 0)

(69. 1)

(10. 1)

(65. 1)

(52. 1)

(51. 1)

(50. 1)

(49. 1)

NSP(10 009009NGP0000 00000 0 0) +
VP(11 999000NGP01113 00000 0 0) =
SAB(10 002000NGP00113 00000 0 0)

NSP(11 999099000000000 00000 0 0) +
NP(11 991099999900000 00000 0 0) +
DP(94 999090000000000 00000 0 0) =
VMD(10 003000000008000 00000 0 0)

NOP(11 999099000000000 00000 0 0) +
DP(94 999090000000000 00000 0 0) +
KD(11 999090000000000 00000 0 0) =
VMD(10 003000000007000 00000 0 0)

NOP(11 999099000000000 00000 0 0) +
NIP(11 999090000000000 00000 0 0) +
DP(94 999090000000000 00000 0 0) =
VMB(10 003000000004000 00000 0 0)

NOP(11 999099000000000 00000 0 0) +
DP(94 999090000000000 00000 0 0) =
VMA(10 003000000003000 00000 0 0)

4-A-15
**LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES D. PRICE**

<table>
<thead>
<tr>
<th>SYMBOL (MF/KBCLYNDGPRAVIT/S/-W/-X/Z)</th>
<th>RULE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{NPX}(11 \ 999099\text{NP}00000 \ 00000 \ 0 \ 0) +)</td>
<td>(53.1)</td>
</tr>
<tr>
<td>(\text{DP}(94 \ 99909000000000 \ 00000 \ 0 \ 0) =)</td>
<td></td>
</tr>
<tr>
<td>(\text{VM}(10 \ 001000\text{NP}01000 \ 00000 \ 0 \ 0))</td>
<td></td>
</tr>
<tr>
<td>(\text{VMA}(10 \ 00900000003000 \ 00000 \ 0 \ 0) =)</td>
<td>(53.2)</td>
</tr>
<tr>
<td>(\text{VM}(10 \ 00300000003000 \ 00000 \ 0 \ 0))</td>
<td></td>
</tr>
<tr>
<td>(\text{VMA}(10 \ 00900000003000 \ 00000 \ 0 \ 0) =)</td>
<td>(54.1)</td>
</tr>
<tr>
<td>(\text{VMR}(10 \ 00200000003000 \ 00000 \ 0 \ 0))</td>
<td></td>
</tr>
<tr>
<td>(\text{VMB}(10 \ 00900000004000 \ 00000 \ 0 \ 0) =)</td>
<td>(53.1)</td>
</tr>
<tr>
<td>(\text{VM}(10 \ 00400000004000 \ 00000 \ 0 \ 0))</td>
<td></td>
</tr>
<tr>
<td>(\text{VMB}(10 \ 00900000004000 \ 00000 \ 0 \ 0) =)</td>
<td>(54.1)</td>
</tr>
<tr>
<td>(\text{VMR}(10 \ 00300000004000 \ 00000 \ 0 \ 0))</td>
<td></td>
</tr>
<tr>
<td>(\text{VMC}(10 \ 00900000007000 \ 00000 \ 0 \ 0) =)</td>
<td>(53.1)</td>
</tr>
<tr>
<td>(\text{VM}(10 \ 00700000007000 \ 00000 \ 0 \ 0))</td>
<td></td>
</tr>
<tr>
<td>(\text{VMD}(10 \ 00900000008000 \ 00000 \ 0 \ 0) =)</td>
<td>(53.2)</td>
</tr>
<tr>
<td>(\text{VM}(10 \ 00800000008000 \ 00000 \ 0 \ 0))</td>
<td></td>
</tr>
<tr>
<td>(\text{VMD}(10 \ 00900000008000 \ 00000 \ 0 \ 0) =)</td>
<td>(54.1)</td>
</tr>
<tr>
<td>(\text{VMR}(10 \ 00500000008000 \ 00000 \ 0 \ 0))</td>
<td></td>
</tr>
</tbody>
</table>
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/KB/CL/YD/GP/PR/VIT/S/-W-/X/X/Z)

RULE NO.

VP(10 0010Y0NGP99VIT 00000 317) =
SAC(10 0010Y0NGP00VIT 00000 317)

VP(10 0010Y0NGPR3VIT 00000 0 0) =
VQO(10 0010Y0NGPR8VIT 00000 0 0)

VP(10 0010Y0NGPR2VIT 00000 0 0) =
VQO(10 0010Y0NGPR3VIT 00000 0 0)

VP(10 0010Y0NGPR5VIT 00000 0 0) =
VQO(10 0010Y0NGPR4VIT 00000 0 0)

VRD(11 9910Y0NGP9999 00000 0 0) =
SR0(10 0010Y0NGP00000 00000 0 0)

VKI(11 9910Y0NGPR9999 00000 0 0) =
SR0(10 0010Y0NGPR00000 00000 0 0)

NV(11 9990Y000099000 00000 0 0) =
NIP(10 0020Y0000R0000 00000 0 0)

NV(10 0090000099000 00000 0 0) =
NSP(10 004000D11300000 00000 0 0)

EPB(10 0090Y0NGPR3100 00000 0 0) =
EPA(10 0020Y0NGPR3100 00000 0 0)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/"BCLYDNGPRAVIT/S/-W/-X/Z)

RULE NO.

EPA(10 0010YD3GP99900 00000 0 0) =
EP(10 0010YD2GP00000 00000 0 0) (63. 1)

EPA(10 0090YDNGP99900 00000 210) =
EP(10 0010YDNGP00000 00000 210) (63. 2)

EP(10 0010YDNGP00000 00000 0 0) =
NPB(10 0030YDNGP00000 00000 0 0) (25. 1)

SAA(10 0090YONGP00000 00000 0 0) =
SA(10 0010YONGP00000 00000 0 0) (67. 1)

SAB(10 0090YONGP00000 00000 0 0) =
SA(10 0030YONGP00000 00000 0 0) (67. 2)

SAC(10 0010YONGP00000 00000 0 0) =
SA(10 0020YONGP00000 00000 0 0) (67. 3)

SA(11 9990Y9990091T 00000 0 0) +
T-(90 0030000100000 00000 0 0) +
KC(11 9990999990000 00000 0 0) =
SD(10 0020Y0000001T 00000 0 0) (76. 1)

SA(11 9990Y9990091T 00000 0 0) =
S-(10 0010Y000000001T 00000 0 0) (77. 1)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES D. PRICE

SYMBOL (MF/KBCLYDNGPRAVIT/S/-N-/X/Z)       RULE NO.

RG(11 9990YDNGP00000 00000 0 0) =
NAP(10 0030DNGP00000 00000 0 0) = (28.1)

KN(10 0090Y000000000 00000 0 0) =
KD(10 0010Y000000000 00000 0 0) = (75.1)

KN(10 0090000000000 00000 0 0) =
NSP(10 0030211300000 00000 0 0) = (45.1)

KC(11 9990Y000000000 00000 0 0) +
T-(90 0030000000000 00000 0 0) +
SA(11 9990Y0999009IT 00000 0 0) =
SU(10 0010Y00000001T 00000 0 0) = (76.1)

KC(11 9990Y000000000 00000 0 0) +
T-(10 0030000000000 00000 0 0) +
KI(10 0090Y099990000 00000 0 0) =
SI(10 0020Y000000000 00000 0 0) = (78.1)

KC(11 9990Y000000000 00000 0 0) =
NPX(10 0060YDNGP00000 00000 0 0) = (48.1)

KK(11 9910Y000000001 00000 0 0) +
T-(90 0030000000000 00000 0 0) +
SA(11 9990Y0999009II 00000 0 0) =
S-(10 0030Y00000001I 00000 0 0) = (77.1)

4-A-19
RULE NO.

(11 99109000000002 00000 0 0) + (10 00300000000000 00000 0 0) + (11 9990Y09990U912 00000 0 0) = (10 0030Y00000012 00000 0 0)

(10 0090Y099990000 00000 0 0) + (10 00300000000000 00000 0 0) + (11 99909000000000 00000 0 0) = (10 0030Y00000000 00000 0 0)

(10 0090Y099990000 00000 0 0) = (10 0010Y00000000 00000 0 0)

(10 00900000000000 00000 0 0) + (10 00700000000000 00000 0 0) + (10 0030Y00000000 00000 0 0) =

(10 0090Y0000000022 00000 0 0) + (10 00600000000000 00000 0 0) + (10 0030Y00000000 00000 0 0)

(10 0090Y0000000099 00000 0 0) + (10 00600000000000 00000 0 0) + (10 0010Y00000000 00000 0 0)

(11 9990Y000000000 00000 0 0) + (10 00500000000000 00000 0 0) + (10 0020Y000000000 00000 0 0) =

4-A-20

130
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES U. PRICE

SYMBOL(MF/KHCLYDNGPRAVIT/S/-w-/X2)

RULE NO.

A-(10 0011YUNG000000 99999 0 0) +
XP(11 99900000090000 00000 013) =
APA(10 0020Y9NG000000 00000 0 0)

A-(10 0011Y2NG000000 99999 0 0) +
XP(11 99900000020000 00000 0 0) =
APA(10 0030Y2NG000000 00000 0 0)

A-(10 0011YDNG000000 99999 0 0) +
B-(94 90600000000000 99999 0 0) =
APA(10 0020YDNG000000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)

B-(10 0010001600000 99999 0 0) +
B-(10 0040001600000 00000 0 0) =
BAC(10 0010003600000 00000 0 0)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/KH/CLY/UNG/PRAVIT/S/-w/-X/Z)

<table>
<thead>
<tr>
<th>Rule No.</th>
<th>Symbol</th>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.2</td>
<td>B-(10 00300012000000 99999 0 0) + B-(10 00400011000000 00000 0 0) = BAC(10 00300031000000 00000 0 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.1</td>
<td>B-(10 00300012000000 99999 0 0) = BAA(10 00300031000000 00000 0 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.2</td>
<td>B-(10 00300011000000 99999 0 0) = BAA(10 00300032000000 00000 0 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.1</td>
<td>B-(10 00300031000000 99999 0 0) = BAD(10 00200039000000 00000 0 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.2</td>
<td>B-(10 00400031000000 00000 0 0) = BAD(10 00100039000000 00000 0 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.1</td>
<td>B-(10 00400012000000 00000 0 0) = BAB(10 00100031000000 00000 0 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.2</td>
<td>B-(10 00400011000000 00000 0 0) = BAB(10 00100032000000 00000 0 0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.1</td>
<td>B-(10 00500012000000 00000 0 0) = BAE(10 00100039000000 00000 0 0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z) RULE NO.

\[ B-(10 ~ 00500022000000 ~ 0000000000) = \]
\[ BAE(10 ~ 00200039000000 ~ 0000000000) \]
\( (18.2) \)

\[ B-(10 ~ 00600011000000 ~ 0000000000) = \]
\[ BAF(10 ~ 00100039000000 ~ 0000000000) \]
\( (19.1) \)

\[ B-(10 ~ 00600021000000 ~ 0000000000) = \]
\[ BAF(10 ~ 00200039000000 ~ 0000000000) \]
\( (19.2) \)

\[ C-(10 ~ 00300000000000 ~ 0000000000) + \]
\[ SA(11 ~ 9990Y099900999 ~ 0000000000) = \]
\[ KN(10 ~ 0010Y0000000000 ~ 0000000000) \]
\( (71.1) \)

\[ C-(10 ~ 00400000000000 ~ 9999990000) + \]
\[ SA(11 ~ 9999Y099900999 ~ 0000000000) = \]
\[ KC(10 ~ 0010Y0000000000 ~ 0000000000) \]
\( (72.1) \)

\[ C-(10 ~ 00500000000000 ~ 9999990000) + \]
\[ SA(11 ~ 9999Y099900999 ~ 0000000817) = \]
\[ KC(10 ~ 0020Y0000000000 ~ 0000000000) \]
\( (72.2) \)

\[ C-(10 ~ 00600000000000 ~ 9999990000) + \]
\[ SA(11 ~ 9999Y099900999 ~ 0000000000) = \]
\[ KC(10 ~ 0030Y0000000000 ~ 0000000000) \]
\( (72.3) \)

\[ C-(10 ~ 00800000000000 ~ 9999990000) + \]
\[ S-(10 ~ 0090Y0000000099 ~ 0000000000) + \]
\[ T-(10 ~ 00600000000000 ~ 0000000000) = \]
\[ SC(10 ~ 0010Y0000000000 ~ 0000000000) \]
\( (79.1) \)

4-A-23
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES D. PRICE

SYMBOL(MF/KBCLDJGPRAVIT/S/-W/-X/Z) RULE NO.

C-(10 00600000000000 99999 0 0) + (79. 2)
S-(11 9990Y00000000 00000 0 0) +
T-(10 00500000000000 00000 0 0) =
SC(10 0020Y00000000 00000 0 0)

C-(10 00600000000000 99999 0 0) + (79. 3)
S-(10 0090Y00000022 00000 0 0) +
T-(10 00700000000000 00000 0 0) =
SC(10 0030Y00000000 00000 0 0)

D-(11 99119000000000 99999 0 0) = (33. 1)
DP(10 00100000000000 00000 0 0)

L-(11 9911Y00000000 99999 0 0) = (43. 1)
DPD(10 0010Y00000000 00000 0 0)

D-(11 99219000000000 99999 0 0) = (33. 1)
DP(10 00200000000000 00000 0 0)

D-(11 9921Y00000000 99999 0 0) = (43. 1)
DPD(10 0020Y00000000 00000 0 0)

D-(10 00319000000000 99999 0 0) + (30. 1)
BP(10 00900099000000 00000 0 0) =
DPA(10 00100000000000 00000 0 0)

D-(10 00419000000000 99999 0 0) + (31. 1)
D-(90 00600000000000 99999 0 0) =
DPB(10 00100000000000 00000 0 0)

4-A-24
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/'X/'Z) RULE NO.

\[ D-(10 005190000000000 99999 0 0) + \]
\[ D-(90 006000000000000 99999 0 0) = \]
\[ LPC(10 001000000000000 00000 0 0) \]

\[ D-(10 006190000000000 99999 0 0) + \]
\[ D-(90 006000000000000 99999 0 0) = \]
\[ UP(10 006000000000000 00000 0 0) \]

\[ D-(10 0071Y000000000 00000 0 0) + \]
\[ VC(10 009000NGPRAVI 1 SWWWw 0 0) = \]
\[ VAA(10 0010YONGPRAVI6 SWWWw 0 0) \]

\[ D-(10 0081Y000000000 00000 0 0) + \]
\[ VC(10 009000NGPRAVI2 SWWWw 0 0) = \]
\[ VAA(10 0010YONGPRAVI7 SWWWw 0 0) \]

\[ E-(10 0011YONGPRAV00 SWWWw 2 9) = \]
\[ EA(10 0010Y9NGPRAV00 SWWWw 2 9) \]

\[ G-(11 9912YONGP03100 99999 0 0) + \]
\[ NPA(11 99909D999000000 00000 0 0) = \]
\[ EPB(10 0010YDNGPR3100 00000 0 0) \]

\[ G-(10 0012YONGP03100 99999 0 0) + \]
\[ R-(10 00300099900000 00000 0 0) = \]
\[ EPB(10 0020Y2NGPR3100 00000 0 0) \]
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70 -- JAMES D. PRICE

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)  RULE NO.

H-(10 011Y00000000 00000 0 0) +                     ( 9. 1)
N-(10 01000NGP00000 99999 0 0) =
NA(10 020Y2NGP00000 00000 0 0)

H-(10 011Y00000000 00000 0 0) +                     ( 12. 1)
R-(10 001000NG30000 00000 0 0) =
RD(10 0020Y2NG30000 00000 0 0)

H-(10 011Y00000000 00000 0 0) +                     ( 12. 2)
R-(10 002000NG30000 00000 0 0) =
RD(10 0030Y2NG30000 00000 0 0)

H-(10 011Y00000000 00000 0 0) +                     ( 34. 1)
E-(10 001000NGPRAV00 SWWWW 0 0) =
EA(10 0010Y2NGPRAV00 SWWWW 0 0)

H-(10 001000000000 00000 0 0) +                     ( 40. 1)
BP(10 009000NG00000 00000 0 0) =
BC(10 002002NG00000 00000 0 0)

H-(10 001000000000 00000 0 0) +                     ( 80. 1)
A-(10 001000NG00000 SWWWW 0 0) =
A-(10 001002NG00000 SWWWW 0 0)

I-(10 0020003600000 00000 0 0) +                     ( 16. 1)
B-(10 0040001600000 00000 0 0) =
BAC(10 0020003600000 00000 0 0)
<table>
<thead>
<tr>
<th>RULE NO.</th>
<th>SYMBOL(location)</th>
</tr>
</thead>
<tbody>
<tr>
<td>18. 1</td>
<td>I-(10 0020003200000 00000 0 0) +</td>
</tr>
<tr>
<td></td>
<td>B-(10 0050003200000 00000 0 0) =</td>
</tr>
<tr>
<td></td>
<td>BAE(10 0020003900000 00000 0 0)</td>
</tr>
<tr>
<td>19. 1</td>
<td>I-(10 0020003100000 00000 0 0) +</td>
</tr>
<tr>
<td></td>
<td>B-(10 0060003100000 00000 0 0) =</td>
</tr>
<tr>
<td></td>
<td>BAF(10 0020003900000 00000 0 0)</td>
</tr>
<tr>
<td>18. 1</td>
<td>I-(10 0030001100000 99999 0 0) +</td>
</tr>
<tr>
<td></td>
<td>B-(10 0050003200000 00000 0 0) =</td>
</tr>
<tr>
<td></td>
<td>BAE(10 0030003900000 00000 0 0)</td>
</tr>
<tr>
<td>19. 1</td>
<td>I-(10 0030003100000 99999 0 0) +</td>
</tr>
<tr>
<td></td>
<td>B-(10 0060003100000 00000 0 0) =</td>
</tr>
<tr>
<td></td>
<td>BAF(10 0030003900000 00000 0 0)</td>
</tr>
<tr>
<td>26. 1</td>
<td>J-(10 0010000000000 99999 0 0) +</td>
</tr>
<tr>
<td></td>
<td>J-(94 9910009990000 99999 0 0) +</td>
</tr>
<tr>
<td></td>
<td>NPB(10 0090000990000 00000 0 0) +</td>
</tr>
<tr>
<td></td>
<td>D-(94 9920000000000 99999 0 0) =</td>
</tr>
<tr>
<td></td>
<td>NPA(10 0020000990000 00000 0 0)</td>
</tr>
<tr>
<td>11. 1</td>
<td>J-(10 0011000000000 99999 0 0) +</td>
</tr>
<tr>
<td></td>
<td>R-(10 0030009990000 00000 0 0) =</td>
</tr>
<tr>
<td></td>
<td>NS(10 0010000990000 00000 0 0)</td>
</tr>
<tr>
<td>64. 1</td>
<td>L-(10 0012000000000 99999 0 0) +</td>
</tr>
<tr>
<td></td>
<td>XP(11 999000000001000 00000 0 0) +</td>
</tr>
<tr>
<td></td>
<td>NSP(12 9999990000000 00000 0 0) +</td>
</tr>
<tr>
<td></td>
<td>DP(94 9999099000000 00000 0 0) =</td>
</tr>
<tr>
<td></td>
<td>SAA(10 00401009900113 00000 0 0)</td>
</tr>
</tbody>
</table>
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES D. PRICE

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z) RULE NO.

L-(10 00:120000000000 00000 0 0) + ( 65. 1)
R-(10 003000NGP00000 00000 0 0) +
VP(11 999000NGP99113 00000 0 0) =
SAB(10 004010NGP00113 00000 0 0)

L-(10 00120000000000 00000 0 0) + ( 65. 2)
NP(10 001009NGP00000 00000 0 0) +
VP(11 999000NGP99113 00000 0 0) =
SAB(10 005010NGP00113 00000 0 0)

L-(10 001L0000000000 00000 0 0) + ( 2. 1)
F-(1F 00CL0DNGPRAVIT SWWW 0 0) =
F-(1F 00CL1DNGPRAVIT SWWW 0 0)

N-(10 0011Y0NGP00000 99999 0 0) + ( 9. 1)
B-(10 001000NGP00000 00000 0 0) =
NA(10 0010Y1NGP00000 00000 0 0)

N-(10 0011Y0NGP00000 99999 2 9) = ( 9. 2)
NA(10 0010Y9NGP00000 00000 2 9)

N-(14 9991Y0NGP00000 99999 1 6) = ( 9. 3)
NA(10 0030Y2NGP00000 00000 0 0)

O-(10 0011Y0000000000 00000 0 0) + ( 13. 1)
R-(10 003000NGP00000 00000 0 0) =
RO(10 0010Y2NGP00000 00000 0 0)
LISTING OF MODERN HEBREW SYM TAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOLS (MF/KBCLYNPGRAVIT/S/-W/-X/-Z)

RULE NO.

O (10 0011Y00000000 00000 0 0) +
HP (10 0010029990000 00000 0 0) =
NO (10 0010Y200000000 00000 0 0)

(42.1)

(42.2)

P (10 0010000000R000 00000 0 0) +
R (10 0050000000000 99999 0 0) +
SRI (10 0090YONGPR000 00000 0 0) =
KI (10 0040YONGPR000 00000 0 0)

(74.1)

(53.1)

(41.1)

(41.2)

(4.1)

P (10 0011Y0000R000 00000 0 0) +
R (10 0030009990000 00000 0 0) =
Z (10 0010Y0000R000 00000 0 0)

P (10 0011Y0000R000 00000 0 0) +
NP (11 9910099990000 00000 0 0) =
XP (10 0010Y0000R000 00000 0 0)

(42.2)

P (10 0011Y0000R000 00000 0 0) +
RG (11 9990099990000 00000 0 0) =
XP (10 0030Y0000R000 00000 0 0)

(42.2)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES D. PRICE

SYMBOL(MF/KBCLYDNPRAVIT/S/-W/-X/Z) RULE NO.

\[ \omega-(10 \ 00900000000000 \ 000000 \ 0 \ 0) + \]
\[ \text{SA}(11 \ 9990YONGP00999 \ 00000 \ 0 \ 0) = \]
\[ \text{KI}(10 \ 0010YONGP00000 \ 00000 \ 0 \ 0) \]

\[ \text{K}- (12 \ 991000NGP00000 \ 00000 \ 0 \ 0) = \]
\[ \text{RSP}(10 \ 002002JN3P00000 \ 00000 \ 0 \ 0) \]

\[ \text{R}-(10 \ 0011YONG300000 \ 00000 \ 0 \ 0) = \]
\[ \text{KD}(10 \ 0010YDNG300000 \ 00000 \ 0 \ 0) \]

\[ \text{R}-(10 \ 002000NGP00000 \ 00000 \ 0 \ 0) + \]
\[ \text{NPC}(93 \ 991002NGP00000 \ 00000 \ 0 \ 0) = \]
\[ \text{RSP}(10 \ 001002NGP00000 \ 00000 \ 0 \ 0) \]

\[ \text{R}-(10 \ 002000NG300000 \ 00000 \ 0 \ 0) = \]
\[ \text{VBB}(10 \ 003000NG3R1113 \ 5W0WW \ 0 \ 0) \]

\[ \text{R}-(10 \ 00300099900000 \ 00000 \ 0 \ 0) + \]
\[ \text{DP}(94 \ 99909000000000 \ 00000 \ 0 \ 0) + \]
\[ \text{Z}-(10 \ 001000000R00000 \ 00000 \ 0 \ 0) + \]
\[ \text{DP}(94 \ 99909000000000 \ 00000 \ 0 \ 0) = \]
\[ \text{VMI}(10 \ 001000000R4000 \ 00000 \ 0 \ 0) \]

\[ \text{R}-(10 \ 00300099900000 \ 00000 \ 0 \ 0) + \]
\[ \text{DP}(94 \ 99909000000000 \ 00000 \ 0 \ 0) + \]
\[ \text{Z}-(91 \ 999090000090000 \ 00000 \ 0 \ 0) + \]
\[ \text{NP}(11 \ 99109999900000 \ 00000 \ 0 \ 0) + \]
\[ \text{DP}(94 \ 99909000000000 \ 00000 \ 0 \ 0) = \]
\[ \text{VMD}(10 \ 0010000008000 \ 00000 \ 0 \ 0) \]

4-A-30

ERIC 140
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/KBCLYDNGPRAVIT/S/-w/-X/Z) RULE NO.

RULE NO.  

R-(10 00300099900000 00000 0 0) +  
Z-(91 99909000090000 00000 0 0) +  
UP(94 99909000090000 00000 0 0) +  
KD(11 99909000090000 00000 0 0) =  
VMC(10 0010000007000 00000 0 0)  

( 51. 1)  

RULE NO.  

R-(10 00300099900000 00000 0 0) +  
UP(94 99909000090000 00000 0 0) +  
NIP(11 99909000090000 00000 0 0) +  
DP(94 99909000090000 00000 0 0) =  
VMB(10 0010000004000 00000 0 0)  

( 50. 1)  

RULE NO.  

R-(10 00300099900000 00000 0 0) +  
UP(94 99909000090000 00000 0 0) +  
Z-(91 99909000090000 00000 0 0) +  
DP(94 99909000090000 00000 0 0) =  
VMA(10 0010000003000 00000 0 0)  

( 49. 1)  

RULE NO.  

R-(10 00411000000000 00000 0 0) +  
VP(11 999090NP00999 00000 0 0) =  
RG(10 0010YDNGP00000 00000 0 0)  

( 70. 1)  

RULE NO.  

R-(10 00411000000000 00000 0 0) +  
SR0(10 00909099900000 00000 0 0) =  
RG(10 0020Y999900000 00000 0 0)  

( 70. 2)  

RULE NO.  

R-(10 00411000000000 00000 0 0) +  
SRI(10 00909099900000 00000 0 0) =  
RG(10 0030Y999900000 00000 0 0)  

( 70. 3)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES D. PRICE

SYMBOL(MF/KBCLYMDNPRAVIT/S/-w/-X/Z) RULE NO.

K-(10 00400000000000 00000 0 0) + (71.1)
SA(11 9990Y09990999 00000 0 0) =
KN(10 0020Y00000000 00000 0 0)

K-(10 00500000000000 99999 0 0) + (74.1)
VP(11 9990Y0NGP09999 00000 0 0) =
KI(10 0020Y0NGP00000 00000 0 0)

R-(10 00500000000000 99999 0 0) + (74.2)
SGO(10 0090Y0NGPR00000 00000 0 0) =
KI(10 0030Y0NGPR00000 00000 0 0)

T-(10 00300000000000 00000 0 0) + (28.1)
NPC(13 99119NGP00000 00000 0 0) +
T-(10 00300000000000 00000 0 0) =
NAP(10 00100NGP00000 00000 0 0)

T-(10 00400000000000 00000 0 0) + (75.1)
T-(10 00100000000000 00000 0 0) +
SC(11 9990Y0000000000 00000 0 0) +
T-(10 00200000000000 00000 0 0) =
KD(10 0020Y0000000000 00000 0 0)

U-(10 00100000000000 00000 0 0) + (8.1)
R-(10 00300099900000 00000 0 0) =
AS(10 00100000000000 00000 0 0)

U-(10 00100000000000 00000 0 0) + (8.2)
NP(10 00100999900000 00000 0 0) =
AS(10 00200000000000 00000 0 0)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70—JAMES D. PRICE

SYMBOL(MF/KBCLY)INGPRAVIT/S/-W/-X/2) RULE NO.

U-(10 00300000000000 00000 0 0) + XP(11 99900000000000 00000 0 0) +
NSP(12 999099NGP00000 00000 0 0) +
DP(94 99909000000000 00000 0 0) =
SAA(10 004000NGP00113 00000 0 0)

U-(10 00300000000000 00000 0 0) + NSP(10 009009NGP00000 00000 0 0) +
VP(11 999000NGP01113 00000 0 0) =
SAB(10 002000NGP00113 00000 0 0)

U-(10 00300000000000 00000 0 0) + R-(10 003000NGP00000 00000 0 0) +
VP(11 999000NGP99113 00000 0 0) =
SAB(10 003000NGP00113 00000 0 0)

U-(10 00400000000000 00000 0 0) + SA(11 99900099900911 00000 0 0) =
KK(10 00100000000001 00000 0 0)

U-(10 00500000000000 00000 0 0) + SA(11 99900099900911 00000 0 0) =
KK(10 00101000000001 00000 0 0)

U-(10 00600000000000 00000 0 0) + SA(11 9990Y099900912 00000 0 0) =
KK(10 0010Y000000002 00000 0 0)

V-(10 0014YONPRAV12 Swwww 0 0) +
U-(10 00200000000000 00000 0 0) =
VB(10 0010YONPRAV52 Swwww 0 0)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/KB/CLYDNGPRAVIT/S/-w-/X/Z)

RULE NO.

V-(10 001000NGPRAV22 SWWWW 0 0) +
U-(10 00200000000000000000 0 0) =
VB(10 001000NGPRAV32 SWWWW 0 0)

V-(10 001410NGPRAV22 SWWWW 0 0) =
VB(10 001010NGPRAV32 SWWWW 0 0)

V-(10 0011YONG2RAV22 SWWWW 0 0) =
VB(10 0010YONG2RAV22 SWWWW 0 0)

V-(10 0011YONGPRAV1T SWWWW 0 0) =
VB(10 0010YONGPRAV1T SWWWW 0 0)

W-(10 001000000RAVO0 SWWWW 0 0) +
VBB(10 0090YONGPRAVIT SWWWW1117) =
VC(10 0010YONGPRAVIT SWWWW1117)

W-(10 0011Y0000RAV00 SWWWW 0 0) =
VB(10 0010Y0992RAV22 SWWWW 0 0)

W-(11 9913Y0000RA000 99999 0 0) +
VM(10 009000000RA000 00000 0 0) =
NW(10 0010Y0000RA000 00000 0 0)

Y-(10 0013Y0000RA000 99999 0 0) +
VM(11 999090999RA000 00000 0 0) =
NV(10 0010Y0000RA000 00000 0 0)
LISTING OF MODERN HEBREW SYNTAX ANALYSIS RULES 7/21/70--JAMES D. PRICE

SYMBOL(MF/,BCLYDNGPRAVIT/S/-w/-x/z) RULE NO.

Y-(10 0013Y0000RA000 99999 0 0) + (59 2)
R-(10 001300099900000 00000 0 0) +
VM(11 999090999RA000 00000 0 0) =
NV(10 0020Y0000RA000 00000 0 0)

Y-(10 0013Y0000RA000 99999 0 0) + (59 3)
RSP(11 99909999900000 00000 0 0) +
VM(11 999090999RA000 00000 0 0) =
NV(10 0030Y0000RA000 00000 0 0)

F-(11-10CLYDNGPRAVIT SWWWw 0 0) + (3 1)
T-(10 00100000000000 00000 0 0) +
F-(11 00CLYDNGPRAVIT SWWWw 0 0) =
F-(11 K1CLYDNGPRAVIT SWWWw 0 0)

F-(11-10CLYDNGPRAVIT SWWWw 0 0) + (3 2)
C-(10 00100000000000 00000 0 0) +
F-(11 00CLYDNGPRAVIT SWWWw 0 0) =
F-(11 K1CLYDNGPRAVIT SWWWw 0 0)

F-(11-10CLYDNGPRAVIT SWWWw 0 0) + (3 3)
C-(10 00200000000000 00000 0 0) +
F-(11 00CLYDNGPRAVIT SWWWw 0 0) =
F-(11 K2CLYDNGPRAVIT SWWWw 0 0)

F-(12-10CLYD99RAVIT SWWWw 0 0) + (3 4)
T-(10 00300000000000 00000 0 0) +
F-(12 0BCLYD99RAVIT SWWWw 0 0) =
F-(12 KBCLYDNGPRAVIT SWWWw 0 0)
SYMBOLS

RULE NO.

F-(12-10CLYD999RAVIT SWWWW 0 0) +
C-(10 00100000000000 00000 0 0) +
F-(12 00CLYD999RAVIT SWWWW 0 0) =
F-(12 K1CLYDNGPRAVIT SWWWW 0 0)

F-(12-10CLYD999RAVIT SWWWW 0 0) +
C-(10 00200000000000 00000 0 0) +
F-(12 00CLYD999RAVIT SWWWW 0 0) =
F-(12 K2CLYDNGPRAVIT SWWWW 0 0)

F-(13 00CLYDNGPRAVIT SWWWW 0 0) +
T-(10 00300000000000 00000 0 0) +
F-(13 00CLYDNGPRAVIT SWWWW 0 0) =
F-(13 20CLYDNGPRAVIT SWWWW 0 0)

F-(13-10CLYDNGPRAVIT SWWWW 0 0) +
T-(10 00300000000000 00000 0 0) +
F-(13 00CLYDNGPRAVIT SWWWW 0 0) =
F-(13 K0CLYDNGPRAVIT SWWWW 6 4)

F-(14 00CLYDNGPRAVIT SWWWW 0 0) +
F-(14 00CLYDNGPRAVIT SWWWW 0 0) =
F-(14 20CLYDNGPRAVIT SWWWW 0 0)

F-(14-10CLYDNGPRAVIT SWWWW 0 0) +
F-(14 00CLYDNGPRAVIT SWWWW 0 0) =
F-(14 K0CLYDNGPRAVIT SWWWW 5 4)
APPENDIX B
SOURCE LANGUAGE LISTING
OF
COMPUTER PROGRAM ANALYZ
AND
ASSOCIATED SUBPROGRAMS
PART IV
APPENDIX B

This appendix contains the source language listing of the following computer programs and associated subprograms:

<table>
<thead>
<tr>
<th></th>
<th>Main Program</th>
<th>Subprogram</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>ANALYZ</td>
<td>ALPHA</td>
</tr>
<tr>
<td>(2)</td>
<td>DIAGRM</td>
<td>LIMIT</td>
</tr>
<tr>
<td>(3)</td>
<td>MACHER</td>
<td>OUTPUT</td>
</tr>
<tr>
<td>(4)</td>
<td>PARSE</td>
<td>PROPH1</td>
</tr>
<tr>
<td>(5)</td>
<td>PROPH2</td>
<td>RERITE</td>
</tr>
<tr>
<td>(6)</td>
<td>RULENO</td>
<td>SYMACH</td>
</tr>
<tr>
<td>(7)</td>
<td>VARATT</td>
<td>RULIST</td>
</tr>
</tbody>
</table>
MAIN PROGRAM ANALYZ
*** *

PROGRAM ANALYZ

* THIS PROGRAM PERFORMS SYNTACTIC ANALYSIS

* OF HEBREW SENTENCES

* WRITTEN BY JAMES D. PRICE JULY, 1970

***

COMMON RULE(900,28), IPSI(100,8,2), RESTRT(5,15), ITABLE(400,29),
1 SYMIN(29), SYMBOL(29), I, IMAXI, J, JMAX, IRULE, IMAX, IRULE1,
2 ISTART, ITRACE, MATCH
COMMON/TREE/ ITREE(50), NODE(7,400), NODE1, NODE2
COMMON/PARS/ SYML(100,6), INDEX(100,8), AMSG(200,6), ENGLSH(20,4),
1 ATTVAL(17,8), ISYMB(100)
COMMON/ABC/ TRANSL(50)
COMMON/SYMB/ SYM(100)
INTEGER RULE, RESTRT, SYMIN, SYMBOL
LOGICAL MATCH
DIMENSION REEDER(36), ROOT(4), HEBREW(12)
READ(5,107) MODE, ITAPE, ITRACE, IOUTPT, IOUTRE, LIMAX
107 FORMAT(12I5)
REWIND ITAPE
IF(MODE.EQ.0) GO TO 10
READ(ITAPE) TRANSL, RULE, IPSI, RESTRT, SYM, SYML, ISYMB, INDEX,
1 AMSG, ATTVAL
REWIND ITAPE
GO TO 4

***

READ IN TRANSLITERATION

10 READ(5,100) NL, (TRANSL(L), L=1, NL)
100 FORMAT(I6, 50A1)

***

READ IN GRAMMAR RULES

I=0
ISYMNO=0
ICLASS=0
IRULE=0
1 I=I+1
READ(5,101) (REEDER(L), L=1, 36)
RULE(I, 1)=0
DO 11 L=1, 3
CALL ALPHA(REEDER(L), N)
11 RULE(I, 1)=RULE(I, 1)+ N*10**(3-L)
IF(RULE(I, 1).EQ.999) GO TO 3
DO 12 L=4, 5
L1=L-2
CALL ALPHA(REEDER(L), N)
12 RULE(I, L1)=N
CALL ALPHA(REEDER(6),N)
CALL ALPHA(REEDER(7),N)
RULE(I,*4)=N
IF(II.EQ.39) RULE(I,*4)=-RULE(I,*4)
DO 121 L=8,20
L1=L-3
CALL ALPHA(REEDER(L),N)
121 RULE(I,L1)=N
CALL ALPHA(REEDER(21),N)
CALL /ALPHA(REEDER(22),N)
RULE(I,18)=N1+N*10
IF(N,GT.0) RULE(I,18)=-RULE(I,18)
DO 122 L=23,27
L1=L-4
CALL ALPHA(REEDER(L),N)
122 RULE(I,L1)=N
RULE(I,24)=0
DO 13 L=28,30
CALL ALPHA(REEDER(L),N)
13 RULE(I,24)=RULE(I,24)+ N*10**(30-L)
DO 14 L=31,32
L1=L-6
CALL ALPHA(REEDER(L),N)
14 RULE(I,L1)=N
RULE(I,27)=0
DO 15 L=33,34
CALL ALPHA(REEDER(L),N)
15 RULE(I,27)=RULE(I,27)+ N*10**(34-L)
RULE(I,28)=0
DO 16 L=35,36
CALL ALPHA(REEDER(L),N)
16 RULE(I,28)=RULE(I,28)+N*10**(36-L)
IF(RULE(I,28).NE.1) GO TO 2
IC=RULE(I,6)
IF(IC.EQ.9) IC=1
IF(IC.EQ.ICLASS.AND.RULE(I,1).EQ.ISYMN0) GO TO 2
IF(RULE(I,1).EQ.97.AND.ISYMN0.EQ.97.AND.RULE(I,3).EQ.ICLASS)GOTO2
ISYMN0=RULE(I,1)
ICLASS=RULE(I,6)
IF(ISYMN0.EQ.97) ICLASS = RULE(I,3)
IF(ICLASS.EQ.9) ICLASS=1
1PSI(ISYMN0,ICLASS,1)=I
GO TO 1
2 IPSI(ISYMN0,ICLASS,2)=I
GO TO 1

*** READ IN RESTRICTION MATRIX

3 IRULE=1
READ(5,102)((RESTRT(I,J),I=1,5),J=1,15)
102 FORMAT(515).

*** READ IN SYMBOL ARRAY

READ(5,104) SYM
104 FORMAT(10(2X,A3))
DO 31 I=1,96
31 READ(5,400) (SYML(I,J),J=1,6),ISYMB(I)
400 FORMAT(6A6,30X,I6)
DO 32 I=1,100
32 READ(5,401) (INDEX(I,J),J=1,8)
401 FORMAT(8I5)
DO 33 I=1,185
33 READ(5,400) (AMSG(I,J),J=1,6)
402 FORMAT(8A6)
WRITE(ITAPE) TRANSL,RULE,IPSI,RESTRT,SYM,SYML,ISYMB,INDEX,AMSG,ATTVAL
REWRITE ITAPE
WRITE(6,200) TRANSL
200 FORMAT(5X,50A1)
WRITE(6,201)(NN,(RULE(NN,L),L=1,28),NN=1,IRULE)
201 FORMAT(29I4)
WRITE(6,202)(MM,((IPSI(MM,NN,L),L=1,2),NN=1,8),MM=1,100)
202 FORMAT(17I5)
WRITE(6,203)((RESTRT(NN,L),NN=1,5),L=1,15)
203 FORMAT(515)
WRITE(6,104) SYM
WRITE(6,403)((SYML(I,J),J=1,6),I=1,100)
403 FORMAT(1H1/(10X,6A6))
WRITE(6,404) ISYMB
404 FORMAT(//5X,100I1)
WRITE(6,405)((INDEX(I,J),J=1,8),I=1,100)
405 FORMAT(1H1/(10X,8I5))
WRITE(6,403)((AMSG(I,J),J=1,6),I=1,200)
WRITE(6,406)((ATTVAL(I,J),J=1,8),I=1,17)
406 FORMAT(1H1/(10X,8A6))
CLEAR REWRITE LIST MATRIX

CONTINUE
DO 5 I=1,400
DO 51 J=1,29
51 ITABLE(I,J) =0
DO 52 J =1,7
52 NODE(J,I) =0
5 CONTINUE

READ EQUIVALENT HEBREW SENTENCE AND INITIAL SYMBOLS

READ(5,105) HEBREW, NOP, IMAXI
105 FORMAT(12A6,2I3)
   IF(IMAXI.EQ.0) GO TO 999
   DO 6 I=1,IMAXI
      READ(5,103)(ITABLE(I,K),K=1,18),(ROOT(L),L=1,4),
         (ENGLSH(I,J),J=1,4)
103 FORMAT(16,001712,4A104A6)
   DO 61
      J=1,4
      J1= J+18
      CALL ALPHA(ROOT(J),NN)
61 ITABLE(I,J1)=NN
   NODE(1,I)=1
   NODE(4,I)=ITABLE(I,1)
   NODE(5,I)=ITABLE(I,6)
6 CONTINUE

LIM=0
ISTART=0
NODE1=0
NODE2=IMAXI
IPASS = 1
ITREE(1) = IMAXI
7 CONTINUE

J1=0
79 CONTINUE
   IF(ITRACE.NE.0) WRITE(6,301) IPASS,NODE1,NODE2
301 FORMAT(10X,18HNEW PASS---IPASS=,I3,9H, NODE1=,I3,
1 9H, NODE2=,I3,1H.)
   CALL RERITE(MON,J1,LIM)
   IF(LIM.GT.LIMAX) GO TO 76
   IF(NODE2.GT.400) GO TO 74
   IF(MON.EQ.0) GO TO 70
   IF(I.NE.IMAXI) GO TO 70
   IF(ITRACE.NE.0) WRITE(6,302) IPASS,IMAXI,NODE1,NODE2
302 FORMAT(10X,18HCOMPLETED PASS NO.,I3,10H. IMAXI=,I3,
1 9H, NODE1=,I3,9H, NODE2=,I3,1H.)
IPASS = IPASS + 1
IMAXI = NODE2 - NODE1
ITREE(IPASS) = IMAXI
ISTART = 0
IF(IMAXI.GT.1) GO TO 7
GO TO 75

*** HUNT BACKWARDS TO FIND FIRST NODE WITH ALTERNATE RULE.

70 CONTINUE
NODE(6,NODE1) = 0
NODE(7,NODE1) = 0
NODE1 = NODE1 - 1
IF(NODE1.LT.1) GO TO 73
IF(NODE(6,NODE1).LE.0) GO TO 70

*** SKIP IF GOVERNED NODE IS SAME AS COMPOUND GOVERNING NODE.

NG = NODE(3,NODE1)
IF(NODE(4,NODE1).NE.NODE(4,NG)) GO TO 271
IF(NODE(5,NODE1).NE.NODE(5,NG)) GO TO 271
IF(ITABLE(NODE1,4).EQ.0.AND.ITABLE(NG,4).GT.0) GO TO 70

271 CONTINUE
NODE2 = NODE(3, NODE1)
IRULE = NODE(6, NODE1)
IMAXI = NODE(7, NODE1)
ISUM = 0
IP = IPASS
DO 71 L = 1, IP
ISUM = ISUM + ITREE(L)
IPASS = L
IF(ISUM.GE.NODE1) GO TO 72

71 CONTINUE
72 IMAXI = ITREE(IPASS)

*** CANCEL ANY PREVIOUS PREDICTION

IF(NODE(6,NODE2).GE.0) GO TO 720
NOD11 = NODE1 + 1
NODE(6, NODE2) = 0
NODE(7, NODE2) = 0
NODE(6, NOD11) = 0
NODE(7, NOD11) = 0

720 CONTINUE
NODE2 = NODE2 - 1
NODE1 = NODE1 - 1
ISTART = IMAXI + NODE1 - ISUM
IF(ITRACE.NE.0) WRITE(6,303) IPASS, ISTART, IMAXI, NODE1, NODE2
303 FORMAT(10X,45H'SKIP ALTERNATE RULE BRANCH. WENT BACK TO PASS, IPASS.
16H, NODE, IMAXI, ISTART, NODE1, NODE2')
JI=1
GO TO 79
73 WRITE(6,173) NOP
173 FORMAT(5X,54HERROR DETECTED IN ALTERNATE RULE BRANCH FOR PROB. NO. 1,13,23H. ANALYSIS TERMINATED.//5X,22HSEE PROGRAM ANALYZ 70.)
GO TO 4
74 WRITE(6,174) NOP
174 FORMAT(5X,32H NODES EXCEED 400 FOR PROBLEM NO. 13, 1 23H. ANALYSIS TERMINATED.)
GO TO 4
75 CONTINUE
IF(IOUTPT.EQ.1) CALL OUTPUT(IPASS)
WRITE(6,106) HEBREW
106 FORMAT(1H1//10X,26HHEBREW SENTENCE ANALYZED--//10X,12A6)
IF(IOUTRE.EQ.1) CALL DIAGRM(IPASS)
CALL PARSE(IPASS,NOP)
WRITE(6,304) NOP,LIM
304 FORMAT(///5)(t36HANALYSIS COMPLETED FOR SENTENCE NO. 13, 1 23H. NO. OF SYMBOL TESTS= 14, 1H.)
GO TO 4
76 WRITE(6,176) LIMAX,NOP
176 FORMAT(5X,21HNO ANALYSIS FOUND BY 16,52H SYMBOL TESTS. ANALYSIS TERMINATED FOR PROBLEM NO. 13, 1H.)
GO TO 4
999 WRITE(6,199)
199 FORMAT(1H1,10(/),1X,17(7H--END--))
STOP
END
SUBPROGRAM ALPHA
SUBROUTINE ALPHA'A,IA)

*****************************************************************
* THIS SUBROUTINE CONVERTS THE ALPHA-NUMERIC SYMBOL (A) TO AN
* INTEGER EQUIVANENT BASED ON DATA IN ARRAY TRANSL.
* FOR USE WITH PROGRAM ANALYZ
* WRITTEN BY JAMES D. PRICE  JULY, 1970
*****************************************************************

COMMON/ABC/ TRANSL(50)

DO 1 L=1,50
   IF(A.EQ.TRANSL(L)) 1,2,1
1 CONTINUE
   IA=49
   RETURN
   2 IA=L-1
   IF(IA.EQ.40) IA=0
   RETURN
END
SUBPROGRAM DIAGRM
SUBROUTINE DIAGRM(IPASS)

******************************************************************************
*  THIS SUBROUTINE CONSTRUCTS A TREE DIAGRAM                       *
*  OF THE SENTENCE ANALYSIS.                                           *
*  FOR USE WITH PROGRAM ANALYZ                                       *
*  WRITTEN BY JAMES D. PRICE  JULY, 1970                           *
******************************************************************************

COMMON RULE(900,28), IPSI(100,8,2), RESTRT(5,15), ITABLE(400,29),
1 SYMIN(29), SYMBOL(29), IMAXI, J, JMAXI, IRULE, IMAX, IRULE1,
2 ISTART, ITRACE, MATCH

COMMON/TREE/ ITREE(50), NODE(7,400), NODE1, NODE2
INTEGER RULE, RESTRT, SYMIN, SYMBOL
COMMON/SYM/ SYM(100)
DIMENSION ALINE(40), CLS(9)
DATA BLANK/3H /
DATA DASHES/3H--/
DATA BAR/3H I/
DATA CLS/3H1 ,3H2 ,3H3 ,3H4 ,3H5 ,3H6 ,3H7 ,3H8 ,3H9 /
IF(ITRACE.NE.0) WRITE(6,99)
99 FORMAT(30H SUBROUTINE DIAGRAM CALLED )

***

COMPUTE LINE POSITION FOR EACH NODE

NP=ITREE(1)
DO 3 L=1,NP
3 NODE(2,L)=L
L=0
4 L=L+1
NG=NODE(3,L)
IF(NODE(1,NG).GT.1) GO TO 5
NODE(2,NG)=NODE(2,L)
GO TO 6
5 L1=L+NODE(1,NG)-1
NODE(2,NG)= (NODE(2,L)+NODE(2,L1))/2
L=L1
6 IF(L.LT.NODE1) GO TO 4
IF(ITRACE.EQ.0) GO TO 71
WRITE(6,103)
103 FORMAT(3H1//10X,11HNODE MATRIX//)
DO 7 L=1,NODE1
7 WRITE(6,102)(NODE(I,L),I=1,5)
102 FORMAT(10X,5I5)
71 CONTINUE

4-B-12
WRITE OUT TREE DIAGRAM

WRITE(6,100)

100 FORMAT(/10X,35H TREE DIAGRAM OF HEBREb SENTENCE //)

NO=1
DO 50 M=1,IPASS
IF(M.EQ.1) GO TO 31

WRITE UPPER/LOWER CONNECTORS (VERTICAL)

21 DO 30 L=1,20
IF((NODE(1,NO).EQ.0.OR.NODE(2,NO).EQ.0).AND.N0.LE.NODE1) NO=NO+1
   L1=L+L-1
   L2=L+L
   IF(L.EQ.NODE(2,NO)) GO TO 22
   ALINE(L2)=BLANK
   ALINE(L1)=BLANK
   GO TO 30
22 ALINE(L1)=BAR
   NO=NO+1
30 CONTINUE
NO=NO-ITREE(M)
WRITE(6,101) ALINE
IF(IUPLLOW.EQ.2) GO TO 42

WRITE LINE OF NODES

31 DO 40 L=1,20
   IF((NODE(1,NO).EQ.0.OR.NODE(2,NO).EQ.0).AND.N0.LE.NODE1) NO=NO+1
   L1=L+L-1
   L2=L+L
   IF(L.EQ.NODE(2,NO)) GO TO 32
   ALINE(L1)=BLANK
   ALINE(L2)=BLANK
   GO TO 40
32 NS=NODE(4,NO)
   NT=NODE(5,NO)
   NG=NODE(3,NO)
   NO=NO+1
   IF(M.EQ.1) GO TO 33
   IF(NS.EQ.NODE(4,NG).AND.NT.EQ.NODE(5,NG)) GO TO 34
33 ALINE(L1)=SYM(NS)
   ALINE(L2)=CLS(NT)
   GO TO 40
34 IF(M.EQ.IPASS) GO TO 33
   ALINE(L2)=BLANK
   ALINE(L1)=BAR
40 CONTINUE
NO=NO-ITREE(M)
WRITE(6,101) ALINE

*** WRITE LOWER CONNECTORS (VERTICAL) ***

41 IUPLow=2
IF(M.EQ.IPAS) GO TO 50
GO TO 21

*** WRITE LOWER CONNECTORS (HORIZONTAL) ***

42 IEND=0
IENDP0=0
DO 20 L=1,20
IF((NODE(1,NO).EQ.0.OR.NODE(2,NO).EQ.0).AND.NO.LE.NODE1) NO=NO+1
L1=L+L-1
L2=L+L
IF(L.EQ.IENDP0) IEND=1
IF(L-NODE(2,NO)) 10,11,12
10 ALINE(L1)=BLANK
ALINE(L2)=BLANK
GO TO 20
11 NG=NODE(3,NO)
NO1=NODE(1,NG)
NO2=NO+NO1-1
IENDP0=NODE(2,NO2)
ALINE(L1)=BLANK
IF(IENDP0.GT.L) GO TO 13
ALINE(L1)=BAR
NO=NO+1
GO TO 20
12 IF(IEND.EQ.1) GO TO 14
IF(L.GT.IENDP0) GO TO 10
ALINE(L1)=DASHES
13 ALINE(L2)=DASHES
GO TO 20
14 ALINE(L1)=DASHES
ALINE(L2)=BLANK
NO=NO+NO1
IEND=0
IENDP0=0
20 CONTINUE
NO=NO-ITREE(M)
IUPLow=1
WRITE(6,101) ALINE
101 FORMAT(1X,40A3)
50 NO=NO+ ITREE(M)
RETURN
END

4-B-14

161
SUBPROGRAM LIMIT
SUBROUTINE LIMIT(INSYM)

********************************************************************************************************************
* THIS SUBROUTINE CHECKS A GIVEN SYMBOL AGAINST SPECIFIED LIMITATIONS. IF NOT SATISFIED, VARIABLE MATCH IS SET TO FALSE. *
* FOR USE WITH PROGRAM ANALYZ *
* WRITTEN BY JAMES D. PRICE JULY, 1970 *
********************************************************************************************************************

COMMON RULE(900,28), IPSI(100,8,2), RESTRT(5,15), ITABLE(400,29),
1 SYMIN(29), SYMBOL(29), I, IMAXI, J, JMAX, IRULE, IMAX, IRULE1,
2 ISTART, ITRACE, MATCH
INTEGER RULE, RESTRT, SYMIN, SYMBOL
DIMENSION INSYM(29)
LOGICAL MATCH
IF (ITRACE.NE.0) WRITE(6,99)
99 FORMAT(30H SUBROUTINE LIMIT CALLED )
IR=INSYM(27)
IV=INSYM(28)
IM=RESTRT(1,IR)+1
DO 2 L=2,IM
  IF (RESTRT(L,IR).GE.0) GO TO 1
  MATCH=.TRUE.
  IREST=-RESTRT(L,IR)
  IF (INSYM(IV).NE.IREST) GO TO 2
  GO TO 3
1 IF (INSYM(IV).EQ.RESTRT(L,IR)) GO TO 14
   MATCH=.FALSE.
2 CONTINUE
   RETURN
3 MATCH=.FALSE.
   RETURN
14 MATCH=.TRUE.
   RETURN
   END
SUBPROGRAM MACHER
SUBROUTINE MACHER(IS1, IS2, MATCH)

******************************************************************************
* THIS SUBROUTINE TESTS FOR A MATCH BETWEEN                            *
* SUBSCRIPTS OF GIVEN SYMBOLS IS1 AND IS2.                              *
* CALLED BY PROPH1 AND PROPH2.                                          *
* WRITTEN BY JAMES D. PRICE OCTOBER 1970                                *
******************************************************************************

DIMENSION IS1(29), IS2(29)
LOGICAL MATCH

MATCH = .TRUE.
IF(IS1(1).NE.IS2(1)) GO TO 4
DO 3 L = 2, 17
IF(IS1(L).EQ.IS2(L)) GO TO 3
IF(L.EQ.3) GO TO 3
IF(L.EQ.7.AND.IS1(8).EQ.0) GO TO 3
IF(IS1(L).EQ.9) GO TO 3
IF(IS2(L).EQ.9) GO TO 3
IF(IS2(L).LT.0.AND.IS1(L).LT.9) GO TO 3
IF(IS2(L).GT.9.AND.IS1(L).LT.9) GO TO 3
GO TO 4
3 CONTINUE
RETURN
4 MATCH = .FALSE.
RETURN
END
SUBPROGRAM OUTPUT
SUBROUTINE OUTPUT(IPASS)

***************************************************************
* THIS SUBROUTINE WRITES THE RESULTS OF THE ITH REWRITE PASS  *
* OF THE GRAMMAR                                               *
* FOR USE WITH PROGRAM ANALYZ                                   *
* WRITTEN BY JAMES D. PRICE  JULY, 1970                       *
***************************************************************

COMMON RULE(900, 28), IPSI(100, 8, 2), RESTRT(5, 15), ITABLE(400, 29),
  1 SYMIN(29), SYMBOL(29), I, IMAXI, J, JMAX, IRULE, IMAX, IRULE
  2 ISTART, ITRACE, MATCH
COMMON/TREE/ ITREE(50), NODE(7, 400), NODE1, NODE2
COMMON/ABC/ TRANSL(50)
INTEGER RULE, RESTRT, SYMIN, SYMBOL
LOGICAL MATCH
IF(ITRACE.NE.0) WRITE(6, 99)
99 FORMAT( 30H SUBROUTINE OUTPUT CALLED )
  J1=1
  DO 2 ITH =1, IPASS
   WRITE(6, 100) ITH
100 FORMAT(1H1//9X, 8HRESULTS OF REWRITE PASS NO.--I3//
1105H SYMBOL M F K B C L Y D N G P R A V I T
  2 S    W    T Y RULE EL NE REST SUB X//)
   J2=J1+ITREE(ITH)-1
   DO 1 JJ= J1, J2
      IR1=ITABLE(JJ, 19)+1
      IR2=ITABLE(JJ, 20)+1
      IR3=ITABLE(JJ, 21)+1
      IR4=ITABLE(JJ, 22)+1
   1 WRITE(6, 101) (ITABLE(JJ, L)+1, L=1, 18), TRANSL(IR1),
      1 TRANSL(IR2), TRANSL(IR3), TRANSL(IR4), (ITABLE(JJ, L)+1, L=23, 29)
101 FORMAT(6X, I3, 4X, 17I3, 4X, 4A1, 3X, I3, 3X, I3, 3I3, 3(2X, I3))
   J1=J2+1
2 CONTINUE
RETURN
END
SUBROUTINE PARSE(IPASS,NOP)

*********************************************************************
* THIS SUBROUTINE ASSEMBLES STATEMENTS ABOUT THE
* SYNTACTIC ANALYSIS OF A SENTENCE. CALLED FROM
* PROGRAM ANALYZ.
* WRITTEN BY JAMES D. PRICE
* FIRL AUG, 1970
*********************************************************************

COMMON RULE(290,28), IPSI(100,8,2), RESTRT(5,15), ITABLE(400,29), 1  
SYMIN(29), SYMBOL(29), I, IMAX, J, JMAX, IRULE, IMAX, IRULE1, 2  
ISTART, ITRACE, MATCH

COMMON/TREE/ ITREE(50), NODE(7,400), NODE1, NODE2

COMMON/PARS/ SYML(100,6), INDEX(100,8), AMSG(200,6), ENGLISH(20,4), 1  
ATTVAL(17,8), SYMB(100)

INTEGER RULE, RESTRT, SYMIN, SYMBOL

DIMENSION ACCUM(200)

DATA THE/6H, THE/T0F/6H, OF/ AS/6H, IS/.COMMA/6H,.

WRITE(6,200) NOP

200 FORMAT(1H1//10X,31HSYNTAX ANALYSIS IF SENTENCE NO. I4,1H///)

*** FOR EACH PASS, EXAMINE SYMBOLS FOR SYNTAX DATA.

NAN=0
NTERM=ITREE(1)
N2= NODE2
DO 90 I=IPASS,1,-1
N1= N2 - ITREE(I)+1
IF(N1.GT.N2.OR.N1.LE.0) GO TO 90
DO 80 J=N1,N2

*** ACCUMULATE DATA ABOUT SYMBOL AT NODE J

ACCUM(1)=THE
M1=2
IS= NODE(4,J)
IF(IS.GT.76.AND.J.GT.NTERM) GO TO 80
IF(IS.LE.0) GO TO 80
IC= NODE(5,J)
ISS= IS
ICC= IC
M2=M1+ISYMB(IS)-1
M3=0
DO 1 M=M1,M2
M3=M3+1
1 ACCUM(M)=SYML(IS,M3)

4-B-22
NG = NODE(3, J)
IF(NG.LE.0 OR. NG.GT. NODE2) GO TO 83

81 NSG = NODE(4, NG)
IF(NSG.LE.0) GO TO 82
NSC = NODE(5, NG)

C *** SKIP IF NODE HAS ALREADY BEEN TREATED.
C *** SKIP IF GOVERNING NODE IS SAME AS GOVERNED NODE.
C
IF(IS0E00 NSG AND. IC.EQ. NSC) GO TO 80
IF(IS. EQ. NSG AND. ICC EQ. NSC) GO TO 82
M1 = M2 + 1
M2 = M1 + 1
ACCUM(M1) = 0F
ACCUM(M2) = THE
M1 = M2 + 1
M2 = M2 + ISYMB(NSG)
M3 = 0
DO 2 M = M1, M2
M3 = M3 + 1
2 ACCUM(M) = SYML(NSG, M3)

82 NG = NODE(3, NG)
IF(NG.EQ.0 OR. NG.GE. NODE2) GO TO 83
ISS = NSG
ICC = NSC
GO TO 81

83 CONTINUE
IF(IS.GT.76) GO TO 84
IMSG = INDEX(IS, IC)
IF(IMSG = 0) GO TO 80
NAN = NAN + 1
WRITE(6, 100) NAN, (ACCUM(M), M = 1, M2), (AMSG(IMSG, L), L = 1, 6)
100 FORMAT(/10X, 1H(o13,2H), 10A6/20(10X, 11A6/))
GO TO 80

84 CONTINUE
NAN = NAN + 1
WRITE(6, 100) NAN, (ACCUM(M), M = 1, M2), AS, (ENGLISH(J, L), L = 1, 4)
M1 = 0
DO 85 M=3,17
IF(M.EQ.6 OR M.EQ.7) GO TO 85
IF((M.EQ.3 OR M.EQ.4) .AND. (ITABLE(J,3).EQ.0 OR ITABLE(J,4).EQ.0))
1 GO TO 85
MM= ITABLE(J,M)
IF(MM.EQ.0) GO TO 85
M1=M1+1
ACCUM(M1)= ATTVAL(M*MM)
85 CONTINUE
IF(M1.EQ.0) GO TO 80
M11=M1-1
WRITE(6,101) (ACCUM(M),COMMA,M=1,M11), ACCUM(M1) ,DOT
101 FORMAT(10X, 9(A6,A2))
80 CONTINUE
. N2 = N1-1
90 CONTINUE
RETURN
END
SUBPROGRAM PROPH1
SUBROUTINE PROPH1(ID, IR, IM)

*******************************************************************************
  THIS SUBROUTINE EXAMINES THE PRESENT SYMBOL FOR POSSIBLE FUTURE SATISFACTION
  OF GRAMMAR RULE AFTER ONE MORE PASS. CALLED FROM SUBROUTINE RERITE.
  WRITTEN BY JAMES D. PRICE AUG. 1970
*******************************************************************************

COMMON RULE(900, 28), IPSI(100, 8, 2), RESTRT(5, 15), ITABLE(400, 29),
  SYMIN(29), SYMBOL(29), I, IMAXJ, JMAX, IRULE, IMAX, IRULE1,
  ISTART, ITRACE, MATCH
COMMON/TREE/ ITREE(50), NODE(7, 400), NODE1, NODE2
DIMENSION IS1(29), IS2(29), IS3(29), IS4(29)
INTEGER RULE, RESTRT, SYMIN, SYMBOL
LOGICAL MATCH
IF (ITRACE .NE. 0) WRITE(6, 199)
199 FORMAT(10X, 25HSUBROUTINE PROPH1 CALLED.)
IF (SYMIN(1) .EQ. RULE(IRULE1, 1)) AND (SYMIN(6) .EQ. 9 .OR.
  SYMIN(6) .EQ. RULE(IRULE1, 6))) GO TO 91

*** LOOK AHEAD ONE LEVEL

IN = NODE1
IR = 0
IM = 0
IS = SYMIN(1)
IC = SYMIN(6)
DO 1 L = 1, 8
  IR1 = IPSI(IS, L, 1)
  IF (IR1 .NE. 0) GO TO 2
1 CONTINUE
  GO TO 92
2 DO 3 L = 8, 1, -1
  IR3 = IPSI(IS, L, 2)
  IF (IR3 .NE. 0) GO TO 4
3 CONTINUE
  GO TO 92
4 IR2 = IR1 + RULE(IR1, 26)
IR4 = IR1
401 CONTINUE
  DO 41 L = 1, 29
    IS1(L) = ITABLE(IN, L)
    IS2(L) = RULE(IR1, L)
    IS3(L) = RULE(IRULE1, L)
    IS4(L) = RULE(IR2, L)
41 CONTINUE
CALL MACHE(I$1,IS2,MATCH)
IF(.NOT.MATCH) GO TO 51
CALL MACHE(IS$3,IS4,MATCH)
IF(MATCH) GO TO 6
CALL PROPH2(ID,IR2)
IF(ID.EQ.4) GO TO 94
CONTINUE
5
ID=0
IR$1= IR2+1
IF(IR1.GT.I$3) GO TO 93
GO TO 4
51 IF(IS2(I$2).EQ.1) GO TO 5
GO TO 7
CONTINUE
6
IN=IN+1
CONTINUE
7
IN=IN+1
IF(IR1.GE.IR2) GO TO 94
GO TO 401

*** NO FUTURE SATISFACTION, PRESENT RULE FAILED IN SYMACH OR LIMIT
91 ID=1
GO TO 99

*** NO FUTURE SATISFACTION, NO RULES TO PREDICT CONDITION.
92 ID=2
GO TO 99

*** NO FUTURE SATISFACTION, PREDICTION RULES EXHAUSTED.
93 ID=3
GO TO 99

*** FUTURE SATISFACTION PREDICTED.
94 ID=4
IR=IR4
IM=IR3
IF(ITRACE.NE.0) WRITE(6,299) ID
299 FORMAT(10X, 3HID=rI201H.)
RETURN
END
SUBROUTINE PROPH2 (ID, IR)

*************************************************************************
* THIS SUBROUTINE EXAMINES THE SYMBOL DEFINED BY IR IN THE RULE MATRIX FOR POSSIBLE FUTURE SATISFACTION OF THE GIVEN GRAMMAR RULE AFTER TWO PASSES. CALLED FROM SUBROUTINE PROPH1.
* WRITTEN BY JAMES D. PRICE OCTOBER, 1970
*************************************************************************

COMMON RULE(900,26), IPSI(100,8,2), RESTRT(5,15), ITABLE(400,29),
1 SYMIN(29), SYMBOL(29), I, IMAX, J, JMAX, IRULE, IMAX, IRULE1,
2 ISTART, ITRACE, MATCH
DIMENSION IS1(29), IS2(29), IS3(29), IS4(29)
INTEGER RULE, RESTRT, SYMIN, SYMBOL
LOGICAL MATCH
IF (ITRACE .NE. 0) WRITE (6, 199)
199 FORMAT (10X, 25H SUBROUTINE PROPH2 CALLED.)

*** LOOK AHEAD ONE LEVEL

IS = RULE (IR, 1)
IC = RULE (IR, 6)
DO 1 L = 1, 8
IR1 = IPSI (IS, L, 1)
IF (IR1 .NE. 0) GO TO 2
1 CONTINUE
GO TO 92
2 DO 3 L = 8, 1, -1
IR3 = IPSI (IS, L, 2)
IF (IR3 .NE. 0) GO TO 4
3 CONTINUE
GO TO 92
4 IR2 = IR1 + RULE (IR1, 26)
DO 41 L = 1, 29
IS1 (L) = RULE (IR, L)
IS2 (L) = RULE (IR1, L)
IS3 (L) = RULE (IRULE1, L)
IS4 (L) = RULE (IR2, L)
41 CONTINUE
CALL MACHEX (IS1, IS2, MATCH)
IF (.NOT. MATCH) GO TO 5
CALL MACHEX (IS3, IS4, MATCH)
IF (MATCH) GO TO 94
5 CONTINUE
IR1 = IR2 + 1
IF (IR1 .GT. IR3) GO TO 93
GO TO 4
C *** NO FUTURE SATISFACTION, NO RULES TO PREDICT CONDITION

92 ID=2
   GO TO 99

C *** NO FUTURE SATISFACTION, PREDICTION RULES EXHAUSTED.

93 ID=3
   GO TO 99

C *** FUTURE SATISFACTION PREDICTED.

94 ID=4
99 IF(ITRACE.NE.0) WRITE(6,299) ID
299 FORMAT(10X,10HPROPH2 ID=I2,1H.)
   RETURN
   END
SUBPROGRAM RERITE
SUBROUTINE RERITE(MON,J1,LIM)

************************************************************************************
** THIS SUBROUTINE REWRITES A STRING OF SYMBOLS ACCORDING TO APPLICABLE **
** GRAMMAR RULES. CALLED FROM ANALYZ FOR USE WITH PROGRAM ANALYZ **
** WRITTEN BY JAMES D. PRICE JULY, 1970 **
************************************************************************************

COMMON RULE(900,28), IPSI(100,8,2), RESTR(5,15), ITABLE(400,29),
1 SYMIN(29), SYMBOL(29), I, IMAX, J, JMAX, IRULE, IMAX, IRULE1,
2 ISTART, ITRACE, MATCH
COMMON/TREE/ ITREE(50), NODE(7,400), NODE1, NODE2
INTEGER RULE, RESTR, SYMIN, SYMBOL
LOGICAL MATCH
IF(TRACE NE.0) WRITE(6,99)

FORMAT(30H SUBROUTINE RERITE CALLED )
99 FORMAT(NSTART = NODE1
MSTART = NODE2
I=0
J=0
MON=0
J2=1
J3=0
1 IF(J2 NE.1) GO TO 101
I = ISTART
NODE1 = NSTART
NODE2 = MSTART
101 IF(I GE. IMAX) RETURN
LIM=LIM+1
I = I +1
NODE1 = NODE1 +1
DO 2 L=1,29
2 SYMIN(L) = ITABLE( NODE1, L)
21 CONTINUE
IF(J .EQ. 0) GO TO 41
NODE(6,NODE1)=0
NODE(7,NODE1)=0
GO TO 4
41 CALL RULENO(J1,J2,J3)
NODE(6,NODE1) = IRULE
NODE(7,NODE1) = IMAX
IX=IRULE+RULE(IRULE+26)
IF(IX NE. IMAX) GO TO 42
NODE(6,NODE1)=0
NODE(7,NODE1)=0
42 CONTINUE
IF(IRULE.EQ.0) GO TO 9
IRULE1 = IRULE-1
JMAX = RULE(IRULE,26)
JJ = JMAX + IRULE
DO 3 L=1,28
3 SYMBOL(L) = RULE(JJ,L)
4 J = J+1
IRULE1 = IRULE1+1
IF(J.GT.JMAX) GO TO 71
CALL SYMACH
IF(MATCH) GO TO 6
ID=0
IF(J2.EQ.2) CALL PROP1(ID,IR,IM)
IF(ID.EQ.4) GO TO 91
IF(IRULE(IRULE1+2).EQ.1) GO TO 52
SYMBOL(26) = SYMBOL(26)-1
GO TO 4
5 J1= J1+1
51 I = ISTART
NODE1 = NSTART
NODE2 = MSTART
J = 0
IF(NODE(6,NODE2).EQ.0) GO TO 101
J1=0
NOD11=NODE1+1
NODE(6,NOD11)=0
NODE(7,NOD11)=0
NODE(6,NODE2)=0
NODE(7,NODE2)=0
GO TO 101
52 CONTINUE
IF(J3.EQ.0) GO TO 5:
J1=J1+1
NODE1=NSTRT1
I=ISTRT1
J=0
GO TO 21
6 CALL VARATT
NODE(3,NODE1) = NODE2 +1
IF(J.EQ.JMAX) GO TO 7
GO TO 101
71 NODE1=NODE1+1
I=I-1
7 CALL LIMIT(SYMBOL)
IF(.NOT.MATCH) GO TO 52
IF(J2.NE.1) GO TO 73
J3=J3+1
DO 72 L=1,29
72 SYMIN(L)=SYMBOL(L)
J=0
J1=0
ISTRT1=I
NSTRT1=NODE1
GO TO 21
73 CONTINUE
  MON=MON+1
  ISTART = I
  NODE2 = NODE2 +1
  IF(NODE2.GT.400) RETURN
  NODE(4,NODE2)=SYMBOL(1)
  NODE(5,NODE2)=SYMBOL(6)
  NODE(1,NODE2) = SYMBOL(26)
  IF(SYMBOL(4).EQ.1) SYMBOL(4)=2
  K=SYMBOL(4)
  IF(K.EQ.0) GO TO 74
  K1=SYMBOL(3)
  K2=SYMBOL(5)
  K3=K-1
  IF(K1.EQ.3.OR.K2.EQ.3) K3=K
  IF(K1.EQ.4) K3=0
  NODE(1,NODE2)=K+K3
74 CONTINUE
  DO 8 L=1,29
  8 ITABLE( NODE2, L)= SYMBOL(L)
     GO TO 11
  9 CONTINUE
  IF(J2.EQ.1.AND.J3.NE.0.AND.SYMBOL(1).EQ.97) GO TO 1
  IF(J3.NE.0) GO TO 73
  IF(J2.EQ.2) GO TO 91
  J1=0
  J2=2.
  GO TO 51
91 CONTINUE
  N1=NODE1
  ISTART=ISTART+1
  NODE1=NSTART+1
  NODE2 = NODE2 +1
  IF(NODE2.GT.400) RETURN
  IF(ID.NE.4) GO TO 92
  NODE(6,NODE2)=IRULE
  NODE(7,NODE2)=IMAX
  NODE(6,N1) = -IR
  NODE(7,N1) = IM
92 CONTINUE
  NODE(3,NODE1) = NODE2
  NODE(1,NODE2) = 1
  NODE(4,NODE2)=ITABLE(NODE1,1)
  NODE(5,NODE2)=ITABLE(NODE1,6)
  DO 10 L=1,29
10 ITABLE( NODE2, L)= ITABLE(NODE1,L)
11 J=0
  J1=0
  J2=1
  J3=0
  NSTART = NODE1
  MSTART = NODE2
  IF(ITRACE.NE.0).WRITE(6,300) NODE2,(ITABLE(NODE2,L),L=1,29)
12 END
300 FORMAT(10X,8HNODE NO.,I3,4H IS ,29I3)
SUBROUTINE RULENO(J1, J2, J3)

********************************************************************
* THIS SUBROUTINE COMPUTES THE ROW NUMBER OF THE FIRST AND LAST    *
* SYMBOL OF THE SET OF RULES GOVERNING A GIVEN INPUT SYMBOL.      *
* CALLED FROM SUBROUTINE RERITE.                                 *
* FOR USE WITH PROGRAM ANALYZ                                     *
* WRITTEN BY JAMES D. PRICE JULY, 1970                           *
********************************************************************

COMMON RULE(900,28), IPSI(100,8,2), RESTRT(5,15), ITABLE(400,29),
1 SYMIN(29), SYMBOL(29), I, IMAX, J, JMAX, IRULE, IMAX, IRULE1,
2 ISTART, ITRACE, MATCH
COMMON/TREE/ ITREE(50), NODE(7,400), NODE1, NODE2
INTEGER RULE, RESTRT, SYMIN, SYMBOL
LOGICAL MATCH
IF(ITRACE .NE. 0) WRITE(6,99)
99 FORMAT( 30H SUBROUTINE RULENO CALLED )
IF(J1 .NE. 0) GO TO 10
GO TO (1,2), J2
1 IC=4
GO TO 3
12 IRULE = IPSI(IS,1,1)
IMAX = IPSI(IS,3,2)
IC=1
GO TO 6
2 CONTINUE
IS=SYMIN(1)
IC=SYMIN(6)
IF(NODE(6,NODE1).EQ.0) GO TO 21
IRULE=IABS(NODE(6,NODE1))
IMAX=NODE(7,NODE1)
GO TO 6

4-B-36
21 CONTINUE
IF(IC.EQ.9) GO TO 20
IRULE=IPSI(IS,1,1)
IF(IRULE.EQ.0) GO TO 3
IF(RULE(IRULE,6).EQ.9) GO TO 31
3 IRULE=IPSI(IS,IC,1)
31 IMAX =IPSI(IS,IC,2)
IF(IRULE.EQ.0) GO TO 20
IF(IMAX.EQ.0) IMAX=IPSI(IS,1,2)
GO TO 6
19 IRULE=0
IMAX =0
GO TO 6
FOR CLASS=9, INCLUDE ALL RULES ON THIS SYMBOL
C
C
20 IRULE=0
DO 30 L=1,8
IF(IPSI(IS,L,1).EQ.0) GO TO 30
IRULE = IPSI(IS,L,1)
GO TO 40
30 CONTINUE
40 IMAX = 0
DO 50 L=8,1,-1
IF(IPSI(IS,L,2).EQ.0) GO TO 50
IMAX = IPSI(IS,L,2)
GO TO 6
50 CONTINUE
6 IF(ISTRACE.NE.0) WRITE(6,300) IS,IC,IRULE,IMAX
300 FORMAT(10X,I4HFOR SYMBOL NO.,I3,8H, CLASS ,I3,9H, IRULE= ,I3, 1 8H, IMAX= ,I3,1H.)
RETURN
END
SUBPROGRAM SYMACH
SUBROUTINE SYMACH

**********************************************************************
* THIS SUBROUTINE TESTS FOR A MATCH                                    *
* BETWEEN SUBSCRIPTS OF A GIVEN SYMBOL                                  *
* AND A CORRESPONDING GRAM. RULE SYMBOL.                              *
* CALLED BY RERITE FROM ANALYZ.                                        *
* WRITTEN BY JAMES D. PRICE JULY, 1970                                *
**********************************************************************

COMMON RULE(900,28), IPSI(100,8,2), RESTRT(5,15), ITABLE(400,29),
1 SYMIN(29), SYMBOL(29), I, IMAX, J, JMAX, IRULE, IMAX, IRULE1,
2 ISTART, ITRACE, MATCH
COMMON TREE, ITREE(50), NODE(7,400), NODE1, NODE2
INTEGER RULE, RESTRT, SYMIN, SYMBOL
LOGICAL MATCH
IF(ITRACE.NE.0) WRITE(6,99) NODE1, SYMIN(1)
99 FORMAT(39H SUBROUTINE SYMACH CALLED FOR NODE ,13, 1)
1 7H, SYM. ,97,1H,
MATCH= .TRUE.,
JJ=IRULE+RULE(IRULE,26)
DC 3 L=1,17
IF(L.NE.1) GO TO 1
IF(SYMIN(1).EQ.RULE(IRULE1,L))GO TO 3
IF(RULE(IRULE1,L).EQ.97.AND.SYMBOL(1).EQ.97) GO TO 3
IF(RULE(IRULE1,L).EQ.97.AND.SYMIN(1).EQ.SYMBOL(1)) GO TO 3
GO TO 4
1 CONTINUE
IF(L.EQ.7.AND.SYMIN(8).EQ.0) GO TO 3
IF(L.EQ.3.AND.RULE(IRULE1,L).NE.97) GO TO 3
IF(RULE(IRULE1,L).EQ.9) GO TO 3
IF(SYMIN(L).EQ.RULE(IRULE1,L)) GO TO 3
IF(SYMIN(L).EQ.9) GO TO 3
IF(RULE(IRULE1,L).LT.0.AND.SYMIN(L).LT.9) GO TO 3
IF(RULE(IRULE1,L).GT.9.AND.RULE(IRULE1,L).EQ.RULE(JJ,L))GO TO 2
GO TO 4
2 IF(SYMBOL(L).EQ.RULE(IRULE1,L)) GO TO 3
IF(SYMBOL(L).LT.9.AND.SYMBOL(L).EQ.SYMIN(L)) GO TO 3
IF(SYMBOL(L).EQ.9) GO TO 3
GO TO 4
3 CONTINUE
CALL LIMIT(SYMIN)
RETURN
4 MATCH=.FALSE.
IF(ITRACE.NE.0) WRITE(6,300) L
300 FORMAT(10X,21HMATCH IS FALSE FOR L=,13,1H,)
RETURN
END
SUBROUTINE VARATT

********** THIS SUBROUTINE COMPUTES THE VALUE OF THE DEPENDENT SUBSCRIPTS OF A GIVEN GRAMMAR RULE. CALLED BY RERITE FOR USE WITH PROGRAM ANALYZ WRITTEN BY JAMES D. PRICE JULY, 1970

**********

COMMON RULE(900,28), IPSI(100,8,2), RESTRT(5,15), ITABLE(400,29), SYMIN(29), SYMBOL(29), IMAXI,J, JMAX, IRULE, IMA, IRULE1.

2 ISTART, ITRACE, MATCH COMMON/TREE/ITREE(50), NODE(7,400), NODE1, NODE2 INTEGER RULE, RESTRT, SYMIN, SYMBOL LOGICAL MATCH

IF(TRACE .NE. 0) WRITE(6,99)

99 FORMAT(30H SUBROUTINE VARATT CALLED ) IF(RULE(IRULE1+1),EQ.97.AND.SYMBOL(1),EQ.97) SYMBOL(1)=SYMIN(1) NODE(3,NODE1)=NODE2 DO 1 L=2,22 IF(RULE(IRULE1+L),EQ.-1) SYMBOL(L)=SYMIN(L)+1 IF(RULE(IRULE1+L),GT.9.AND.SYMBOL(L),EQ.RULE(IRULE1+L)) 1 SYMBOL(L)=SYMIN(L)

1 CONTINUE IF(SYMBOL(1),.NE.SYMIN(1)) RETURN IF(SYMBOL(3),.EQ.2.ASYMBOL(4),.NE.0) GO TO 2 RETURN

2 IF(SYMBOL(5),.EQ.1) SYMBOL(10)=3 IF(SYMBOL(10),.EQ.0) SYMBOL(10)=SYMIN(10) IF(SYMBOL(11),.EQ.0) SYMBOL(11)=SYMIN(11) IF(SYMBOL(12),.EQ.0) SYMBOL(12)=SYMIN(12) IF(SYMBOL(10),.LT.SYMIN(10)) SYMBOL(10)=SYMIN(10) IF(SYMBOL(11),.GT.SYMIN(11)) SYMBOL(11)=SYMIN(11) IF(SYMBOL(12),.GT.SYMIN(12)) SYMBOL(12)=SYMIN(12) RETURN

END
PROGRAM RULIST
DIMENSION A(26), B(6), TITLE(12), SYM(100)
DATA B/6H999.6H0.6H-4.6H1/ 
RULE=0.
IPAGE=0.
LINE=0.
READ(5,100) TITLE
100 FORMAT(12A6)
READ(5,105) SYM
105 FORMAT(10(2X,A3)) 
1 IF(LINE.GT.0) GO TO 2
IPAGE=IPAGE+1
WRITE(6,101) TITLE,IPAGE
101 FORMAT(1H1///10)012A6,10Xt5HPAGE...../ 
1 58H SYMBOL(MF/KBCLYDNGPRAVIT/S/-W/-X/Z) RULE NO. 
2 READ(5,102) I,(A(L),L=2,28) 
102 FORMAT(I3.2(2)0A1),1X,A2.15(2X,A1',1X,A2,4(2X,A1),2A3,2A1,2A2) 
IF(I.EQ.999) GO TO 9 
3 IF(A(25)-B(6)) 5,4,5 
4 WRITE(6,103) 
103 FORMAT(/) 
LINE=LINE+3 
K=4 
42 IF(A(25)-A(26)) 43,42,43 
43 CONTINUE 
ISUB=ISUB+1 
IF(RULE=A(24)) 41,10,41 
41 ISUB=1 
RULE=A(24) 
GO TO 10 
5 IF(A(25)-A(26)) 6,7,6 
6 K=4 
IF(A(25)-B(2)) 8,61,8 
61 K=5 
GO TO 8
K=3
WRITE(6,104) SYM(I),(A(L),L=2,22),A(27),A(28),B(K)
104 FORMAT(8X,A3,1H(,2A1,A2,13A1,A2,4A1,2A2,1H),1X,A1)
GO TO 11
WRITE(6,106) SYM(I),(A(L),L=2,22),A(27),A(28),B(K),A(24),ISUB
106 FORMAT(8X,A3,1H(,2A1,A2,13A1,A2,4A1,2A2,1H),1X,A1,11X,1H(A3,1H,,12,1H))
11 CONTINUE
   LINE=LINE+1
   IF(K.EQ.5.AND.LINE.GT.40) LINE=0
   GO TO 1
WRITE(6,109)
109 FORMAT(1H1,10(/),1X,17(7H-End--))
STOP
END
PART IV

APPENDIX C
SAMPLE OUTPUT
FROM SUBPROGRAM OUTPUT
<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MFKBCLYDNGPRATVISW</th>
<th>TY</th>
<th>RULE</th>
<th>EL</th>
<th>NE</th>
<th>REST</th>
<th>SUB</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>1000020000000000001</td>
<td>HAM*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>87</td>
<td>1300031000113000000</td>
<td>XYYM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>87</td>
<td>1000031000113000000</td>
<td>NxMN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>87</td>
<td>1000031000113000000</td>
<td>BalQ</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>94</td>
<td>100001000011301111112</td>
<td>HYH*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>87</td>
<td>100001100011300000012</td>
<td>SRR*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>77</td>
<td>10000100001100000003</td>
<td>GDL*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>89</td>
<td>1000010000000002000001</td>
<td>B***</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>85</td>
<td>10000100001130000001</td>
<td>AR**</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>87</td>
<td>10000310001130000000</td>
<td>YSRL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>10000500000000000001</td>
<td>?***</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>90</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>87</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>32</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>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>77</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>
</tr>
<tr>
<td>89</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>
</tr>
<tr>
<td>85</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>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
## RESULTS OF REWRITE PASS NO.- 3

| SYMBOL | M | F | K | B | C | L | Y | D | N | G | P | P | A | R | V | I | T | S | W | TY | RULE | EL | NE | REST | SUB | X |
| 90     | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | HAM* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6      | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0 | 9 | 0 | 1 | 0 | 0 | 0 |
| 33     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 1 | 1 | 1 | 12 | HYH* | 0 | 36 | 0 | 1 | 5 | 17 | 0 |
| 6      | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0 | 9 | 0 | 1 | 2 | 9 | 0 |
| 77     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | GDL* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 89     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | B*** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 85     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | AR&* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0 | 25 | 0 | 1 | 0 | 0 | 0 |
| 92     | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ??? | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SYMBOL | M | F | B | C | L | Y | D | N | G | P | R | A | V | I | T | S | W | TY | RULE | EL | NE | REST | SUB | X |
| 90     | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | HAM* 0 | 0 | 0 | 0 | 0 | 0 |
| 22     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 0 | 25 | 0 | 1 | 0 | 0 |
| 34     | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 1 | 1 | 12 | HYH* 0 | 37 | 0 | 1 | 11 | 17 | 0 |
| 6      | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 0 | 9 | 0 | 1 | 2 | 9 | 0 |
| 77     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | GDL* 0 | 0 | 0 | 0 | 0 | 0 |
| 69     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | B*** 0 | 0 | 0 | 0 | 0 | 0 |
| 23     | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 0 | 26 | 0 | 2 | 0 | 0 | 0 |
| 92     | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ??? 0 | 0 | 0 | 0 | 0 | 0 |
| SYMBOL | F | K | B | C | L | Y | D | N | G | P | R | A | V | I | T | S | W | T | Y | R | U | L | E | R | E | S | T | U | S | X |
|        | 0 | 26| 0 | 1 | 1 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|        | 0 | 38| 0 | 1 | 1 | 2 | 0 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 29| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

```
| HAM*  | 0000 |
| HYH*  | 0000 |
| GDL*  | 0000 |

```

```
| B*    | 0000 |
| **    | 0000 |
```

```
| 90    | 23  |
| 35    | 6   |
| 77    | 89  |
| 26    | 92  |

4-C-5
| SYMBOL | M | F | K | B | C | L | Y | D | N | G | P | R | A | V | I | T | S | W | TY | RULE | EL | NE | REST | SUB | X |
|    90  | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |   |    |   |    |    |   |   |
|    26  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |   | 0 |   | 29 |   |    |   |   |
|    36  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 1 | 1 | 1 |   | 0 |   | 39 |   | 2 |   |   |
|    6   | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 |   |   | 9 |   | 4 |   | 2 |   |
|    77  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |   |   |   |   | 4 |   | 2 |   |
|    38  | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |   |   | 4 |   | 2 |   | 0 |   |
|    92  | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |   |   |   |   |   | 4 |   | 2 |   |
| SYMBOL | M | F | K | B | C | L | Y | D | N | G | P | R | A | V | I | T | S | W | TY | RULE | EL | NE | REST | SUB | X |
|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|-----|-----|-----|-----|
| 90     | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1   | HAM*| 0   | 0   | 0   | 0   | 0   | 0 |
| 42     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0000| 0   | 45  | 0   | 1   | 0   | 0   |
| 36     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 1 | 1 | 1 | 12  | HYH*| 0   | 39  | 0   | 1   | 2   | 10  |
| 6      | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0000| 0   | 9   | 0   | 1   | 2   | 9   |
| 4      | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0000| 0   | 7   | 0   | 1   | 2   | 10  |
| 92     | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1   | *** | 0   | 0   | 0   | 0   | 0   | 0   |
## RESULTS OF REWRITE PASS NO. 9

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MFKBCLYNING</th>
<th>PRAVITS</th>
<th>WT</th>
<th>TY</th>
<th>RULE</th>
<th>EL</th>
<th>NE</th>
<th>REST</th>
<th>SUB</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>100020000000000000001</td>
<td>HAM*</td>
<td>00000000</td>
<td>04501000</td>
<td>00000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>1000100211300000000000</td>
<td>00000000</td>
<td>0</td>
<td>45</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>100010011130111112</td>
<td>HYH*</td>
<td>0390122100</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>1000100911300000000000</td>
<td>00000000</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>1000500000000000000000</td>
<td>?***</td>
<td>00000000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SYMBOL</td>
<td>M</td>
<td>F</td>
<td>K</td>
<td>B</td>
<td>C</td>
<td>L</td>
<td>Y</td>
<td>D</td>
<td>N</td>
<td>G</td>
</tr>
<tr>
<td>90</td>
<td>1</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>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>36</td>
<td>.1</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>3</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
RESULTS OF REWRITE PASS NO. 11

| SYMBOL | M | F | K | B | C | L | Y | D | N | G | P | R | A | V | I | T | S | W | TY | RULE | EL | NE | REST | SUB | X |
| 90     | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | HAM* | 0 | 0 | 0 | 0 | 0 | 0 |
| 42     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0 | 45 | 0 | 1 | 0 | 0 | 0 |
| 36     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 1 | 1 | 1 | 1 | 12 | HYH* | 0 | 39 | 0 | 1 | 2 | 10 | 0 |
| 26     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 9 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0 | 29 | 0 | 1 | 0 | 0 | 0 |
| 92     | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | ???* | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
## RESULTS OF REWRITE PASS NO.- 12

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>M</th>
<th>F</th>
<th>K</th>
<th>B</th>
<th>C</th>
<th>L</th>
<th>Y</th>
<th>D</th>
<th>N</th>
<th>G</th>
<th>P</th>
<th>R</th>
<th>A</th>
<th>V</th>
<th>I</th>
<th>T</th>
<th>S</th>
<th>W</th>
<th>TY</th>
<th>RULE</th>
<th>EL</th>
<th>NE</th>
<th>REST</th>
<th>SUB</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>1</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>1</td>
<td>HAM*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>45</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</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>1</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>HYH*</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>?***</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>SYMBOL</td>
<td>M</td>
<td>F</td>
<td>K</td>
<td>B</td>
<td>C</td>
<td>L</td>
<td>Y</td>
<td>D</td>
<td>N</td>
<td>G</td>
<td>P</td>
<td>R</td>
<td>A</td>
<td>V</td>
<td>I</td>
<td>T</td>
<td>S</td>
<td>W</td>
<td>TY</td>
<td>RULE</td>
<td>EL</td>
<td>NE</td>
<td>REST</td>
<td>SUB</td>
<td>X</td>
</tr>
<tr>
<td>90</td>
<td>1</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>1</td>
<td>HAM*</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>42</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>45</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</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>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>HYH*</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</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>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>53</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>???</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>SYMBOL</td>
<td>M</td>
<td>F</td>
<td>K</td>
<td>B</td>
<td>C</td>
<td>L</td>
<td>Y</td>
<td>D</td>
<td>N</td>
<td>G</td>
<td>P</td>
<td>R</td>
<td>A</td>
<td>V</td>
<td>I</td>
<td>T</td>
<td>S</td>
<td>W</td>
<td>T</td>
<td>Y</td>
<td>RULE</td>
<td>EL</td>
<td>NE</td>
<td>REST</td>
<td>SUB</td>
</tr>
<tr>
<td>90</td>
<td>1</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>1</td>
<td>HAM*</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>42</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>45</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>53</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>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>56</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>?***</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>
</tbody>
</table>
## RESULTS OF REWRITE PASS NO. - 15

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>M</th>
<th>F</th>
<th>K</th>
<th>B</th>
<th>C</th>
<th>L</th>
<th>Y</th>
<th>D</th>
<th>N</th>
<th>G</th>
<th>P</th>
<th>R</th>
<th>A</th>
<th>V</th>
<th>I</th>
<th>T</th>
<th>S</th>
<th>W</th>
<th>TY</th>
<th>RULE</th>
<th>EL</th>
<th>NE</th>
<th>REST</th>
<th>SUB</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>1</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>HAM*</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>62</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>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>65</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>?**</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>
### RESULTS OF REWRITE PASS NO. - 16

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MFKBCLYDNGPRAVITS</th>
<th>W</th>
<th>TY</th>
<th>RULE</th>
<th>EL</th>
<th>NE</th>
<th>REST</th>
<th>SUB</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>1 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 1</td>
<td>HAM*</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>64</td>
<td>1 0 0 0 3 0 0 0 0 1 1 1 3 0 0 1 1 1 0</td>
<td>0000</td>
<td>0</td>
<td>67</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>1 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 1</td>
<td>???</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>

---

4-C-16

209
<p>| SYMBOL | M | F | K | B | C | L | Y | D | N | G | P | R | A | V | I | T | S | W | TY | RULE | EL | NE | REST | SUB | X |
| 71     | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 0 | 0 | 0 | 0 | 0 | 0 |
| 92     | 1 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 *** | 0 | 0 | 0 | 0 | 0 | 0 |</p>
<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>M</th>
<th>F</th>
<th>K</th>
<th>B</th>
<th>C</th>
<th>L</th>
<th>Y</th>
<th>D</th>
<th>N</th>
<th>G</th>
<th>P</th>
<th>R</th>
<th>A</th>
<th>V</th>
<th>I</th>
<th>T</th>
<th>S</th>
<th>W</th>
<th>TY</th>
<th>RULF</th>
<th>EL</th>
<th>NE</th>
<th>REST</th>
<th>SUB</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</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>0000</td>
<td>0</td>
<td>78</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>1</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>?***</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>

RESULTS OF REWRITE PASS NO.- 18

4-C-18
RESULTS OF REWRITE PASS NO.- 19

| 76     | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

79 0 2 0 0 0 0
Appendix

PART IV

APPENDIX D

Examples of Exhaustive Syntactic Analysis by Computer (With Associated Tree Diagrams). Tree diagrams are produced by subprogram DIAGRM; syntax analysis statements are produced by subprogram PARSE.
SYNTAX ANALYSIS IF SENTENCE NO. 101.

( 1 ) THE COMPLETED SENTENCE(SC) IS AN INTEROGATIVE SENTENCE.

( 2 ) THE INTEROGATIVE SENTENCE(SI) OF THE COMPLETED SENTENCE(SC) HAS NO DEPENDENT CLAUSE.

( 3 ) THE INTEROGATIVE CLAUSE(KI) OF THE INTEROGATIVE SENTENCE(SI) QUESTIONS TRUTH/CIRCUMSTANCES OF SN.

( 4 ) THE INDEPENDENT CLAUSE(SA) OF THE INTEROGATIVE CLAUSE(KI) OF THE INTEROGATIVE SENTENCE(SI) IS A DEFINITE INDEPENDENT CLAUSE.

( 5 ) THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE INTEROGATIVE CLAUSE(KI) OF THE INTEROGATIVE SENTENCE(SI) HAS A NAMED SUBJECT.

( 6 ) THE SUBJECT PHRASE(NSP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE INTEROGATIVE CLAUSE(KI) OF THE INTEROGATIVE SENTENCE(SI) IS A NOUN PHRASE.

( 7 ) THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE INTEROGATIVE CLAUSE(KI) OF THE INTEROGATIVE SENTENCE(SI) IS A VERB PHRASE + VRB.MODIF.PHRASE.


( 9 ) THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE INTEROGATIVE CLAUSE(KI) OF THE INTEROGATIVE SENTENCE(SI) IS A COPULATIVE PHRASE.


(12) The regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the interrogative clause (KI) of the interrogative sentence (SI) contains no construct nouns.

(13) The simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the interrogative clause (KI) of the interrogative sentence (SI) has a basic noun phrase as nucleus.

(14) The basic noun phrase (NA) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the interrogative clause (KI) of the interrogative sentence (SI) has a non-determinate noun.

(15) The post-nominal adjective phrase (AP) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the interrogative clause (KI) of the interrogative sentence (SI) is a basic post-nominal adj. phrase.

(16) The basic post-nominal adjective phrase (APA) of the post-nominal adjective phrase (AP) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the interrogative clause (KI) of the interrogative sentence (SI) expresses the comparative degree.

(17) The general noun phrase (NP) of the subject phrase (NSP) of the definite independent clause (SAB) of the independent clause (SA) of the interrogative clause (KI) of the interrogative sentence (SI) is a reg. noun phrase + (app. noun. phrase).

(18) The preposition phrase (XP) of the basic post-nominal adjective phrase (APA) of the post-nominal adjective phrase (AP) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the interrogative clause (KI) of the interrogative sentence (SI) governs a noun phrase.

(19) The regular noun phrase (NPA) of the general noun phrase
E(NP) of the subject phrase(NSP) of the definite independent clause(SAB) of the interrogative clause(KI) of the interrogative sentence(SI) contains no construct nouns.

(20) The seven-tense verb phrase(VAA) of the verb phrase(VA) of the predicate phrase(VP) of the definite independent clause(SAB) of the interrogative clause(KI) of the interrogative sentence(SI) is an emphatic verb phrase.

(21) The simple noun phrase(NPB) of the regular noun phrase(NPA) of the general noun phrase(NP) of the subject phrase(NSP) of the definite independent clause(SAB) of the interrogative clause(KI) of the interrogative sentence(SI) has a basic noun phrase as nucleus.

(22) The emphatic verb phrase(VC) of the seven-tense verb phrase(VAA) of the verb phrase(VA) of the predicate phrase(VP) of the definite independent clause(SAB) of the interrogative clause(KI) of the interrogative sentence(SI) expresses no special emphasis.

(23) The regular noun phrase(NPA) of the general noun phrase(NP) of the subject phrase(NSP) of the definite independent clause(SAB) of the interrogative clause(KI) of the interrogative sentence(SI) has a (determinate) proper noun.

(24) The basic noun phrase(NA) of the simple noun phrase(NPB) of the regular noun phrase(NPA) of the general noun phrase(NP) of the subject phrase(NSP) of the definite independent clause(SAB) of the interrogative clause(KI) of the interrogative sentence(SI) contains a (determinate) proper noun.

(25) The three-tense verb phrase(VBB) of the emphatic verb phrase(VC) of the seven-tense verb phrase(VAA) of the verb phrase(VA) of the predicate phrase(VP) of the definite independent clause(SAB) of the interrogative clause(KI) of the interrogative sentence(SI) is a verb or participle.

(26) The verb mood phrase(VB) of the three-tense verb phrase(VBB) of the emphatic verb phrase(VC) of the seven-tense verb phrase(VAA) of the verb phrase(VA) of the predicate phrase(VP) of the definite independent clause(SAB) of the interrogative clause(KI) of the interrogative sentence(SI) is a verb or infinitive absolute.

SING., MASC., THIRD, ACT., IND., PAST.

SING., MASC., THIRD.

SING., MASC., THIRD.

ANALYSIS COMPLETED FOR SENTENCE NO. 101, NO. OF SYMBOL TESTS= 265.
SYNTAX ANALYSIS IF SENTENCE NO. 102.

(1) THE COMPLETED SENTENCE(SC) IS AN IMPERATIVE SENTENCE.

(2) THE BASIC SENTENCE(S) OF THE COMPLETED SENTENCE(SC) IS A SIMPLE SENTENCE.

(3) THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A DEFINITE INDEPENDENT CLAUSE.

(4) THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) HAS A NAMED SUBJECT.

(5) THE SUBJECT PHRASE(NSP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A NOUN PHRASE.

(6) THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A VERB PHRASE + VERB.MODIF.PHRASE.

(7) THE VERB PHRASE(VA) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A SEVEN-TENSE VERB PHRASE.

(8) THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A COPULATIVE PHRASE.

(9) THE COPULATIVE PHRASE(NPX) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A NOUN PHRASE.


(11) THE REGULAR NOUN PHRASE(NPA) OF THE GENERAL NOUN PHRASE(NP) OF THE COPULATIVE PHRASE(NPX) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) CONTAINS NO CONSTRUCT NOUNS.

(12) THE SIMPLE NOUN PHRASE(NPB) OF THE REGULAR NOUN PHRASE(NPA) OF THE GENERAL NOUN PHRASE(NP) OF THE COPULATIVE PHRASE(NPX) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) CONTAINS NO CONSTRUCT NOUNS.
(13) **The basic noun phrase** (NA) of the **simple noun phrase** (NPB) of the **regular noun phrase** (NPA) of the **general noun phrase** (NP) of the **copulative phrase** (NPX) of the **verb modifying phrase** (VM) of the **predicate phrase** (VP) of the **definite independent clause** (SAB) of the **independent clause** (SA) of the **basic sentence** (S) has a nondeterminate noun.

(14) **The post-nominal adjective phrase** (AP) of the **simple noun phrase** (NPB) of the **regular noun phrase** (NPA) of the **general noun phrase** (NP) of the **copulative phrase** (NPX) of the **verb modifying phrase** (VM) of the **predicate phrase** (VP) of the **definite independent clause** (SAB) of the **independent clause** (SA) of the **basic sentence** (S) is a **basic post-nominal adj. phrase**.

(15) **The basic post-nominal adjective phrase** (APA) of the **post-nominal adjective phrase** (AP) of the **simple noun phrase** (NPB) of the **regular noun phrase** (NPA) of the **general noun phrase** (NP) of the **copulative phrase** (NPX) of the **verb modifying phrase** (VM) of the **predicate phrase** (VP) of the **definite independent clause** (SAB) of the **independent clause** (SA) of the **basic sentence** (S) expresses the comparative degree.

(16) **The general noun phrase** (NP) of the **subject phrase** (NSP) of the **definite independent clause** (SAB) of the **independent clause** (SA) of the **basic sentence** (S) is a **reg. noun phrase** *(app. noun. phrs.)*.

(17) **The preposition phrase** (XP) of the **basic post-nominal adjective phrase** (APA) of the **post-nominal adjective phrase** (AP) of the **simple noun phrase** (NPB) of the **regular noun phrase** (NPA) of the **general noun phrase** (NP) of the **copulative phrase** (NPX) of the **verb modifying phrase** (VM) of the **predicate phrase** (VP) of the **definite independent clause** (SAB) of the **independent clause** (SA) of the **basic sentence** (S) governs a noun phrase.

(18) **The regular noun phrase** (NPA) of the **general noun phrase** (NP) of the **subject phrase** (NSP) of the **definite independent clause** (SAB) of the **independent clause** (SA) of the **basic sentence** (S) contains no construct nouns.

(19) **The seven-tense verb phrase** (VAA) of the **verb phrase** (VA) of the **predicate phrase** (VP) of the **definite independent clause** (SAB) of the **independent clause** (SA) of the **basic sentence** (S) is an emphatic verb phrase.

(20) **The simple noun phrase** (NPB) of the **regular noun phrase** (NPA) of the **general noun phrase** (NP) of the **subject phrase** (NSP) of the **definite independent clause** (SAB) of the **independent clause** (SA) of the **basic sentence** (S) has a basic noun phrase as nucleus.
(21) The emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence (S) expresses no special emphasis.

(22) The regular noun phrase (NPA) of the general noun phrase (NP) of the preposition phrase (XP) of the basic post-nominal adjective phrase (APA) of the post-nominal adjective phrase (AP) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence (S) contains construct noun(s).

(23) The basic noun phrase (NA) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the subject phrase (NSP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence (S) has a (determinate) proper noun.

(24) The three-tense verb phrase (VBB) of the emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence (S) is a verb or participle.

(25) The verb mood phrase (VB) of the three-tense verb phrase (VBB) of the emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence (S) is a verb or infinitive absolute.

(26) The basic noun phrase (NA) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the preposition phrase (XP) of the basic post-nominal adjective phrase (APA) of the post-nominal adjective phrase (AP) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence (S) has a (determinate) proper noun.

(27) The verb (V) of the verb mood phrase (VB) of the three-tense verb phrase (VBB) of the emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence (S) is.

(28) The noun absolute (N) of the basic noun phrase (NA)
THE INDEPENDENT CLAUSE (SA) OF THE BASIC SENTENCE (S) IS POET.
SING., MASC., SECD.


SING., MASC., THIRD.

YSIS COMPLETED FOR SENTENCE NO. 102, NO. OF SYMBOL TESTS = 261.

4-D-10

223
SYNTAX ANALYSIS IF SENTENCE NO. 103.

(1) THE COMPLETED SENTENCE(SC) IS A DECLARATIVE SENTENCE.

(2) THE BASIC SENTENCE(S) OF THE COMPLETED SENTENCE(SC) IS A SIMPLE SENTENCE.

(3) THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A DEFINITE INDEPENDENT CLAUSE.

(4) THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) HAS A NAMED SUBJECT.

(5) THE SUBJECT PHRASE(NSP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A NOUN PHRASE.

(6) THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A VERB PHRASE + VERB.MODIF.PHRASE.

(7) THE VERB PHRASE(VA) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A SEVEN-TENSE VERB PHRASE.

(8) THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A COPULATIVE PHRASE.

(9) THE COPULATIVE PHRASE(NPX) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A NOUN PHRASE.

(10) THE GENERAL NOUN PHRASE(NP) OF THE COPULATIVE PHRASE(NPX) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A REGULAR NOUN PHRASE + [APP.NOUN.PHRASE].

(11) THE REGULAR NOUN PHRASE(NPA) OF THE GENERAL NOUN PHRASE(NP) OF THE COPULATIVE PHRASE(NPX) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) CONTAINS NO CONSTRUCT NOUNS.

(12) THE SIMPLE NOUN PHRASE(NPB) OF THE REGULAR NOUN PHRASE(NPA) OF THE GENERAL NOUN PHRASE(NP) OF THE COPULATIVE PHRASE(NPX) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) CONTAINS NO CONSTRUCT NOUNS.
ATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) HAS A BASIC NOUN PHRASE AS NUCLEUS.


(16) THE GENERAL NOUN PHRASE(NP) OF THE SUBJECT PHRASE(NSP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A REG. NOUN PHRASE.


(18) THE REGULAR NOUN PHRASE(NPA) OF THE GENERAL NOUN PHRASE(NP) OF THE SUBJECT PHRASE(NSP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) CONTAINS NO CONSTRUCT NOUNS.

(19) THE SEVEN-TENSE VERB PHRASE(VAA) OF THE VERB PHRASE(VA) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS AN EMPHATIC VERB PHRASE.

(20) THE SIMPLE NOUN PHRASE(NPB) OF THE REGULAR NOUN PHRASE(NPA) OF THE SUBJECT PHRASE(NSP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) HAS A BASIC NOUN PHRASE AS NUCLEUS.
(21) The emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the basic sentence(s) expresses no special emphasis.

(22) The regular noun phrase (NPA) of the general noun phrase (NP) of the preposition phrase (XP) of the basic post-nominal adjective phrase (APA) of the post-nominal adjective phrase (AP) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence(s) expresses no special emphasis.

(23) The basic noun phrase (NA) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the subject phrase (NSP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence(s) has a (determinate) proper noun.

(24) The three-tense verb phrase (VBB) of the emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence(s) is a verb or participle.

(25) The verb mood phrase (VB) of the three-tense verb phrase (VBB) of the emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence(s) is a verb or infinitive absolute.

(26) The basic noun phrase (NA) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the preposition phrase (XP) of the basic post-nominal adjective phrase (APA) of the post-nominal adjective phrase (AP) of the simple noun phrase (NPB) of the regular noun phrase (NPA) of the general noun phrase (NP) of the copulative phrase (NPX) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence(s) has a (determinate) proper noun.

(27) The verb (V) of the verb mood phrase (VB) of the three-tense verb phrase (VBB) of the emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence(s) is is.

(28) The noun absolute (N) of the basic noun phrase (NA)

SING., MASC., THIRD.


SING., MASC., THIRD.

ANALYSIS COMPLETED FOR SENTENCE NO. 103, NO. OF SYMBOL TESTS = 263.
HEBREW SENTENCE ANALYZED--
HWA YSB BKPR QTN.

TREE DIAGRAM OF HEBREW SENTENCE NO. 4

R-2  V-1  P-1  N-1  A-1  T-6
I   I   I   I   I   I
I   I   I   I   I   I
I   I   I   I   I   I
RSP1 VB1 I   I   APA1 I
I   I   I   I   I   I
I   I   I   I   I   I
I   I   I   I   I   I
I   VBB1 I   NA1 AP1 I
I   I   I   I   I   I
I   I   I   I   I   I
I   VC3 I   NPB1 I
I   I   I   I   I   I
I   I   I   I   I   I
I   I   I   I   I   I
I   VAA1 I   NPA1 I
I   I   I   I   I   I
I   I   I   I   I   I
I   I   I   I   I   I
I   I   P-1 NP1 I
I   I   I   I   I   I
I   I   I   I   I   I
I   I   I   I   I   I
I   I   XP1 I   I   I
I   I   I   I   I   I
I   I   I   I   I   I
I   I   VA1 VM5 I
I   I   I   I   I   I
I   I   I   I   I   I
I   NSP2 VP1 I   I
I   I   I   I   I   I
I   SAB1 I   I   I
I   I   I   I   I   I
I   SA3 I   I   I
I   I   I   I   I   I
I   S-1 I   I   I
I   T-6 I   I   I

---

I  SC1  82S  229
4-D-16
SYNTAX ANALYSIS IF SENTENCE NO. 4.

(1) THE COMPLETED SENTENCE(SC) IS A DECLARATIVE SENTENCE.

(2) THE BASIC SENTENCE(S) OF THE COMPLETED SENTENCE(SC) IS A SIMPLE SENTENCE.

(3) THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A DEFINITE INDEPENDENT CLAUSE.

(4) THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) HAS A NAMED SUBJECT.

(5) THE SUBJECT PHRASE(NSP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A SUBJECT PRONOUN PHRASE.

(6) THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A VERB PHRASE + VERB.MODIF.PHRASE.

(7) THE VERB PHRASE(VA) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A SEVEN-TENSE VERB PHRASE.

(8) THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A PREPOSITIONAL PHRASE.

(9) THE PREPOSITION PHRASE(XP) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) GOVERNS A NOUN PHRASE.

(10) THE GENERAL NOUN PHRASE(NP) OF THE PREPOSITION PHRASE(XP) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A REG.NOUN.PHRASE.+{APP.NOUN.PHRSE}.

(11) THE SEVEN-TENSE VERB PHRASE(VAA) OF THE VERB PHRASE(VA) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS AN EMPHATIC VERB PHRASE.

(12) THE REGULAR NOUN PHRASE(NPA) OF THE GENERAL NOUN PHRASE(NP) OF THE PREPOSITION PHRASE(XP) OF THE VERB MODIFYING PHRASE(VM) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A NOUN PHRASE.
THE BASIC SENTENCE(S) CONTAINS NO CONSTRUCT NOUNS.

(13) THE EMPHATIC VERB PHRASE(VC) OF THE SEVENTENSE VERB PHRASE(VAA) OF THE VERB PHRASE(VA) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) EXPRESS NO SPECIAL EMPHASIS.


(15) THE SEVEN-TENSE VERB PHRASE(VBB) OF THE EMPHATIC VERB PHRASE(VC) OF THE SEVENTENSE VERB PHRASE(VAA) OF THE VERB PHRASE(VA) OF THE PREDICATE PHRASE(VP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A VERB OR PARTICIPLE.


(18) THE SUBJECT PRONOUN PHRASE(RSP) OF THE SUBJECT PHRASE(NSP) OF THE DEFINITE INDEPENDENT CLAUSE(SAB) OF THE INDEPENDENT CLAUSE(SA) OF THE BASIC SENTENCE(S) IS A SUBJECT PRONOUN + (APPPOS. N. PH.).


21) The pronoun (r) of the subject pronoun phrase (RSP) of the subject phrase (NSP) of the definite independent clause (SAB) of the independent clause(s) of the basic sentence(s) is he. Sing., masc., third.

22) The verb (V) of the verb mood phrase (VB) of the three-tense verb phrase (VBB) of the emphatic verb phrase (VC) of the seven-tense verb phrase (VAA) of the verb phrase (VA) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause(s) of the basic sentence(s) is dwell. Sing., masc., third, r=2, a=5, act., ind., past.

23) The noun absolute (N) of the basic noun phrase (NA) of the simple noun phrase (NPR) of the regular noun phrase (NPA) of the general noun phrase (NP) of the preposition phrase (XP) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence(s) is village. Sing., masc., third.

24) The adjective (A) of the basic post-nom. adjective phrase (APA) of the post-nominal adjective phrase (AP) of the simple noun phrase (NPR) of the regular noun phrase (NPA) of the general noun phrase (NP) of the preposition phrase (XP) of the verb modifying phrase (VM) of the predicate phrase (VP) of the definite independent clause (SAB) of the independent clause (SA) of the basic sentence(s) is small. Sing., masc.

Analysis completed for sentence no. 4, no. of symbol tests: 196.