

Print
655.38
M
C. J.

IAB 1952

PRINT
655.38
M
c.1
Mitre Corporation.
DOTSYS II USER'S GUIDE.

3-
Re
PRINT
655.38
M
c. 1
Mitre Corporation
DOTSYS II USER'S GUIDE

**PROPERTY OF
IOWA STATE COMMISSION
FOR THE BLIND**

see the
MITRE Corp

Print
6233
11

MITRE Technical Report

MTR- 1853 _____ 1
No. Vol. Series Rev. Supp. Corr.

Subject: DOTSYS II User's Guide and Transfer and Maintenance Manual

Author: Dr. J. K. Millen

Dept.: D-73

Date: 24 August 1970

Contract No. SR 21697

Contract Sponsor: Sensory Aids Evaluation and Development Center, Massachusetts Institute of Technology

Project: 1248

Issued at: Bedford, Massachusetts

Department Approval:

[Handwritten signature]
[Handwritten signature]

MITRE Project Approval:

R. A. J. Gildea



18WA 005.13. 24646 1970

PC 400, 79

ERRATA

- Page i: Subject: Insert "and" between "Transfer" and "Maintenance".
- Page 2: First paragraph, sixth line: Delete the sentence "The output of DOTSYS II is the braille equivalent of the input text".
- Page 3: Fifth paragraph ("PUNCHED OUTPUT"), fourth line: Replace "Part 1" by "[1]".
- Page 22: First paragraph, fourth line: Replace "[1]" by "[4]".
- Page 25: Second paragraph, eighth line: Replace "Part 1" by "[1]".
- Page 31: Second paragraph, third line: Replace "received" by "reserved".
- Page 34: Footnote ****: Replace "NDTC" by "(NDTC)", and replace "(NSV + NIC)" by "(NSV) + (NIC)". (A parenthesized data name denotes the contents of the data item named.)
- Page 36: Add as the first line: "00000 IDENTIFICATION DIVISION!".
- Page 47: Line 00530: Replace asterisks (*) by spaces.
- Page 48: Line 00590: Replace asterisks (*) by spaces.
- Page 53: ALLY entry should be: "ALLY\$ 04 04 32619999".

DISTRIBUTION LIST

INTERNAL

C-01

C. W. Farr

D-06

P. R. Vance

D-07

J. H. Burrows

A. J. Roberts

D-11

J. J. Croke

C. E. Duke

J. F. Jacobs

D-12

C. A. Zraket

D-53

F. Engel

D-63

R. S. Nielsen

D-73

W. Amory

N. A. Anschuetz

E. H. Bensley

J. A. Clapp

T. L. Connors

C. G. Crothers

M. T. Gattozzi

W. R. Gerhart

R. A. J. Gildea (25)

J. B. Glore

O. R. Kinney

E. L. Lafferty

Dr. J. K. Millen (25)

D-73 (cont.)

J. Mitchell

C. M. Sheehan

J. E. Sullivan

N. B. Sutherland

L. M. Thomas

Dr. D. E. Walker

E. W. Williamson

PROJECT

G. Dalrymple, M.I.T.

M. Leonard, M.I.T.

Prof. R. W. Mann, M.I.T.

V. A. Proscia, M.I.T. (50)

EXTERNAL

H. Bassler

Colonial Penn Insurance Co.

112 South 16th Street

Philadelphia, Pennsylvania 19102

Dr. M. P. Boyles (3)

Director, Computer-Braille Project

Instructional Services Center

Atlanta, Georgia 30315

L. L. Clark (2)

Director, IRIS

American Foundation for the Blind

15 West 16th Street

New York, New York 10011

E. L. Glaser

Computation Center

Case Western Reserve University

University Circle

Cleveland, Ohio 44106

DISTRIBUTION LIST (cont.)

EXTERNAL

Dr. C. E. Hallenbeck
Department of Psychology
University of Kansas
Lawrence, Kansas 66044

R. Haynes
American Printing House for
the Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

Dr. K. R. Ingham
Room 20-B-207
Massachusetts Institute
of Technology
77 Massachusetts Avenue
Cambridge, Massachusetts
02139

R. E. LaGrone
IBM Corporation
Federal Systems Division
Department PC4, Room 2P25
18100 Frederick Pike
Gaithersburg, Maryland 20760

Dr. L. Leffler
Applied Mathematics Division
Argonne National Laboratories
9700 South Cass Avenue
Argonne, Illinois 60440

R. J. McNaughton
RCA
Aerospace Systems Division
Burlington, Massachusetts
01801

Prof. A. Nemeth
Mathematics Department
University of Detroit
4001 W. McNichols
Detroit, Michigan 48821

Dr. B. Perella
Department of Defense
Fort George Meade, Maryland

A. Schack
Schack Associates
127 West 12th Street
New York, New York 10011

J. Siems
American Printing House for the Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

Dr. E. J. Waterhouse
Director
Perkins School for the Blind
175 North Beacon Street
Watertown, Massachusetts 02172

V. Zickle
American Printing House for the Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

FOREIGN

P. W. F. Coleman
4 George Street
Eastleigh, Hants
SO5 4 BU, United Kingdom

C. W. Garland
Royal National Institute for
the Blind
224-6 Great Portland Street
London W. 1, England

R. House
Research Group for Quantitative
Linguistics
Sodermalmstorg 8, Stockholm
Sweden

DISTRIBUTION LIST (conc.)

FOREIGN

Dr. G. Lamprecht
Wistfalische Wilhelms Universitat
44 Munster, Germany
Roxeler Strasse 64

B. Lindqvist
De Blindas Forening
Utvechlingsavdelningen
Gotlandsgatan 46
116 65 Stockholm
Sweden

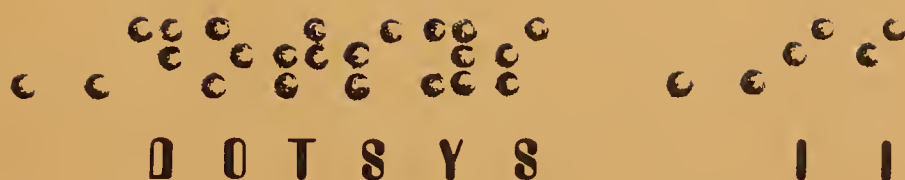
J. Lindstrom
Handikappinstitutet
Pa Iach, 161 03 Bromma 3
Sweden

V. Mogleby
Husby OFF Skole for Blinde
Hovseterveien 3
Oslo 7
Norway

W. Sorke
Deutsche Blindenstudienanstalt
355 Marburg/Lahn
Am Schlag 8
West Germany

J. Vinding
Statens Institut for Blinde
og Svagsynede
Rymarksvej 1
2900 Hellerup
Denmark

Prof. H. Werner
Institut fur Numerische und
Instrumentelle Mathematik
Universitat Munster
Schlossplatz 5
44 Munster, West Germany



user's guide and transfer and maintenance manual

by j. k. millen

july 1970

the MITRE corporation

Print
655.38
M

MITRE Technical Report

MTR- 1853
No. Vol. Series Rev. Supp. Corr.

Subject: DOTSYS II User's Guide and Transfer
Maintenance Manual

Author: Dr. J. K. Millen

Dept.: D-73

Date: 2 July 1970

Contract No : SR 21697

Contract Sponsor: Sensory Aids Evaluation and Development
Center, Massachusetts Institute of
Technology

Project: 1248

Issued at: Bedford, Massachusetts

Department Approval: Edward H. Bandy

MITRE Project Approval: R. A. J. Gildea
R. A. J. Gildea



ABSTRACT

DOTSYS II is a COBOL program to translate natural language into braille. It is table-driven, and has been supplied with tables which equip it for use on English text. This document has two parts: Part 1, the User's Guide, and Part 2, the Transfer and Maintenance Manual. For the purposes of Part 1, it is assumed that the user has a program deck lacking only DOTSYS II input, and knows how to submit a computer run. Part 1 explains how to prepare the control cards and input text, and how to interpret the output. Part 2 explains when and how to make changes in the tabular input and in the program itself that may be made necessary by transfer of DOTSYS II, a change in peripheral units, a change in braille translation rules, or merely a desire to improve the conformity of the translation with the braille ideal.

TABLE OF CONTENTS

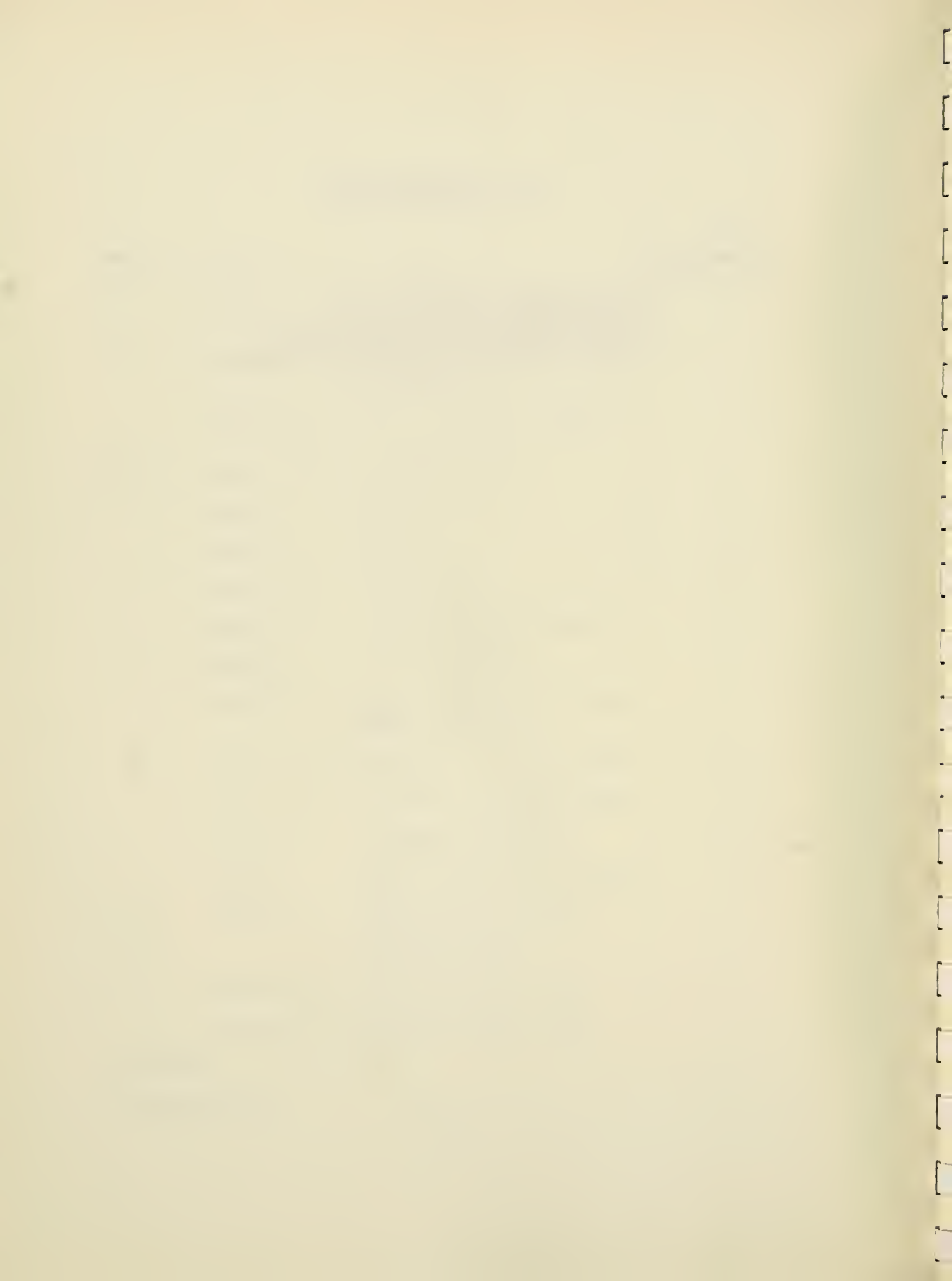
	<u>Page</u>
LIST OF ILLUSTRATIONS	vii
PART 1 USER'S GUIDE	1
SECTION I INTRODUCTION	2
SECTION II OUTPUT	3
PROOF OUTPUT	3
BRAILLE OUTPUT	3
PUNCHED OUTPUT	3
SECTION III CONTROL CARDS	5
SECTION IV KEYPUNCH INSTRUCTIONS	7
INTRODUCTION	7
CARD BOUNDARIES	7
CAPITALIZATION	7
ITALICS	7
CHARACTERS REQUIRING SPECIAL SYMBOLS	8
Mathematical Symbols	8
Quotation Marks	8
Accent Marks	8
Brackets	8
FORMAT CONTROLS	9
Paragraphs	9
New Line	9
New Page	9
Tabulation	9
SECTION V NOTES TO THE BRAILLE EDITOR	10
Letter Sign (=)	10
Grade Switch (\$G)	10
Division Symbol (\$/)	11
Termination Sign (\$T)	11

TABLE OF CONTENTS (Cont.)

	<u>Page</u>
SECTION VI INPUT ERRORS	12
APPENDIX I IBM OPERATING SYSTEM/360 PROGRAM DECK COMPOSITION	13
APPENDIX II SAMPLE INPUT DECK AND PROOF OUTPUT FROM IT	14
APPENDIX III SUMMARY OF SPECIAL SYMBOLS	19
PART 2 TRANSFER AND MAINTENANCE MANUAL	20
SECTION I INTRODUCTION	21
SECTION II TRANSFER	22
SECTION III OUTPUT DEVICES	24
SECTION IV CONTRACTION TABLE ADDITIONS	25
SECTION V TABLE SIZE BOUNDS	26
APPENDIX I REPLACING THE ALPHABET TABLE SEARCH BY INDEXING	27
APPENDIX II DECK SETUP OF TABULAR INPUT	29
APPENDIX III FLOW CHART OF OUTPUT SECTION	30
APPENDIX IV GLOSSARY OF DATA NAMES	31
APPENDIX V FORMAT OF TABULAR INPUT CARDS	34
APPENDIX VI CURRENT STATE VARIABLES AND INPUT CLASSES	35
APPENDIX VII DOTSYS II SOURCE LISTING	36
APPENDIX VIII LISTING OF TABULAR INPUT	51
REFERENCES	61
DISTRIBUTION LIST	63

LIST OF ILLUSTRATIONS

<u>Figure Number</u>		<u>Page</u>
1	Sample of Proof Output	4
2	Braille Output for Line in Figure 1	4
3	Punched Card Output for Line in Figure 1	4
4	Suggested Number of Braille Lines Per Page	6



PART 1
USER'S GUIDE

SECTION I

INTRODUCTION

DOTSYS II is a computer program to translate natural language text into braille. It is table-driven, and has been supplied with tables which equip it for use on English text. The input to DOTSYS II is in the form of punched (Hollerith) cards, or card images on another medium, comprising (1) control cards, and (2) English input text. The output of DOTSYS II is the braille equivalent of the input text. The output of DOTSYS II is the braille equivalent of the input text, in a form specified by the control cards. Depending on the control cards, the output may be either printed or punched, or both. If printed, the output may be either in proof form or in a form suitable for production of tactile braille in conjunction with a modification of the printer. The punched output is suitable for storage of the braille output and may later be transformed into tactile braille with another program.

For the purposes of this document, it is assumed that the user has a program deck¹ lacking only DOTSYS II input, and knows how to submit a computer run. This document explains how to prepare the control cards and input text, and how to interpret the output. A sample of control cards and input text with its consequent output is given in Appendix II.

¹Appendix I specifies the contents of the program deck in an IBM Operating System/360 environment.

SECTION II

OUTPUT

PROOF OUTPUT

When proof output is called for, the first three pages of output display the contents of internal tables, of interest only when contemplating changes in the translation capability. The purpose of these tables is given in Part 1; their modification is discussed in Part 2, and is not likely to be of immediate concern to the user.

Each sign in the braille translation of the input text is printed using periods for braille dots. Below the braille sign are up to three characters intended to help identify the sign. If the sign represents a letter, for example, the letter itself is printed below the sign. The identification characters depend only on the sign and not its context. For example, the sign for "K" has the identification character "K" even when it represents the word "knowledge." Below the identification characters is printed a numerical code in the range 0 through 63, which may be used to check the punched output.

In Figure 1 is a sample of proof output. Note that there is no hyphenation; lines are broken only at spaces.

BRAILLE OUTPUT

Braille output is intended for production on a line printer modified in one of several ways described in Part 2, to produce tactile braille. It includes only the braille signs. Figure 2 shows a line of braille output.

PUNCHED OUTPUT

At the option of the user, punched output may be produced either alone or in addition to one of the two forms of printed output. Braille signs are coded on punched cards as two-digit numbers in the range 00 through 63, according to a correspondence defined in Part 1. Forty 2-digit numbers are punched on each card; however, after the end of what would have been a printed line, only 00's are punched. Figure 3 pictures a card on which is coded the same braille signs as in Figure 1.

```

      . . . . . . . . . . . . . . . .
      . . . . . . . . . . . . . . . .
      . . . . . . . . . . . . . . . .
      7 S A M P L E      B R L      L I N E
     32 14 01 13 15 07 17 00 03 23 07 00 07 20 17 00 00 00 00 00

```

Figure 1. Sample of Proof Output

```

      . . . . . . . . . . . . . . . .
      . . . . . . . . . . . . . . . .
      . . . . . . . . . . . . . . . .

```

Figure 2. Braille Output for Line in Figure 1

```

321401131507170003230700072017000000000000000000000000000000000000

```

Figure 3. Punched Card Output for Line in Figure 1

SECTION III

CONTROL CARDS

The first four cards of input are control cards, on which the user specifies certain parameters of the output format. Any information on the control cards in addition to the required information specified in the following paragraphs is ignored by DOTSYS II.

1. The first card specifies whether proof output or braille output will be produced. If the letter in column 1 is P, then the output will be in proof form. If the letter in column 1 is B, then the output will be in braille form.

2. The second card specifies whether punched output will be produced. If the letter in column 1 is Y (standing for YES), then there will be punched output. If the letter in column 1 is N (standing for NO), then there will be no punched output. If the letter in column 1 is O (standing for ONLY), then only punched output will be produced, and the character in column 1 on the first card is immaterial (although the first card must still be present).

3. The third card specifies the maximum number of braille signs permitted in an output line. This number must be in the range 2 through 40 and is punched in columns 18 and 19, with a leading zero if 9 or less.

4. The fourth card specifies the number of lines printed on each page. This number must be in the range 1 through 99 and is punched in columns 18 and 19, with a leading zero if 9 or less. A number must be put on this card even if only punched output is specified (on the second card), despite the fact that page boundaries are not indicated on punched output. Figure 4 gives suggested values for 11-inch paper.

	Printer Setting Lines/Inch		
	6	8	10
Proof Output	10	13	16
Braille Output	15	20	25

Figure 4. Suggested Number of Braille Lines Per Page

SECTION IV

KEYPUNCH INSTRUCTIONS

INTRODUCTION

Text is punched in columns 1-72 of each input card (after the four control cards), using an EBCDIC keypunch (e.g., IBM 029) or the equivalent multiple punches on another model. Letters, numbers, and punctuation are reproduced as they appear in normal typewritten English text, with the exception of quotation marks, accent marks, mathematical symbols, and brackets ([,]). However, additional symbols must be added to the text to indicate capitalization, italics, and format controls.

CARD BOUNDARIES

Column 72 of a card is considered to be adjacent to column 1 of the next card. However, for the convenience of the keypunch operator, extra spaces may be left after the last full word on a card, so that the next word can start on the next card. The extra spaces are collapsed automatically into a single space (or to two spaces after a period or colon).

CAPITALIZATION

If the first letter of the inkprint word is capitalized, the keypunched word must be preceded immediately by a logical-not sign (7). If the whole inkprint word is capitalized, the keypunched word must be preceded immediately by two logical-not signs (7 7). When Roman numerals are written as capital letters, a single logical-not sign must be used before a single letter and two logical-not signs must be used before numerals containing two or more letters.

ITALICS

If only one, two, or three successive inkprint words are italicized, each of the words must be preceded immediately by an underline () when keypunched.

If four or more successive inkprint words are italicized, the first word must be preceded immediately by two underlines () and the last by one underline.

CHARACTERS REQUIRING SPECIAL SYMBOLS

Mathematical Symbols

Symbols such as + (plus), - (minus), = (equal), * >(greater than), and <(less than) must be spelled out as words, even though they appear on the keypunch.

Quotation Marks

The single quote character on the keypunch (') is always taken to mean an apostrophe; thus, special symbols must be used for single quotes.

\$' is punched for a left single quote

\$'R is punched for a right single quote

Ordinarily, the double quote (") on the keypunch may be used for both left and right double quotes. However, double quotes within the scope of another pair of double quotes must be represented by special symbols.

\$" is punched for a left double quote in quoted text

\$"R is punched for a right double quote in quoted text

Accent Marks

Any accent mark used with a letter (such as é, è, ê, ä, ç) is represented by preceding the keypunched letter with a cents sign (¢). For example, Abbé is keypunched →ABB¢E.

Brackets

< is punched for a left bracket ([).

> is punched for a right bracket (]).

*A keypunched equals sign is interpreted as a letter sign.

FORMAT CONTROLS

Paragraphs

The symbol \$P must be keypunched to indicate the beginning of a new paragraph. The text immediately following the P is started on the next line after two spaces.

New Line

The symbol \$L may be used to begin a new output line. Caution: if there are spaces immediately after the L, one space will be put at the beginning of the new line. Any number of lines may be skipped by using the \$L symbol repeatedly.

New Page

The symbol \$PG may be used to begin a new page. It should also be used at the end of the text input.

Tabulation

If the symbol \$TABnb or \$TABnmb is punched, where n's represent digits and b represents a blank, the text beginning immediately after the blank will be started at column n or nn respectively. Tabulation is implemented by the automatic insertion of spaces into the output.

SECTION V

NOTES TO THE BRAILLE EDITOR

This section indicates where the intervention of the editor is required, and presents the tools available to the editor to implement his intervention.

Generally speaking, the nineteen problem situations listed in the Memorandum on Braille Translation ([2]) by R. L. Haynes summarize where human intervention is needed. Some of the problems are new words, foreign words, acronyms, single letters, and measures.* In addition to the problems in Haynes' list, the editor's attention is required for correct use of italics ([3], Rule II.10), and quotation marks ([3], Rule I.2), and placement of footnotes ([3], Rule IV.22).

There are basically two ways in which the editor can ensure proper translation in problem situations: (1) instructing the keypunch operator to add certain symbols to the input text; and (2) modifying the tables which control DOTSYS II. Modification of tables is explained in Part 2; the only times it would be done under normal circumstances would be when a problem word occurs often in a particular body of text; in that case, its correct translation would be added to the contraction table.

In the remainder of this section we shall discuss the special symbols which can be inserted in the input text and the situations in which they are helpful or necessary. These symbols are: the letter sign (=), the grade switch (\$G), the division symbol (\$/), and the termination symbol (\$T).

Letter Sign (=)

All uses of the letter sign must be inserted in the input text, except when an uncapitalized letter from A through J follows a number or is separated by a hyphen following a number.

Grade Switch (\$G)

An occurrence of the grade switch, \$G, changes the mode of translation from grade 2 to grade 1 or vice versa. It must precede and follow any sequence of characters in which no contractions are to be used, such as a foreign word.

* Also coinage and weights. Transposition is currently not automatic.

Division Symbol (\$/)

The division symbol is the most useful and versatile tool of the braille editor, although its operation is simple: it merely prevents contraction of a letter group which crosses it. For example, for the lisped word "thentury," the "the" contraction is avoided by inserting the division symbol after the "th," thus: TH\$/ENTURY. It may be placed between the parts of compound words, such as SPARE\$/RIB. It may be placed between prefixes and stems, as in BI\$/NOMIAL, or indicate syllable division, as in PERITO\$/NEUM and SKE\$/DADDLE.

It must be used in a time interval after the hyphen separating minutes from hours, as in 9:30 - \$/10:30, in order to produce a number sign correctly.

Termination Sign (\$T)

This translates into the double sign dot-6, dot-3, in the output where required. ([3], Rule II.11).

SECTION VI

INPUT ERRORS

Two kinds of errors in the input text are recognized by DOTSYS II and handled predictably:

1. The occurrence of a character in the input text with no braille translation.

If proof output has been specified (on the first control card), the untranslatable character and the next nine characters are replaced by the message " NEW CHAR ". Otherwise, the untranslatable character is replaced by a space.

The untranslatable character may be given a braille translation by adding it to the DOTSYS II alphabet table, as described in Part 2.

2. Improper use of the \$TAB control symbol.

(The symbol \$TAB must be followed immediately by one or two digits and a blank).

If proof output has been specified (on the first control card), the next ten characters (beginning with \$) are replaced by the message " BAD TAB ". In any case the output continues at the beginning of the next line.

If no error message appears in the proof output, but the braille translation is incorrect, take your problem to someone who has read [1] and Part 2.

APPENDIX I

IBM OPERATING SYSTEM/360 PROGRAM DECK COMPOSITION

The control cards and input necessary for a DOTSYS II run, given an object deck, are shown below, under the assumption that a catalogued procedure COBFLG is available to link edit and execute a COBOL program. If no such catalogued procedure is available, refer to the IBM System/360 Operating System COBOL Programmer's Guide (C28-6380) for the control cards it would contain.

<pre>// EXEC COBFLG //LKED.SYSIN DD *</pre>	
<p style="text-align: center;">DOTSYS II object deck</p>	
<pre>/* //GO.SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=120,BLKSIZE=600) //GO.SYSPUNCH DD SYSOUT=B,DCB=(RECFM=FBA,LRECL=81,BLKSIZE=405) //GO.SYSIN DD *</pre>	
<p style="text-align: center;">DOTSYS II tabular input (See Part 2, Appendix II)</p>	
<p>DOTSYS II control cards</p>	<p>proof or braille punched output signs/line lines/page</p>
<p style="text-align: center;">input text</p>	
<pre>/*</pre>	

SAMPLE INPUT DECK AND PROOF OUTPUT FROM IT

PROOF

NO PUNCHED OUTPUT

SIGNS/LINE 24

LINES/PAGE 09

\$PG\$TAB6 →SAMPLE →RUN\$P→ALL OF THE SPECIAL SYMBOLS,

AS DESCRIBED IN __→→DOTSYS →→II →USER'S _→GUIDE <→KEYPUNCH →INSTRUCTIONS

> ARE USED. →HE SAID, "I SAID, \$'THE SIGN FOR =K IS GIVEN THE PROOF

SYMBOL =K EVEN WHEN IT REPRESENTS THE WORD "\$\$GKNOWLEDGE\$G.\$"R\$"R"

→THE EDITOR MAY WISH TO USE THE TERMI\$/NATOR\$T.

TRANSITION TABLE

STATE VARIABLE		01	02	03	04	05
INPUT CLASS	01	R	S	-	-	-
INPUT CLASS	02	S	R	-	-	-
INPUT CLASS	03	-	R	-	-	-
INPUT CLASS	04	R	S	-	-	-
INPUT CLASS	05	-	-	T	-	-
INPUT CLASS	06	R	S	-	-	-
INPUT CLASS	07	R	R	-	-	-
INPUT CLASS	08	R	R	-	-	-
INPUT CLASS	09	R	R	-	-	-
INPUT CLASS	10	R	R	-	S	-
INPUT CLASS	11	R	R	-	R	-
INPUT CLASS	12	-	R	-	-	S
INPUT CLASS	13	-	R	-	-	R
INPUT CLASS	14	R	R	-	-	-
INPUT CLASS	15	R	S	-	-	-

RIGHT CONTEXT TABLE

NON-TERMINAL	INPUT CLASSES		
P	14	03	02 02
L	01	13	13 13

DECISION TABLE

COLUMN		01	02	03	04	05	06	07	08
INPUT CLASS	01	Y	-	-	-	-	-	-	-
INPUT CLASS	02	-	-	Y	-	-	-	-	-
INPUT CLASS	03	G	-	-	-	-	-	-	-
INPUT CLASS	04	-	Y	-	-	-	-	-	-
INPUT CLASS	05	G	-	-	-	-	-	-	-
INPUT CLASS	06	-	-	-	Y	-	-	-	-
INPUT CLASS	07	-	-	-	Y	-	-	-	-
INPUT CLASS	08	-	-	-	-	Y	-	-	-
INPUT CLASS	09	-	-	-	-	-	Y	-	-
INPUT CLASS	10	-	-	-	-	-	-	Y	-
INPUT CLASS	11	-	-	-	-	-	-	-	Y
INPUT CLASS	12	G	-	-	-	-	-	-	-
INPUT CLASS	13	G	-	-	-	-	-	-	-
INPUT CLASS	14	G	-	-	-	-	-	-	-
INPUT CLASS	15	-	-	N	G	-	-	-	-
STATE-VARIABLE	01	-	-	N	-	-	-	-	-
STATE-VARIABLE	02	-	Y	-	N	-	-	-	-
STATE-VARIABLE	03	N	N	-	N	-	-	-	-
STATE-VARIABLE	04	-	-	-	-	Y	N	N	Y
STATE-VARIABLE	05	-	-	-	-	N	Y	-	-

.
..
? K N O W L E D G E . " " " " THE ED I T O R
38 05 29 21 58 07 17 25 27 17 50 52 52 04 52 00 32 46 00 43 10 30 21 23

..
M A Y W I S H T O U S E THE
13 01 61 00 58 10 41 00 22 37 14 17 00 46 00 00 00 00 00 00 00 00 00 00

.
T ER M I N A T O R .
30 59 13 10 29 01 30 21 23 32 04 50 00 00 00 00 00 00 00 00 00 00 00 00

APPENDIX III

SUMMARY OF SPECIAL SYMBOLS

<u>Symbol</u>	<u>Input Representation</u>	<u>Text Reference</u>
Capitalize Letter	^	Section IV, Para. 3
Capitalize Word	^^	Section IV, Para. 3
Italicize Word	_	Section IV, Para. 4
Italics Start	_ _	Section IV, Para. 4
Left Single Quote	\$'	Section IV, Para. 5 (2)
Right Single Quote	\$'R	Section IV, Para. 5 (2)
Left Double Quote	\$"	Section IV, Para. 5 (2)
Right Double Quote	\$"R	Section IV, Para. 5 (2)
Accent Marks	ç	Section IV, Para. 5 (3)
Left Bracket ()	<	Section IV, Para. 5 (4)
Right Bracket ()	>	Section IV, Para. 5 (4)
Start Paragraph	\$P	Section IV, Para. 6 (1)
Begin New Line	\$L	Section IV, Para. 6 (2)
Begin New Page	\$PG	Section IV, Para. 6 (3)
Skip to Column nn	\$TABnn	Section IV, Para. 6 (4)
Letter Sign	=	Section V, Para. 1
Change Grade	\$G	Section V, Para. 2
Divide Word	\$/	Section V, Para. 3
Termination Sign	\$T	Section V, Para. 4

PART 2
TRANSFER AND MAINTENANCE MANUAL

SECTION I

INTRODUCTION

This manual presents the details of the COBOL implementation of DOTSYS II. It is for use in making changes in the tabular input, and in the program itself, that may be made necessary by the transfer of DOTSYS II to another installation, a change in peripheral units, a change in the braille translation rules, or merely a desire to improve the conformity of the translation with the braille ideal.

The prospective user of this document should be aware of the contents of the other two DOTSYS II manuals: DOTSYS II Finite-State Syntax-Directed Braille Translation [1] and DOTSYS II User's Guide (Part 1 of this MTR). To transfer the program, or make changes in the program involving input or output devices, the reader must also be familiar with the implementation of COBOL at his installation.

Appended to this document is essential reference information such as the DOTSYS II source listing, a glossary of data names, a complete set of tables, and the meanings of the state variables and input classes. Also included is a discussion of a program change to eliminate the alphabet table search.

SECTION II

TRANSFER

This section contains a checklist of steps required to bring DOTSYS II into operation on any computer having a Standard COBOL compiler and sufficient core storage to execute DOTSYS II. 'Standard COBOL' means COBOL according to [1], having the level 1 nucleus, table handling, and sequential access. The core storage required depends on the compiler and on the size of the contraction table. With the IBM 360 level F compiler and 256 contraction table entries, about 19,500 characters are required. That table size is sufficient to handle all the braille rules except those involving pronunciation, syllabification, or affix-stem division.

It is assumed here that the program and tables will be key-punched from the listings appended to this manual. If a card deck or the equivalent has been provided, information about how it was produced, and how it is to be used, should accompany it.

The program can be made about fifteen per cent faster, without affecting its translation capability, by means of replacing the alphabet search by a machine-dependent indexing scheme, at the expense of reducing further transferability and increasing the program size. The method of doing this is explained in Appendix I.

Here is the checklist of steps to transfer DOTSYS II:

1. Rewrite the environment division of the program in accordance with the COBOL manual for your computer and peripheral units. Keep in mind that SYSINPUT is the input file, and SYSPRINT and SYSPUNCH are output files. Other information about them is in the file description (FD) entries in the data division.
2. Check the file description entries (in the file section of the data division) to ensure that the block size and label records are specified correctly for your installation.

Punched output records may or may not begin with a control character (for pocket selecting), and the control character, if present, varies with the installation. DOTSYS II is set up to place a control character in the beginning of each punched output record;

the control character used is the value of the 77-level data item POCKET-SELECT in the working storage section. If no control character is to be used, remove the 02-level data item FILLER in CODED-OUTPUT, and change the sentence

WRITE CODED-OUTPUT AFTER POCKET-SELECT

in the BREAK6 paragraph of the output section to

WRITE CODED-OUTPUT.

3. Check the COBOL character set at your installation against the characters in the program. Some characters, such as the single quote ('), greater than (>), less than (<), and equals (=), may have to be replaced by other characters or expressions.

4. Check the character set of your computer and keypunch against the characters in the tables. Remove or replace table entries containing illegal characters. Inkprint characters having no single-character machine-readable form may be represented by multiple-character symbols, just as in the case of single quotation marks. They are implemented via contraction table entries. Be sure to inform the keypunch operator of your choices in this matter.

5. Compile the program to obtain an object deck. If the object program is too large, try leaving out the table display routine, from TABLE-DISPLAY through SKIP-DISPLAY. As a last resort, decrease the contraction table bound (and shorten the table input accordingly).

6. Execute the program. Input consists of the tabular input, as described in Appendix II, followed (with no separation) by the user input, as described in the User's Guide (Part 1).

SECTION III

OUTPUT DEVICES

The three kinds of output provided by DOTSYS II are discussed in Part 1 and [1]. The 'braille' output is suitable for embossing on a line printer which has been modified by placing an elastic band between the print hammers and the paper. The elastic band might be a rubber band, a garter belt, or whatever the user's ingenuity can provide. It can be held on by tape or wire at the ends, or, on an IBM 1403 line printer, by hooks obtainable free of charge from IBM if requested under the number RPQ818047. The number of lines per inch should be set as close as possible to ten.

Alterations in the output may be needed with other line printer modifications or different methods of embossing braille. It may be satisfactory for the user to write his own program to convert the coded punched output of DOTSYS II into the form required by his embossing device; if not, the output section of DOTSYS II must be altered. To make this task as easy as possible, a flow chart of the output section (Appendix III) and a glossary of all data names in the program (Appendix IV) have been appended, as well as the program listing (Appendix VII).

SECTION IV

CONTRACTION TABLE ADDITIONS

The purpose and composition of contraction table entries are given in [1], Sections III.C.1 and III.C.2. After deciding upon the values of each of the fields of a new entry, punch a card for the entry in the format given in Appendix V, and place it in the table in accordance with the rules in [1], Section III.C.3.

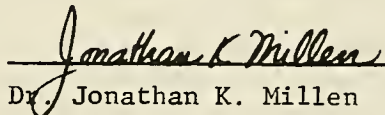
Deciding upon the input class of the contraction table entry requires information given in Appendix VI about the presently implemented input classes. Roughly speaking, parts of words should be put in input class 1, 4, or 6, while whole words should be put in input class 6 and given a right context of P. If specific punctuation following a word is included in the STRING field, the entry should be put in input class 7 instead. Creation of a new input class will rarely be necessary; study all of Section III in Part 1 before attempting it.

The capacity of the contraction table is the number in the OCCURS clause for the data item TABLE-ENTRY in CONTRACTION-TABLE. Change it if necessary to keep it greater than the number of contraction table entries.

SECTION V

TABLE SIZE BOUNDS

The following data items are conceptually variable in size: STATE-VECTOR, CONTRACTION-TABLE, ALPHABET, RIGHT-CONTEXT-TABLE, DECISION-TABLE, and TRANSITION-TABLE. Each has an OCCURS clause in the O2-level item subordinate to it which gives the maximum size provided for it during compilation. If additions are made to the tables, as might result from an extension of the finite-state logic, increase the OCCURS numbers where necessary to accommodate the larger tables.


Dr. Jonathan K. Millen

JKM:ces

APPENDIX I

REPLACING THE ALPHABET TABLE SEARCH BY INDEXING

The first step in translating the current contents of the buffer is to find the alphabet table entry for the leftmost character in the buffer (cf. III,A,D in [1]). In order to preserve machine-independence, DOTSYS II does this by searching the alphabet table linearly from the top for a match with the leftmost character in the buffer. If the alphabet table has been ordered with the higher-frequency items nearer the top, an average of about nine entries are tested. On the IBM 360/50, that takes about one millisecond, or roughly fifteen per cent of the average time spent per character.

On some machines, including the IBM 360, it is possible to nearly eliminate the time required to find the appropriate alphabet table entry, by using the test character itself as an index. This is done by moving the character to a data item defined as USAGE DISPLAY (the default case) but redefined as USAGE COMPUTATIONAL. For example, on the IBM 360, the data item below describes a half-word integer whose value is determined by the character moved into its right-hand byte; its left-hand byte is binary zeroes.

```
01 CHARACTER-INDEX.  
   02 LEFT          PICTURE X, VALUE LOW-VALUE.  
   02 RIGHT        PICTURE X.  
01 N REDEFINES CHARACTER-INDEX,  
   PICTURE S999, USAGE COMPUTATIONAL.
```

Thus, if a character is moved to RIGHT, then N is a number determined by that character, and can be used as an index into an array of pointers to the appropriate places in the alphabet table. For example, suppose the array of pointers is called ARRAY-OF-POINTERS and the leftmost character in the buffer is a space, whose alphabet table entry comes first. Moving the space to RIGHT gives N a value of hexadecimal 40 or decimal 64; the 64th entry in ARRAY-OF-POINTERS should be 1 since the space entry is first in the alphabet table. Thus, the two sentences,

```
MOVE RL1 TO RIGHT.  
MOVE ARRAY-OF-POINTERS (N) TO LETTER.
```

replace the alphabet table search, provided that ARRAY-OF-POINTERS has been properly initialized, which takes only two sentences in the

READ-ALPHABET loop during initialization. The same thing can be done to replace the search occurring in the paragraphs following TEST-NON-TERMINAL (which check right context) in the translation section.

Note that with indexing, there is no longer any need to order the alphabet table in any particular way; however, the order of contraction table sections must still match the order of the associated alphabet table entries.

The price paid for the reduced search time is an increased storage requirement, since ARRAY-OF-POINTERS must have 2^k entries, where k is the number of bits in a character. On the IBM 360, ARRAY-OF-POINTERS would take up $2^8 \times 2 = 512$ bytes.

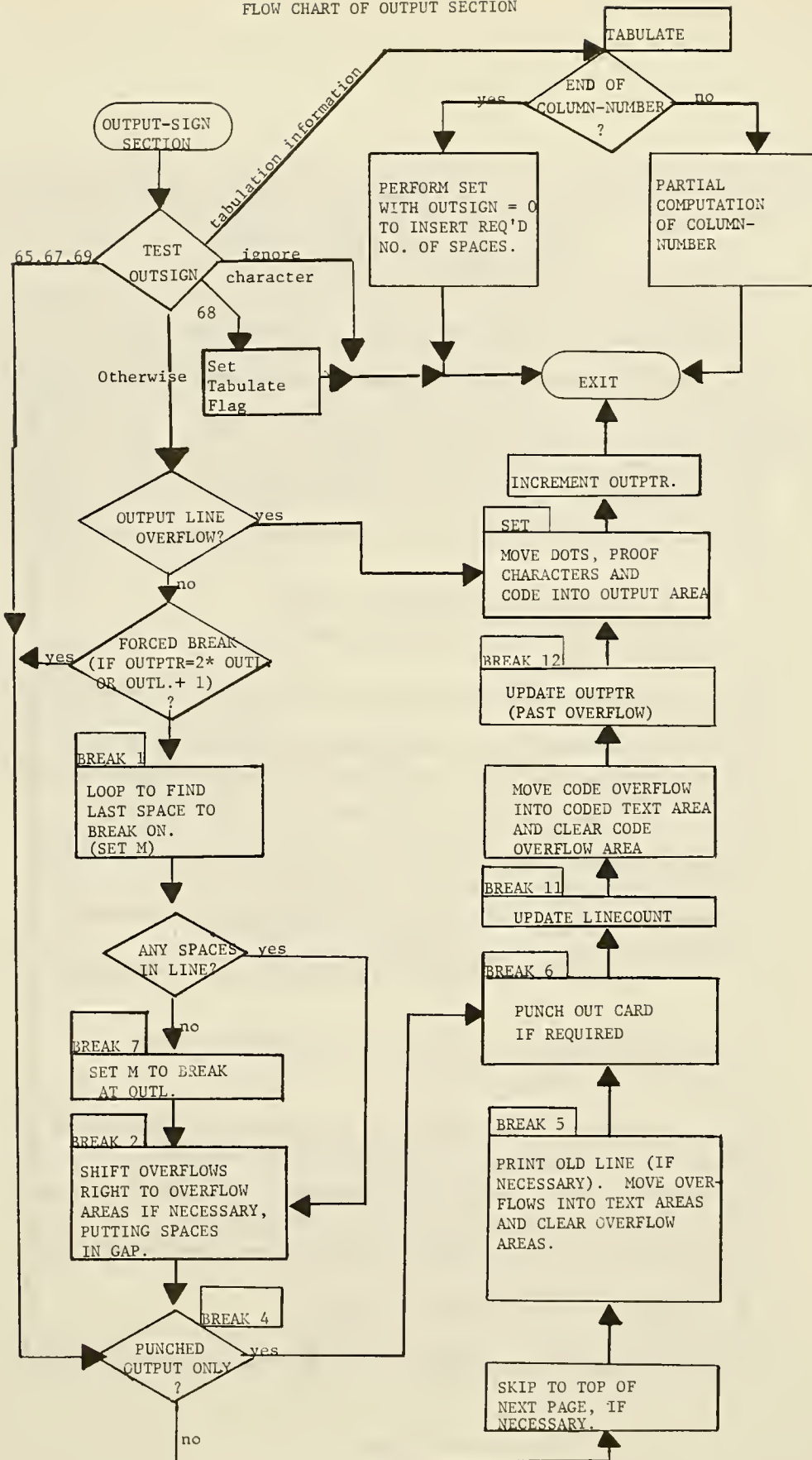
APPENDIX II

DECK SETUP OF TABULAR INPUT

For the formats of these cards, see Appendix V.

Up to 64 alphabet table cards; last card must have CHAR-CLASS of 99.
Up to 450 contraction table cards; last card must have INPUT-CLASS of 99.
NNF card (number of right-context table entries)
right-context table cards (one for each entry)
NSV card (number of state variables)
NIC card (number of input classes)
transition table cards (one for each column) (there are (NSV) columns)
NDTC card (number of decision table columns)
decision table cards (one for each column)
Exactly 64 alphabet table cards.

FLOW CHART OF OUTPUT SECTION



APPENDIX IV

GLOSSARY OF DATA NAMES

All data names used in the program are listed alphabetically below. Each variable (77-level data item) is accompanied by a short explanation.

For main data items the reader is referred to the text in this manual or [1]. The name of a subordinate data item is accompanied by the name of its main data item. COBOL received words are described as such.

ALPHABET (See [1])

ALPHABETIC-ENTRY IN ALPHABET

BRAILLE-LINE IN OUTPUT-WORK-AREA

-LINE-CHARTS IN OUTPUT-WORK-AREA

-LINE-INDEXED IN OUTPUT-WORK-AREA

-LINES IN OUTPUT-WORK-AREA

-LINES-INDEXED IN OUTPUT-WORK-AREA

-OVERFLOW IN OUTPUT-WORK-AREA

-SIGN IN OUTPUT-WORK-AREA

-TEXT IN OUTPUT-WORK-AREA

BUFFER (See [1])

CHAR IN INPUT-RECORD

-CLASS IN ALPHABET

CODE-LINE IN OUTPUT-WORK-AREA

-LINE-INDEXED IN OUTPUT-WORK-AREA

CODED-OUT IN CODED-OUTPUT

-OUTPUT is the punched data record.

-OVERFLOW IN OUTPUT-WORK-AREA

-SIGN IN OUTPUT-WORK-AREA

-TEXT IN OUTPUT-WORK-AREA

COLUMN-PAIR IN OUTPUT-WORK-AREA

CONDITION IN DECISION-TABLE

D_n IN LETTER-TO-DIGIT-CODE

DECISION IN DECISION-TABLE

-TABLE (See [1])

-TABLE-COLUMN IN DECISION-TABLE

DIGIT IN LETTER-TO-DIGIT-CODE

-SIGNS LETTER-TO-DIGIT-CODE

-SIGNS-INDEXED LETTER-TO-DIGIT-CODE

DOTS-n-n IN SIGN TABLE

EXTENT IN ALPHABET
 FIELD_n, FIELD_{nm} IN INPUT-RECORD
 FILLER is a COBOL reserved word for a data item whose contents are not used explicitly.
 FIRSTCHAR IN BUFFER
 FOUND-SECOND-SYMBOL is a flag that records if the second-character in the buffer has been successfully matched during the contraction table search. Its possible values are 'Y' for 'Yes' and 'N' for 'No'.
 I is a general-purpose integer variable.
 INDEX points to the contraction table entry currently being examined.
 INITIAL-STATE holds the initial value of the state-vector.
 INPTR points to the next character in the input record.
 INPUT-CLASS IN CONTRACTION-TABLE
 INPUT-RECORD is the input data record.
 J is a general-purpose integer variable.
 K is a general-purpose integer variable.
 L IN BUFFER
 LETTER points to the alphabet table entry for the first character in the buffer.
 -T0-DIGIT-CODE is used to convert braille signs for the letters A-J back into digits in the OUTPUT-SIGN section.
 OUTPTR points to the next position available for a braille sign in the output line or its overflow. Its value is 80 or less. It is used in the OUTPUT-SIGN section.
 OUTPUT-COLUMN IN OUTPUT-WORK-AREA
 -COLUMN-PAIRS IN OUTPUT-WORK-AREA
 -LINE is the printed data record.
 -TEXT IN OUTPUT-WORK-AREA
 -WORK-AREA holds all output.
 OUTSIGN is given the numerical code of the next braille sign to be placed in the output line, prior to PERFORM-ing the OUTPUT-SIGN section.
 POCKET-SELECT holds the punch control character to select stacker 1.
 POINTER is the index (1 to 9) of the next characters to be compared in the buffer and the STRING field of a contraction table entry.
 PROOF IN OUTPUT-WORK-AREA
 -CHARACTERS IN SIGN-TABLE
 -LINE IN OUTPUT-WORK-AREA
 -LINE INDEXED IN OUTPUT-WORK-AREA
 -LINE-OVERFLOW IN OUTPUT-WORK-AREA
 -OR-BRAILLE records whether proof output (value 'P') or printed braille output (value 'B') is desired. It is read from the first control card during initialization, and used only in the OUTPUT-SIGN section.

PROOF-TEXT IN OUTPUT-WORK-AREA
 PUNCH is the punched file.
 PUNCHED-OUTPUT records whether punched output will be produced:
 'Y' for 'Yes', 'N' for 'No', 'O' for 'Only punched output'. It
 is read from the second control card during initialization.

R IN BUFFER
 RCHAR IN BUFFER
 RCHARS IN BUFFER
 RIGHT-CONTEXT IN CONTRACTION TABLE
 -CONTEXT-CLASS IN RIGHT-CONTEXT-TABLE
 -CONTEXT-CLASSES IN RIGHT-CONTEXT-TABLE
 -CONTEXT-TABLE (See [1])
 -CONTEXT-TABLE-ENTRY IN RIGHT-CONTEXT-TABLE

RL1, RL9 IN BUFFER
 RR1, RR9 IN BUFFER
 SHIFT IN CONTRACTION-TABLE
 SIGN IN CONTRACTION-TABLE
 -TABLE (See [1])
 -TABLE ENTRY IN SIGN-TABLE

SIGNS IN CONTRACTION-TABLE
 SPACE is a COBOL reserved word for a data item filled with blank
 characters.

SPACES is a COBOL reserved word for a data item filled with blank
 characters.

SINGLE-SIGN IN ALPHABET
 STATE-VARIABLE IN STATE-VECTOR
 -VECTOR (See [1])

STRING IN CONTRACTION-TABLE
 SYMBOL IN ALPHABET
 SYSINPUT is the input data record.
 SYSPRINT is the printed file.

TABLE-CHAR IN CONTRACTION-TABLE
 -ENTRY IN CONTRACTION-TABLE
 -INFO IN INPUT-RECORD

TABULATE is a flag used in the OUTPUT-SIGN section to handle tabulation.

TEMP holds the right-hand nine characters of the buffer temporarily,
 during shifting.

TEMPL is temporary storage for one character. Its initial value must
 be non-blank for its use during initialization.

TEXT IN INPUT-RECORD
 THREE-SPACES is three spaces.

TRANSITION IN TRANSITION-TABLE
 -TABLE (See [1])
 -TABLE-COLUMN IN TRANSITION-TABLE

ZEROS is a COBOL reserved word for a data item filled with zeros,
 (either characters or integers).

APPENDIX V

FORMAT OF TABULAR INPUT CARDS

The table below shows which punched card columns are used for each data item:

Punched Card Columns	1-9	16	18,19	21,22	24-31
Alphabet Table**	SYMBOL*		CHAR-CLASS		SINGLE-SIGN*
Contraction Table	STRING	RIGHT-CONTEXT	INPUT CLASS	SHIFT	SIGNS
(NNT)			NNT		
Right-Context Table		NON-TERMINAL			RIGHT-CONTEXT-CLASSES
(NSV)			NSV		
(NIC)			NIC		
Transition Table***					
(NDTC)			NDTC		
Decision Table****					
Sign Table	(all)				

* Left justified

** The extent field is filled automatically by the program during initialization.

*** The kth card for the table (k=1,...,(NSV)) contains the characters in the kth column of the table, top to bottom, in columns 1 to (NIC).

**** The kth card for the table (k=1,...,NDTC) contains the characters in the kth column of the table, top to bottom, in columns 1 to (NSV + NIC).

Note: All numbers above are expressed as two-digit numbers, and therefore must be given a leading 0 if less than 10.

APPENDIX VI

CURRENT STATE VARIABLES AND INPUT CLASSES

State Variable	1 after the start of a number
	2 after the start of a word
	3 grade 1 translation
	4 in a quotation
	5 in italicized text
Input Class	1 contraction always used in grade 2
	2 digits
	3 most punctuation
	4 contractions used after the start of a word
	5 \$G (grade switch)
	6 contractions used only at the start of a word
	7 isolated full-word contractions
	8 \$P" (start paragraph in quotation)
	9 \$P (start paragraph in italics)
	10 " (left quote)
	11 " (right quote)
	12 _ _ (begin italics)
	13 _ (last word of italics)
	14 (space)
	15 A to J occurring in a number

APPENDIX VII

1

DOTSYS II SOURCE LISTING

```

00001 PROGRAM-ID. 'GRADE2'.
00002 AUTHOR. J. K. MILLEN.
00003 INSTALLATION. MIT
00004
00005 ENVIRONMENT DIVISION.
00006
00007 CONFIGURATION SECTION.
00008 SOURCE-COMPUTER. IBM-360 M65.
00009 OBJECT-COMPUTER. IBM-360 M50.
00010
00011 INPUT-OUTPUT SECTION.
00012 FILE-CONTROL.
00013     SELECT SYSINPUT, ASSIGN TO 'SYSIN' UTILITY.
00014     SELECT SYSPRINT, ASSIGN TO 'SYSPRINT' UTILITY.
00015     SELECT PUNCH, ASSIGN TO 'SYSPUNCH' UTILITY.
00016
00017
00018
00019
00020 DATA DIVISION.
00021
00022
00023 FILE SECTION.
00024
00025 FD SYSINPUT
00026     LABEL RECORDS ARE STANDARD,
00027     RECORDING MODE IS F,
00028     RECORD CONTAINS 80 CHARACTERS,
00029     BLOCK CONTAINS 0 RECORDS,
00030     DATA RECORD IS INPUT-RECORD.
00031
00032 01 INPUT-RECORD.
00033     02 TEXT.
00034         03 CHAR OCCURS 80 TIMES, PICTURE X.
00035     02 TABLE-INFO REDEFINES TEXT.
00036         03 FIELD1.
00037             04 FIELD11 PICTURE X(1).
00038             04 FIELD12 PICTURE X(9).
00039         03 FILLER PICTURE X(5).
00040         03 FIELD2 PICTURE X.
00041         03 FILLER PICTURE A.
00042         03 FIELD3 PICTURE 99.
00043         03 FILLER PICTURE A.
00044         03 FIELD4 PICTURE 99.

```



```

00045          03 FILLER          PICTURE A.
00046          03 FIELD5.
00047          04 FIELD51        PICTURE 99.
00048          04 FIELD52        PICTURE 99.
00049          04 FIELD53        PICTURE 99.
00050          04 FIELD54        PICTURE 99.
00051          03 FILLER          PICTURE X(49).
00052
00053 FD  SYSPRINT
00054          LABEL RECORDS ARE STANDARD,
00055          RECORDING MODE IS F,
00056          DATA RECORD IS OUTPUT-LINE.
00057
00058 01  OUTPUT-LINE.
00059          02 FILLER          PICTURE A.
00060          02 OUT.
00061          03 OUT-PLACE      OCCURS 120 TIMES, PICTURE X.
00062
00063 FD  PUNCH
00064          LABEL RECORDS ARE STANDARD,
00065          RECORDING MODE IS F,
00066          DATA RECORD IS CODED-OUTPUT.
00067
00068 01  CODED-OUTPUT.
00069          02 FILLER          PICTURE X.
00070          02 CODED-OUT      PICTURE X(80).
00071
00072
00073 WORKING-STORAGE SECTION.
00074
00075          77 FOUND-SECOND-SYMBOL PICTURE X.
00076          77 I              PICTURE S999, USAGE COMPUTATIONAL.
00077          77 INDEX          PICTURE S999, USAGE COMPUTATIONAL.
00078          77 INITIAL-STATE  PICTURE X(10), VALUE 'NNNNNNNNNN'.
00079          77 INPTR PICTURE S99, USAGE COMPUTATIONAL.
00080          77 J              PICTURE S999, USAGE COMPUTATIONAL.
00081          77 K              PICTURE S999, USAGE COMPUTATIONAL.
00082          77 LETTER        PICTURE S99, USAGE COMPUTATIONAL.
00083          77 LINECOUNT    PICTURE S99, USAGE COMPUTATIONAL,
00084                          VALUE 0.
00085          77 LPG            PICTURE S99, USAGE COMPUTATIONAL.
00086          77 M              PICTURE S99, USAGE COMPUTATIONAL.
00087          77 N              PICTURE S999, USAGE COMPUTATIONAL.
00088          77 NALPHABET    PICTURE S999, USAGE COMPUTATIONAL,
00089                          VALUE 0.
00090          77 NDTC          PICTURE S99, USAGE COMPUTATIONAL.
00091          77 NIC           PICTURE S99, USAGE COMPUTATIONAL.

```

00092 77 NN PICTURE S999, USAGE COMPUTATIONAL.
 00093 77 NNT PICTURE S99, USAGE COMPUTATIONAL.
 00094 77 NSV PICTURE S99, USAGE COMPUTATIONAL.
 00095 77 NXTCHR PICTURE X.
 00096 77 OUTPTR PICTURE S99, VALUE 1, USAGE COMPUTATIONAL.
 00097 77 OUTL PICTURE S99, USAGE COMPUTATIONAL,
 00098 VALUE 35.
 00099 77 OUTSIGN PICTURE S99, USAGE COMPUTATIONAL.
 00100 77 POCKET-SELECT PICTURE X, VALUE 'V'.
 00101 77 POINTER PICTURE S99, USAGE COMPUTATIONAL.
 00102 77 PROOF-OR-BRAILLE PICTURE X.
 00103 77 PUNCHED-OUTPUT PICTURE X.
 00104 77 TABULATE PICTURE S99, USAGE COMPUTATIONAL,
 00105 VALUE 0.
 00106 77 TEMP PICTURE X(9).
 00107 77 TEMPL PICTURE X, VALUE 'X'.
 00108 77 THREE-SPACES PICTURE AAA, VALUE SPACES.
 00109
 00110 01 BUFFER.
 00111 02 L.
 00112 03 RL9 PICTURE X(9).
 00113 03 RR1 PICTURE X.
 00114 02 R REDEFINES L.
 00115 03 RL1 PICTURE X.
 00116 03 RR9 PICTURE X(9).
 00117 02 RCHARS REDEFINES L.
 00118 03 FIRSTCHAR PICTURE X.
 00119 03 RCHAR OCCURS 9 TIMES, PICTURE X.
 00120
 00121 01 STATE-VECTOR.
 00122 02 STATE-VARIABLE OCCURS 10 TIMES, DEPENDING ON NSV,
 00123 PICTURE X.
 00124
 00125 01 CONTRACTION-TABLE.
 00126 02 TABLE-ENTRY OCCURS 450 TIMES.
 00127 03 STRING.
 00128 04 TABLE-CHAR OCCURS 9 TIMES, PICTURE X.
 00129 03 RIGHT-CONTEXT PICTURE X.
 00130 03 INPUT-CLASS PICTURE S99, USAGE COMPUTATIONAL.
 00131 03 SHIFT PICTURE S99, USAGE COMPUTATIONAL.
 00132 03 SIGNS.
 00133 04 SIGN OCCURS 4 TIMES, PICTURE S99,
 00134 USAGE COMPUTATIONAL.
 00135
 00136 01 ALPHABET.
 00137 02 ALPHABETIC-ENTRY OCCURS 64 TIMES.
 00138 03 SYMBOL PICTURE X.

00139 03 CHAR-CLASS PICTURE S99, USAGE COMPUTATIONAL.
 00140 03 SINGLE-SIGN PICTURE S99, USAGE COMPUTATIONAL.
 00141 03 EXTENT PICTURE S999, USAGE COMPUTATIONAL.
 00142
 00143 01 RIGHT-CONTEXT-TABLE.
 02 RIGHT-CONTEXT-TABLE-ENTRY OCCURS 2 TIMES.
 03 NON-TERMINAL PICTURE X.
 03 RIGHT-CONTEXT-CLASSES.
 04 RIGHT-CONTEXT-CLASS OCCURS 4 TIMES,
 PICTURE S99, USAGE COMPUTATIONAL.
 00149
 00150 01 DECISION-TABLE.
 02 DECISION-TABLE-COLUMN OCCURS 10 TIMES,
 DEPENDING ON NDC.
 03 DECISION OCCURS 20 TIMES,
 DEPENDING ON NIC, PICTURE X.
 03 CONDITION OCCURS 10 TIMES,
 DEPENDING ON NSV, PICTURE X.
 00157
 00158 01 TRANSITION-TABLE.
 02 TRANSITION-TABLE-COLUMN OCCURS 10 TIMES,
 DEPENDING ON NSV.
 03 TRANSITION OCCURS 20 TIMES,
 DEPENDING ON NIC, PICTURE X.
 00163
 00164 01 SIGN-TABLE.
 02 SIGN-TABLE-ENTRY OCCURS 64 TIMES.
 03 DOTS-1-4 PICTURE XX.
 03 DOTS-2-5 PICTURE XX.
 03 DOTS-3-6 PICTURE XX.
 03 PROOF-CHARACTERS PICTURE XXX.
 00170
 00171 01 OUTPUT-WORK-AREA.
 02 BRAILLE-LINES.
 03 BRAILLE-LINE OCCURS 3 TIMES.
 04 BRAILLE-TEXT PICTURE X(120).
 04 BRAILLE-OVERFLOW PICTURE X(120).
 02 BRAILLE-LINES-INDEXED REDEFINES BRAILLE-LINES.
 03 BRAILLE-LINE-INDEXED OCCURS 3 TIMES.
 04 BRAILLE-SIGN OCCURS 80 TIMES, PICTURE XXX.
 02 BRAILLE-LINE-CHARS REDEFINES BRAILLE-LINES.
 03 LINE-CHARS OCCURS 3 TIMES.
 04 LINE-CHAR OCCURS 240 TIMES, PICTURE X.
 02 PROOF-LINE.
 03 PROOF-TEXT PICTURE X(120).
 03 OUTPUT-TEXT REDEFINES PROOF-TEXT.
 04 OUTPUT-COLUMN OCCURS 120 TIMES, PICTURE X.

```

00186           03 OUTPUT-COLUMN-PAIRS REDEFINES PROOF-TEXT.
00187           04 COLUMN-PAIR OCCURS 60 TIMES, PICTURE 99.
00188           03 PROOF-LINE-OVERFLOW PICTURE X(120).
00189   02 PROOF-LINE-INDEXED REDEFINES PROOF-LINE.
00190           03 PROOF OCCURS 80 TIMES, PICTURE XXX.
00191   02 CODE-LINE.
00192           03 CODED-TEXT      PICTURE X(80).
00193           03 CODED-OVERFLOW PICTURE X(80).
00194   02 CODE-LINE-INDEXED REDEFINES CODE-LINE.
00195           03 CODED-SIGN      OCCURS 80 TIMES, PICTURE 99.
00196
00197   01 LETTER-TO-DIGIT-CODE.
00198   02 DIGIT-SIGNS.
00199           03 D0 PICTURE S99, USAGE COMPUTATIONAL, VALUE 26.
00200           03 D1 PICTURE S99, USAGE COMPUTATIONAL, VALUE 1.
00201           03 D2 PICTURE S99, USAGE COMPUTATIONAL, VALUE 3.
00202           03 D3 PICTURE S99, USAGE COMPUTATIONAL, VALUE 9.
00203           03 D4 PICTURE S99, USAGE COMPUTATIONAL, VALUE 25.
00204           03 D5 PICTURE S99, USAGE COMPUTATIONAL, VALUE 17.
00205           03 D6 PICTURE S99, USAGE COMPUTATIONAL, VALUE 11.
00206           03 D7 PICTURE S99, USAGE COMPUTATIONAL, VALUE 27.
00207           03 D8 PICTURE S99, USAGE COMPUTATIONAL, VALUE 19.
00208           03 D9 PICTURE S99, USAGE COMPUTATIONAL, VALUE 10.
00209   02 DIGIT-SIGNS-INDEXED REDEFINES DIGIT-SIGNS.
00210           03 DIGIT OCCURS 10 TIMES, PICTURE S99,
00211                USAGE COMPUTATIONAL.
00212
00213
00214
00215   PROCEDURE DIVISION.
00216
00217
00218   INITIALIZATION SECTION.
00219
00220   OPEN-FILES.
00221       OPEN INPUT SYSINPUT.
00222       OPEN OUTPUT SYSPRINT.
00223       MOVE SPACES TO OUT.
00224       WRITE OUTPUT-LINE AFTER ADVANCING 0.
00225       MOVE 999 TO EXTENT (1).
00226       MOVE 1 TO N.
00227   READ-ALPHABET.
00228       MOVE EXTENT (1) TO EXTENT (N).
00229       READ SYSINPUT, AT END GO TO FIN.
00230       ADD 1 TO NALPHABET.
00231       MOVE FIELD11 TO SYMBOL (N).
00232       MOVE FIELD3 TO CHAR-CLASS (N).

```

```

00233         MOVE FIELD51 TO SINGLE-SIGN (N).
00234         ADD 1 TO N.
00235         IF FIELD3 NOT = '99' THEN GO TO READ-ALPHABET.
00236
00237         MOVE 1 TO I.
00238         MOVE 1 TO N.
00239     FILL-TABLE.
00240         READ SYSINPUT, AT END GO TO FIN.
00241         IF FIELD11 = TEMPL THEN GO TO MOVE-FIELDS.
00242         MOVE FIELD1 TO TEMPL.
00243         MOVE N TO EXTENT (I).
00244         ADD 1 TO I.
00245     MOVE-FIELDS.
00246         MOVE FIELD12 TO STRING (N).
00247         MOVE FIELD2 TO RIGHT-CONTEXT (N).
00248         MOVE FIELD3 TO INPUT-CLASS (N).
00249         MOVE FIELD4 TO SHIFT (N).
00250         MOVE FIELD51 TO SIGN (N, 1).
00251         MOVE FIELD52 TO SIGN (N, 2).
00252         MOVE FIELD53 TO SIGN (N, 3).
00253         MOVE FIELD54 TO SIGN (N, 4).
00254         ADD 1 TO N.
00255         IF FIELD3 NOT = '99' THEN GO TO FILL-TABLE.
00256
00257     READ-RIGHT-CONTEXT-TABLE.
00258         READ SYSINPUT, AT END GO TO FIN.
00259         MOVE FIELD3 TO NNT.
00260         MOVE 1 TO N.
00261     READ-NON-TERMINALS.
00262         READ SYSINPUT, AT END GO TO FIN.
00263         MOVE FIELD2 TO NON-TERMINAL (N).
00264         MOVE FIELD51 TO RIGHT-CONTEXT-CLASS (N, 1).
00265         MOVE FIELD52 TO RIGHT-CONTEXT-CLASS (N, 2).
00266         MOVE FIELD53 TO RIGHT-CONTEXT-CLASS (N, 3).
00267         MOVE FIELD54 TO RIGHT-CONTEXT-CLASS (N, 4).
00268         ADD 1 TO N.
00269         IF N NOT > NNT THEN GO TO READ-NON-TERMINALS.
00270
00271     READ-STATE-TABLES.
00272         READ SYSINPUT, AT END GO TO FIN.
00273         MOVE FIELD3 TO NSV.
00274         READ SYSINPUT, AT END GO TO FIN.
00275         MOVE FIELD3 TO NIC.
00276         MOVE 1 TO N.
00277     READ-TRANSITION-TABLE.
00278         READ SYSINPUT, AT END GO TO FIN.
00279         MOVE TABLE-INFO TO TRANSITION-TABLE-COLUMN (N).

```

```

00280         ADD 1 TO N.
00281         IF N NOT > NSV THEN GO TO READ-TRANSITION-TABLE.
00282         READ SYSINPUT, AT END GO TO FIN.
00283         MOVE FIELD3 TO NDTC.
00284         MOVE 1 TO N.
00285 READ-DECISION-TABLE.
00286         READ SYSINPUT, AT END GO TO FIN.
00287         MOVE TABLE-INFO TO DECISION-TABLE-COLUMN (N).
00288         ADD 1 TO N.
00289         IF N NOT > NDTC THEN GO TO READ-DECISION-TABLE.
00290         MOVE INITIAL-STATE TO STATE-VECTOR.
00291
00292 READ-SIGN-TABLE.
00293         MOVE 1 TO N.
00294 READ-SIGN-TABLE-ENTRY.
00295         READ SYSINPUT, AT END GO TO FIN.
00296         MOVE TABLE-INFO TO SIGN-TABLE-ENTRY (N).
00297         ADD 1 TO N.
00298         IF N NOT > 64 THEN GO TO READ-SIGN-TABLE-ENTRY.
00299
00300 READ-CONTROL-CARDS.
00301         READ SYSINPUT, AT END GO TO FIN.
00302         MOVE FIELD11 TO PROOF-OR-BRAILLE.
00303         READ SYSINPUT, AT END GO TO FIN.
00304         MOVE FIELD11 TO PUNCHED-OUTPUT.
00305         READ SYSINPUT, AT END GO TO FIN.
00306         MOVE FIELD3 TO OUTL.
00307         READ SYSINPUT, AT END GO TO FIN.
00308         MOVE FIELD3 TO LPG.
00309         IF PUNCHED-OUTPUT NOT = 'N' THEN OPEN OUTPUT PUNCH.
00310
00311
00312 TABLE-DISPLAY.
00313
00314         IF PROOF-OR-BRAILLE NOT = 'P' THEN GO TO SKIP-DISPLAY.
00315         MOVE '          RIGHT CONTEXT TABLE' TO OUT.
00316         WRITE OUTPUT-LINE AFTER 3.
00317         MOVE 'NON-TERMINAL  INPUT CLASSES' TO OUT.
00318         WRITE OUTPUT-LINE AFTER 2.
00319         MOVE 1 TO K.
00320 DISP14. MOVE SPACES TO OUTPUT-TEXT.
00321         MOVE NON-TERMINAL (K) TO OUTPUT-COLUMN (6).
00322         MOVE RIGHT-CONTEXT-CLASS (K, 1) TO COLUMN-PAIR (8).
00323         MOVE RIGHT-CONTEXT-CLASS (K, 2) TO COLUMN-PAIR (10).
00324         MOVE RIGHT-CONTEXT-CLASS (K, 3) TO COLUMN-PAIR (12).
00325         MOVE RIGHT-CONTEXT-CLASS (K, 4) TO COLUMN-PAIR (14).
00326         MOVE OUTPUT-TEXT TO OUT.

```

```

00327          WRITE OUTPUT-LINE AFTER 2.
00328          ADD 1 TO K.
00329          IF K NOT > NNT THEN GO TO DISP14.
00330 DISP13. MOVE SPACES TO OUT.
00331          WRITE OUTPUT-LINE AFTER 0.
00332          MOVE '          DECISION TABLE' TO OUT.
00333          WRITE OUTPUT-LINE AFTER 3.
00334          MOVE 'COLUMN' TO OUTPUT-TEXT.
00335          PERFORM DISP1 VARYING K FROM 1 BY 1 UNTIL K = NDTC.
00336 DISP1. COMPUTE I = 9 + 2 * K.
00337          MOVE K TO COLUMN-PAIR (I).
00338 END-DISP1.
00339          MOVE OUTPUT-TEXT TO OUT.
00340          WRITE OUTPUT-LINE AFTER 2.
00341 DISP5.
00342          MOVE 1 TO N.
00343 DISP6. MOVE 'INPUT CLASS' TO OUTPUT-TEXT.
00344          MOVE N TO COLUMN-PAIR (9).
00345          MOVE 1 TO K.
00346 DISP8. COMPUTE I = 18 + 4 * K.
00347          MOVE DECISION (K, N) TO OUTPUT-COLUMN (I).
00348          MOVE OUTPUT-TEXT TO OUT.
00349          ADD 1 TO K.
00350          IF K NOT > NDTC THEN GO TO DISP8.
00351 DISP7. WRITE OUTPUT-LINE AFTER 2.
00352          ADD 1 TO N.
00353          IF N NOT > NIC THEN GO TO DISP6.
00354          MOVE SPACES TO OUT.
00355          WRITE OUTPUT-LINE AFTER 1.
00356          MOVE 1 TO N.
00357 DISP2. MOVE 'STATE-VARIABLE' TO OUTPUT-TEXT.
00358          MOVE N TO COLUMN-PAIR (9).
00359          MOVE 1 TO K.
00360 DISP4. COMPUTE I = 18 + 4 * K.
00361          MOVE CONDITION (K, N) TO OUTPUT-COLUMN (I).
00362          MOVE OUTPUT-TEXT TO OUT.
00363          ADD 1 TO K.
00364          IF K NOT > NDTC THEN GO TO DISP4.
00365 DISP3. WRITE OUTPUT-LINE AFTER 2.
00366          ADD 1 TO N.
00367          IF N NOT > NSV THEN GO TO DISP2.
00368 DISP9. MOVE SPACES TO OUT.
00369          WRITE OUTPUT-LINE AFTER 0.
00370          MOVE '          TRANSITION TABLE' TO OUT.
00371          WRITE OUTPUT-LINE AFTER 3.
00372          MOVE 'STATE VARIABLE' TO OUTPUT-TEXT.
00373          PERFORM DISP1 VARYING K FROM 1 BY 1 UNTIL K > NSV.

```

```

00374         MOVE OUTPUT-TEXT TO OUT.
00375         WRITE OUTPUT-LINE AFTER 2.
00376         PERFORM DISP10 THRU DISP11 VARYING N FROM 1 BY 1
00377         UNTIL N = NIC.
00378     DISP10. MOVE 'INPUT CLASS' TO OUTPUT-TEXT.
00379         MOVE N TO COLUMN-PAIR (9).
00380         MOVE 1 TO K.
00381     DISP12. COMPUTE I = 18 + 4 * K.
00382         MOVE TRANSITION (K, N) TO OUTPUT-COLUMN (I).
00383         MOVE OUTPUT-TEXT TO OUT.
00384         ADD 1 TO K.
00385         IF K NOT > NSV THEN GO TO DISP12.
00386     DISP11. WRITE OUTPUT-LINE AFTER 2.
00387     END-DISP11. MOVE SPACES TO OUT.
00388         WRITE OUTPUT-LINE AFTER 0.
00389     SKIP-DISPLAY.
00390
00391     READ-FIRST-RECORD.
00392         READ SYSINPUT, AT END GO TO FIN.
00393         MOVE 11 TO INPTR.
00394         MOVE SPACES TO BRAILLE-LINES.
00395         MOVE SPACES TO PROOF-LINE.
00396         MOVE ZEROS TO CODE-LINE.
00397         MOVE TEXT TO BUFFER.
00398         GO TO TRANSLATION.
00399     NOTHING. EXIT.
00400     END-INITIALIZATION.
00401
00402
00403     TRANSLATION SECTION.
00404
00405
00406     INSPECT-BUFFER.
00407         MOVE 1 TO LETTER.
00408     IDENTIFY-FIRST-CHARACTER.
00409         IF SYMBOL (LETTER) = RL1 THEN GO TO LOOKUP.
00410         IF LETTER = NALPHABET THEN GO TO ERROR-1.
00411         ADD 1 TO LETTER.
00412         GO TO IDENTIFY-FIRST-CHARACTER.
00413
00414     LOOKUP.
00415         COMPUTE NN = LETTER + 1.
00416         MOVE EXTENT (LETTER) TO INDEX.
00417         MOVE 'N' TO FOUND-SECOND-SYMBOL.
00418     MATCH.
00419         IF INDEX NOT < EXTENT (NN) THEN GO TO MOVE-SINGLE-SIGN.
00420         MOVE 1 TO POINTER.

```



```

00421 COMPARE.
00422     IF POINTER > 9 THEN GO TO TEST-NON-TERMINAL.
00423     IF TABLE-CHAR (INDEX, POINTER) = '$'
00424     THEN GO TO TEST-NON-TERMINAL.
00425     IF RCHAR (POINTER) NOT = TABLE-CHAR (INDEX, POINTER)
00426     THEN GO TO COMPARE-FAIL.
00427     MOVE 'Y' TO FOUND-SECOND-SYMBOL.
00428     ADD 1 TO POINTER.
00429     GO TO COMPARE.
00430 COMPARE-FAIL.
00431     IF FOUND-SECOND-SYMBOL = 'Y'
00432     THEN IF POINTER = 1 THEN GO TO MOVE-SINGLE-SIGN.
00433
00434 MISMATCH.
00435     ADD 1 TO INDEX.
00436     GO TO MATCH.
00437
00438 TEST-NON-TERMINAL.
00439     IF RIGHT-CONTEXT (INDEX) = SPACE THEN GO TO OUTPUT-LOGIC.
00440     MOVE 1 TO N.
00441 TNT1.
00442     IF SYMBOL (N) = RCHAR (POINTER) THEN GO TO TNT2.
00443     IF N = NALPHABET THEN GO TO MISMATCH.
00444     ADD 1 TO N.
00445     GO TO TNT1.
00446 TNT2.
00447     MOVE 1 TO K.
00448     IF RIGHT-CONTEXT (INDEX) NOT = NON-TERMINAL (K)
00449     THEN GO TO TNT3.
00450 TNT5.
00451     MOVE 1 TO J.
00452 TNT4.
00453     IF CHAR-CLASS (N) = RIGHT-CONTEXT-CLASS (K, J)
00454     THEN GO TO OUTPUT-LOGIC.
00455     IF RIGHT-CONTEXT-CLASS (K, J) = 99 THEN GO TO MISMATCH.
00456     ADD 1 TO J.
00457     IF J NOT > 4 THEN GO TO TNT4 ELSE GO TO MISMATCH.
00458 TNT3.
00459     ADD 1 TO K.
00460     IF K NOT > NNT THEN GO TO TNT5 ELSE GO TO MISMATCH.
00461
00462 OUTPUT-LOGIC.
00463
00464     MOVE INPUT-CLASS (INDEX) TO I.
00465     MOVE 1 TO K.
00466 LOGIC1.
00467     IF DECISION (K, I) = '-' THEN GO TO LOGIC4.

```

```

00468         IF DECISION (K, I) = 'G' THEN GO TO TRANSDUCE.
00469         MOVE 1 TO N.
00470 LOGIC2.
00471         IF CONDITION (K, N) = '-' THEN GO TO LOGIC3.
00472         IF STATE-VARIABLE (N) NOT = CONDITION (K, N)
00473         THEN GO TO LOGIC4.
00474 LOGIC3.
00475         IF N = NSV THEN GO TO LOGIC5.
00476         ADD 1 TO N.
00477         GO TO LOGIC2.
00478 LOGIC4.
00479         IF K = NDC THEN GO TO MISMATCH.
00480         ADD 1 TO K.
00481         GO TO LOGIC1.
00482 LOGIC5.
00483         IF DECISION (K, I) = 'Y' THEN GO TO TRANSDUCE,
00484         ELSE GO TO MISMATCH.
00485
00486 TRANSDUCE.
00487         MOVE 1 TO J.
00488 PUT-OUT-SIGNS.
00489         IF SIGN (INDEX, J) = 99 THEN GO TO END-OUTPUT-SIGNS.
00490         MOVE SIGN (INDEX, J) TO OUTSIGN.
00491         PERFORM OUTPUT-SIGN.
00492         ADD 1 TO J.
00493         IF J NOT > 4 THEN GO TO PUT-OUT-SIGNS.
00494 END-OUTPUT-SIGNS.
00495
00496         MOVE 1 TO J.
00497 SHIFT-LEFT.
00498         MOVE RR9 TO TEMP.
00499         MOVE TEMP TO RL9.
00500         PERFORM INPUT-CHAR.
00501         MOVE NXTCHR TO RR1.
00502         IF J = 99 THEN GO TO TRANSITION-LOGIC.
00503         ADD 1 TO J.
00504         IF J NOT > SHIFT (INDEX) THEN GO TO SHIFT-LEFT.
00505
00506 TRANSITION-LOGIC.
00507
00508         MOVE 1 TO N.
00509 LOGIC6.
00510         IF TRANSITION (N, I) = '-' THEN NEXT SENTENCE,
00511         ELSE IF TRANSITION (N, I) = 'S'
00512         OR (TRANSITION (N, I) = 'T' AND STATE-VARIABLE (N) = 'N')
00513         THEN MOVE 'Y' TO STATE-VARIABLE (N),
00514         ELSE MOVE 'N' TO STATE-VARIABLE (N).

```

```

00515         ADD 1 TO N.
00516         IF N NOT > NSV THEN GO TO LOGIC6.
00517         GO TO TRANSLATION.
00518
00519         MOVE-SINGLE-SIGN.
00520         MOVE SINGLE-SIGN (LETTER) TO OUTSIGN.
00521         PERFORM OUTPUT-SIGN.
00522         MOVE CHAR-CLASS (LETTER) TO I.
00523         MOVE 99 TO J.
00524         GO TO SHIFT-LEFT.
00525
00526
00527
00528         ERROR-1.
00529         IF PROOF-OR-BRAILLE = 'P'
00530         THEN MOVE '*NEW CHAR*' TO BUFFER
00531         ELSE MOVE SPACE TO RL1.
00532         GO TO TRANSLATION.
00533
00534
00535         INPUT-CHAR SECTION.
00536
00537         MOVE-NEXT-CHAR.
00538         IF INPTR = 73 THEN GO TO READ-NEW-RECORD.
00539         MOVE CHAR (INPTR) TO NXTCHR.
00540         ADD 1 TO INPTR.
00541         GO TO END-INPUT.
00542         FIN.
00543         PERFORM PAD OUTL TIMES.
00544         PAD.
00545         MOVE 0 TO OUTSIGN.
00546         PERFORM OUTPUT-SIGN.
00547         CLOSE-FILES.
00548         CLOSE SYSINPUT.
00549         CLOSE SYSPRINT.
00550         IF PUNCHED-OUTPUT NOT = 'N' THEN CLOSE PUNCH.
00551         STOP RUN.
00552
00553         READ-NEW-RECORD.
00554         READ SYSINPUT, AT END GO TO FIN.
00555         MOVE CHAR (1) TO NXTCHR.
00556         MOVE 2 TO INPTR.
00557         END-INPUT.  EXIT.
00558
00559
00560         OUTPUT-SIGN SECTION.
00561

```

```

00562 CONTROL-CHARACTER-TEST.
00563     IF OUTSIGN < 64 THEN GO TO NOT-CONTROL-CHAR.
00564     IF OUTSIGN = 66 THEN GO TO END-OUTPUT.
00565     IF OUTSIGN = 69 THEN GO TO BREAK4.
00566     IF OUTSIGN = 65 OR OUTSIGN = 67 THEN GO TO BREAK4.
00567 NOT-CONTROL-CHAR.
00568     IF TABULATE > 0 THEN IF OUTSIGN = 60 THEN GO TO END-OUTPUT
00569     ELSE GO TO TABULATION.
00570     IF OUTSIGN = 68 THEN MOVE 99 TO TABULATE.
00571     IF OUTPTR NOT > OUTL
00572     OR (OUTSIGN NOT = 0 AND OUTPTR NOT = 2 * OUTL)
00573     THEN GO TO SET.
00574     IF OUTPTR = OUTL + 1 OR OUTPTR = 2 * OUTL THEN GO TO BREAK4.
00575     MOVE OUTL TO M.
00576 BREAK1.
00577     IF M = 0 THEN GO TO BREAK7.
00578     IF PROOF (M) = THREE-SPACES THEN GO TO BREAK2.
00579     SUBTRACT 1 FROM M.
00580     GO TO BREAK1.
00581
00582 TABULATION.
00583     IF OUTSIGN = 0 THEN GO TO TAB4.
00584     MOVE 1 TO N.
00585 TAB1.
00586     IF OUTSIGN = DIGIT (N) THEN GO TO TAB3.
00587     ADD 1 TO N.
00588     IF N NOT > 10 THEN GO TO TAB1.
00589     IF PROOF-OR-BRAILLE = 'P'
00590     THEN MOVE '*BAD TAB**' TO BUFFER.
00591     GO TO BREAK4.
00592 TAB3.
00593     IF TABULATE NOT = 99 THEN GO TO TAB2.
00594     COMPUTE TABULATE = N - 1.
00595     GO TO END-OUTPUT.
00596 TAB2.
00597     COMPUTE TABULATE = 10 * TABULATE + N - 1.
00598     GO TO END-OUTPUT.
00599 TAB4.
00600     IF TABULATE = 99 THEN GO TO TAB1.
00601     SUBTRACT OUTPTR FROM TABULATE.
00602     IF TABULATE < 0 THEN ADD OUTL TO TABULATE.
00603     MOVE TABULATE TO N.
00604     MOVE 0 TO TABULATE.
00605     PERFORM SET THRU END-SET N TIMES.
00606     GO TO END-OUTPUT.
00607
00608 BREAK7.

```

```

00609         MOVE OUTL TO M.
00610 BREAK2.
00611         IF M = 40 THEN GO TO BREAK4.
00612         MOVE OUTPTR TO N.
00613         COMPUTE K = N + 40 - M.
00614 BREAK3.
00615         MOVE PROOF (N) TO PROOF (K).
00616         MOVE SPACES TO PROOF (N).
00617         MOVE CODED-SIGN (N) TO CODED-SIGN (K).
00618         MOVE ZEROS TO CODED-SIGN (N).
00619         MOVE 1 TO NN.
00620 BREAK8.
00621         MOVE BRAILLE-SIGN (NN, N) TO BRAILLE-SIGN (NN, K).
00622         MOVE SPACES TO BRAILLE-SIGN (NN, N).
00623         ADD 1 TO NN.
00624         IF NN NOT > 3 THEN GO TO BREAK8.
00625         SUBTRACT 1 FROM N.
00626         SUBTRACT 1 FROM K.
00627         IF N > M THEN GO TO BREAK3.
00628 BREAK4.
00629         IF PUNCHED-OUTPUT = '0' THEN GO TO BREAK6.
00630         IF LINECOUNT < LPG
00631         THEN GO TO BREAK5.
00632         MOVE SPACES TO OUT.
00633         WRITE OUTPUT-LINE AFTER 0.
00634         MOVE 0 TO LINECOUNT.
00635 BREAK5.
00636         IF OUTPTR = 1 THEN IF OUTSIGN = 65 OR OUTSIGN = 67
00637         THEN GO TO BREAK12.
00638         MOVE SPACES TO OUT.
00639         WRITE OUTPUT-LINE AFTER ADVANCING 1.
00640         MOVE 1 TO NN.
00641 BREAK9.
00642         MOVE 1 TO N.
00643         COMPUTE K = 3 * OUTL + 1 - N.
00644 BREAK10.
00645         IF PROOF-OR-BRAILLE = 'P' THEN GO TO BREAK13.
00646         MOVE LINE-CHAR (NN, N) TO OUT-PLACE (K).
00647         ADD 1 TO N.
00648         SUBTRACT 1 FROM K.
00649         IF N NOT > OUTL * 3 THEN GO TO BREAK10.
00650 BREAK13.
00651         IF PROOF-OR-BRAILLE = 'P' THEN MOVE BRAILLE-TEXT (NN) TO OUT.
00652         WRITE OUTPUT-LINE AFTER ADVANCING 1.
00653         MOVE BRAILLE-OVERFLOW (NN) TO BRAILLE-TEXT (NN).
00654         MOVE SPACES TO BRAILLE-OVERFLOW (NN).
00655         ADD 1 TO NN.

```

```

00656         IF NN NOT > 3 THEN GO TO BREAK9.
00657         IF PROOF-OR-BRAILLE = 'B' THEN GO TO BREAK6.
00658         MOVE PROOF-TEXT TO OUT.
00659         WRITE OUTPUT-LINE AFTER ADVANCING 1.
00660         MOVE 1 TO N.
00661 BREAK14.
00662         MOVE CODED-SIGN (N) TO PROOF (N).
00663         ADD 1 TO N.
00664         IF N NOT > OUTL THEN GO TO BREAK14.
00665         MOVE PROOF-TEXT TO OUT.
00666         WRITE OUTPUT-LINE AFTER ADVANCING 1.
00667 BREAK6.
00668         IF PUNCHED-OUTPUT = 'N' THEN GO TO BREAK11.
00669         MOVE CODED-TEXT TO CODED-OUT.
00670         WRITE CODED-OUTPUT AFTER POCKET-SELECT.
00671 BREAK11.
00672         IF OUTSIGN = 69 THEN MOVE LPG TO LINECOUNT ELSE
00673         ADD 1 TO LINECOUNT.
00674         MOVE PROOF-LINE-OVERFLOW TO PROOF-TEXT.
00675         MOVE SPACES TO PROOF-LINE-OVERFLOW.
00676         IF PUNCHED-OUTPUT = 'N' AND PROOF-OR-BRAILLE = 'B'
00677         THEN GO TO BREAK12.
00678         MOVE CODED-OVERFLOW TO CODED-TEXT.
00679         MOVE ZEROS TO CODED-OVERFLOW.
00680 BREAK12.
00681         IF OUTSIGN = 65 OR OUTSIGN = 69 THEN MOVE 1 TO OUTPTR,
00682         ELSE IF OUTSIGN = 67 THEN MOVE 3 TO OUTPTR,
00683         ELSE IF OUTPTR = OUTL + 1 THEN MOVE 0 TO OUTPTR,
00684         ELSE IF OUTPTR = 2 * OUTL THEN MOVE OUTL TO OUTPTR,
00685         ELSE COMPUTE OUTPTR = OUTPTR - M.
00686 SET.
00687         IF OUTSIGN > 64 THEN GO TO END-SET.
00688         IF OUTPTR = 0 THEN GO TO INCREMENT-OUTPTR.
00689         MOVE OUTSIGN TO CODED-SIGN (OUTPTR).
00690         IF OUTSIGN = 0 THEN MOVE 64 TO OUTSIGN.
00691         MOVE DOTS-1-4 (OUTSIGN) TO BRAILLE-SIGN (1, OUTPTR).
00692         MOVE DOTS-2-5 (OUTSIGN) TO BRAILLE-SIGN (2, OUTPTR).
00693         MOVE DOTS-3-6 (OUTSIGN) TO BRAILLE-SIGN (3, OUTPTR).
00694         MOVE PROOF-CHARACTERS (OUTSIGN) TO PROOF (OUTPTR).
00695 INCREMENT-OUTPTR.
00696         ADD 1 TO OUTPTR.
00697         MOVE 0 TO OUTSIGN.
00698 END-SET. EXIT.
00699 END-OUTPUT. EXIT.

```

APPENDIX VIII

LISTING OF TABULAR INPUT

SPACE	14	00
E	01	17
T	01	30
A	01	01
I	01	10
S	01	14
O	01	21
L	01	07
R	01	23
N	01	29
W	01	58
C	01	09
D	01	25
M	01	13
H	01	19
Y	01	61
P	01	15
U	01	37
F	01	11
B	01	03
G	01	27
K	01	05
V	01	39
J	01	26
Q	01	31
X	01	45
Z	01	53
↳ LOGICAL NOT	03	32
◦ PERIODECIMAL	03	40
: COLON	03	18
- HYPHEN (MINUS)	03	36
1	02	01
0	02	26
8	02	19
6	02	11
2	02	03
5	02	17
3	02	09
4	02	25
9	02	10
7	02	27
' APOSTROPHE	01	04
\$ DOLLAR SIGN	03	50
_ ITALICS	13	40
& AMPERSAND	03	47
" DOUBLE QUOTE	03	52
* ASTERISK	03	20
% PERCENT	01	15

< LESS THAN	03	54
> GREATER THAN	03	54
, COMMA	03	02
? QUESTION MARK	03	38
EXCLAMATION PT	03	22
; SEMICOLON	03	06
(LEFT PAREN	03	54
) RIGHT PAREN	03	54
# NUMBER SIGN	03	60
/ SLASH	03	12
= LETTER SIGN	01	48
ACCENT	01	08
END ALPHABET	99	00
\$	14 01	99999999
HIS \$	01 04	00389999
IN \$	01 03	00209999
WAS \$	01 04	00529999
WERE \$	01 05	00549999
BE \$	01 03	00069999
ENOUGH \$	01 07	00189999
E\$	15 01	48179999
EDOM\$	P 01 04	17252113
ED\$	01 02	43999999
ER\$	01 02	59999999
ENCE\$	04 04	48179999
ENESS\$	01 05	17481499
EN\$	01 02	34999999
EAR\$	01 03	17289999
EAB\$	04 01	17999999
EANCE\$	01 05	17401799
EATIDN\$	04 06	17322999
EAND\$	01 04	17479999
EA\$	L 04 02	02999999
EVERY\$	P 06 05	17999999
EVER\$	01 04	16179999
EITHER\$	01 06	17109999
ETHERE\$	01 05	17462399
THERED\$	01 06	46234399
THERE\$	01 05	16469999
THEIR\$	01 05	56469999
THESE\$	01 05	24469999
THEMSELVES\$	01 10	46133914
THENCE\$	01 06	57481799
THE\$	01 03	46999999
THAT 'LL\$	P 06 07	30040707
THAT 'D\$	P 06 06	30042599
THAT\$	01 04	30999999
THIS\$	P 06 04	57999999
THROUGH\$	01 07	16579999
THOSE\$	01 05	24579999
THYSELF\$	01 07	57611199
TH\$	01 02	57999999
TOGETHER\$	01 08	30272399
TODAY\$	01 05	30259999
TOMORROW\$	01 08	30139999
TONIGHT\$	01 07	30299999
TO AND FRO\$	06 06	30210047
TO \$	07 03	22999999
TION\$	04 04	48299999

TIME\$		01 04 16309999
TRED\$		01 03 30231799
TLED\$		01 03 30071799
TLEF\$	L	01 03 30071799
AS\$		15 01 48019999
AND THE \$		07 04 47999999
AND A \$		07 04 47999999
AND OF \$		07 04 47999999
AND WITH \$		07 04 47999999
AND FOR \$		07 04 47999999
AND\$		01 03 47999999
ANTENNA\$		01 05 01293034
ANTERIOR\$		01 05 01293059
ANTE\$		01 04 01293017
ANCE\$		04 04 40179999
ARIGHT\$		01 06 01162399
AR\$		01 02 28999999
AS \$	P	06 02 53999999
ASTHMA\$		01 05 01145713
ATION\$		04 05 32299999
ABOUT\$		01 05 01039999
ABOVE\$		01 05 01033999
AGAINST\$		01 07 01271299
AGAIN\$		01 05 01279999
AFTERNOON\$		01 09 01112999
AFTERWARD\$		01 09 01115899
AFTERE\$		01 05 01113059
AFTERI\$		01 05 01113059
AFTER\$		01 05 01119999
ALLY\$		04 04 04619999
ALWAYS\$		01 06 01075899
ALSO\$		01 04 01079999
ALMOST\$		01 06 01071399
ALREADY\$		01 07 01072399
ALTHOUGH\$		01 08 01075799
ALTOGETHER\$		01 10 01073099
ACROSS\$		01 06 01092399
ACCORDING\$		01 09 01099999
AUNDER\$		01 04 01372925
AIRE\$		01 04 01102317
ATHAM\$		01 04 01301901
AED\$		01 02 01179999
AEAN\$		01 02 01179999
I\$		15 01 48109999
INTO \$		07 05 20229999
INTO\$		01 04 20302199
ING\$		04 03 44999999
INESS\$		01 05 10481499
IN' \$		01 03 10290499
IN\$		04 02 20999999
IN\$	L	01 02 20999999
ITY\$		04 03 48619999
ITSELF\$		01 06 45119999
ITS\$		01 03 45149999
IT'LL\$	P	06 05 45040707
IT'S\$	P	06 04 45041499
IT'D\$	P	06 04 45042599
IT\$	P	06 02 45999999
IETNAMESE\$		01 04 10173029

IMMEDIATE\$		01	09	10131399
STILL'S\$	P	06	07	12041499
STILL\$	P	06	05	12999999
ST\$		01	02	12999999
SAID\$		01	04	14259999
SOMED\$	P	01	05	14211343
SOME\$		01	04	16149999
SO'S\$	P	06	04	14041499
SO\$	P	06	02	14999999
SION\$		04	04	40299999
SHALL\$	P	06	05	41999999
SHOULDER\$		01	06	41510725
SHOULD\$		01	06	41259999
SH\$	P	06	02	14199999
SH\$		01	02	41999999
SUCH\$		01	04	14339999
SPHER\$		01	05	14151959
SPIRIT\$		01	06	56149999
SEVERE\$		01	05	14173959
OF THE \$		07	03	55999999
OF A \$		07	03	55999999
OF\$		01	02	55999999
OUT\$	P	06	03	51999999
OUND\$		04	04	40259999
UGHT\$		01	05	16519999
UNT\$		04	04	40309999
OURSELVES\$		01	09	51233914
OU\$		01	02	51999999
ONEN\$		01	02	21299999
ONER\$		01	04	21295999
ONED\$		01	04	21294399
ONET\$		01	03	21291799
ONESS\$		01	05	21481499
ONESELF\$		01	07	16211199
ONEY\$		01	03	21291799
ONEE\$		01	03	21291799
ONE\$		01	03	16219999
ONG\$		04	03	48279999
OSOME\$		01	04	19141913
OW\$		01	02	42999999
O'CLOCK\$		01	07	21040999
LIKE\$	P	06	04	07999999
LITTLE\$		01	06	07079999
LESS\$		04	04	40149999
LETTER\$		01	06	07239999
LORD\$		01	04	16079999
RIGHT\$		01	05	16239999
RATHER\$	P	06	06	23999999
RAFTER\$		01	04	23011130
RECEIVE\$		01	07	23093999
REACH\$		01	03	23029999
READM\$		01	02	23179999
READJ\$		01	02	23179999
READ\$		01	03	23029999
REALLY\$		01	03	23029999
REALL\$		01	02	23179999
REAL\$		01	03	23029999
REAPP\$		01	02	23179999
REAP\$		01	03	23029999

REAM\$		01	03	23029999
REAVE\$		01	03	23029999
REAU\$		04	03	23029999
REA\$		01	02	23179999
RED\$	L	06	02	23179999
REJOICE\$		01	07	23260999
REJOICINGS\$		01	09	23260927
RECEIVING\$		01	09	23093927
NOT\$	P	06	03	29999999
NAME\$		01	04	16299999
NESS\$		04	04	48149999
NEITHER\$		01	07	29171099
NECESSARY\$		01	09	29170999
WHICH\$	P	06	05	49999999
WHERE\$		01	05	16499999
WHOSE\$		01	05	24499999
WH\$		01	02	49999999
WITH THE \$		07	05	62999999
WITH A \$		07	05	62999999
WITH\$		01	04	62999999
WILL'S\$	P	06	06	58041499
WILL\$	P	06	04	58999999
WREA\$		01	04	58230299
WOULD\$		01	05	58259999
WORK\$		01	04	16589999
WORD\$		01	04	24589999
WORLD\$		01	05	56589999
WEETHEART\$		01	04	58171730
C\$		15	01	48099999
CHILDREN\$		01	08	33299999
CHILD'S\$	P	06	07	33041499
CHILD\$	P	06	05	33999999
CHARACTER\$		01	09	16339999
CHS\$		01	03	33149999
CH\$		01	02	33999999
CONCEIVE\$		01	08	18093999
CONCEIVING		01	10	18093927
COM\$	L	06	03	36999999
CONE\$		01	01	09999999
CON\$	L	06	03	18999999
COULD\$		01	05	09259999
COLONEL\$		01	04	09210721
CAN'T\$	P	06	05	09043099
CAN'S\$	P	06	05	09041499
CANNOT\$		01	06	56099999
CAN\$	P	06	03	09999999
CATION\$		06	04	09013010
CCH\$		01	03	09339999
CC\$	L	04	02	18999999
D\$		15	01	48259999
DAY\$		01	03	16259999
DO\$	P	06	02	25999999
DDAY\$		01	04	25162599
DD\$	L	04	02	50999999
DISH\$		01	04	25104199
DISC\$	P	06	04	25101409
DISPIRIT\$		01	08	25105614
DIS\$	L	06	03	50999999
DINGY\$		06	05	25202761

DECLARE\$		01 07 25090799
DECEIVE\$		01 07 25093999
DECEIVING\$		01 09 25093927
DECLARING\$		01 09 25090727
DER\$	L	06 02 25179999
DED\$	L	06 02 25179999
DENN\$		01 03 25349999
DENY\$		01 03 25349999
DEN\$	L	06 02 25179999
DESHABILLE\$		06 04 25171419
MENT\$		04 04 48309999
MANY\$		01 04 56139999
MUCH\$		01 04 13339999
MUST\$		01 04 13129999
MORE\$	P	06 04 13999999
MOTHER\$		01 06 16139999
MISTER\$		01 06 13101259
MIST\$	P	01 04 13101299
MIS\$		01 03 13101499
MYSELF\$		01 06 13611199
H\$		15 01 48199999
HADD\$		01 04 19015099
HAD\$		01 03 56199999
HAVE\$	P	06 04 19999999
HIMSELF\$		01 07 19131199
HIM\$		01 03 19139999
HEREN\$		01 03 19599999
HERED\$		01 03 19599999
HERER\$		01 03 19599999
HERE\$		01 04 16199999
HERSELF\$		01 07 19591199
YOUNG\$		01 05 16619999
YOURSELF\$		01 08 61231199
YOURSELVES\$		01 10 61233914
YOU'RE\$	P	06 06 61042317
YOUR\$		01 04 61239999
YOU'LL\$	P	06 05 61040707
YOU'VE\$	P	06 06 61043917
YOU\$	P	06 03 61999999
Y AND BY\$		01 08 00036199
PARTAK\$		01 03 15289999
PARTOOK\$		01 03 15289999
PART\$		01 04 16159999
PAID\$		01 04 15259999
PEOPLE'S\$	P	06 08 15041499
PEOPLE\$	P	06 06 15999999
PERHAPS\$		01 07 15591999
PERCEIVE\$		01 08 15590939
PERCEIVING\$		01 09 15590939
UNDER\$		01 05 16379999
UNEA\$		06 03 37291799
UPON\$		01 04 24379999
US\$	P	06 02 37999999
F\$		15 01 48119999
FOR THE \$		07 04 63999999
FOR A \$		07 04 63999999
FORE\$		06 04 11212317
FOR\$		01 03 63999999
FRIENDE\$		01 05 11231034

FRIENDI\$		01	05	11231034
FRIEND\$		01	06	11239999
FROM\$	P	06	04	11999999
FULL	P	01	04	11370707
FUL\$		04	03	48079999
FFOR\$		01	04	11639999
FF\$	L	04	02	22999999
FIRST\$		01	05	11129999
FATHER\$		01	06	16119999
B\$		15	01	48039999
BUT\$	P	06	03	03999999
BBLE\$		01	04	03609999
BB\$	L	04	02	06999999
BEEN\$	P	06	04	03173499
BFFORE\$		01	06	06119999
BECAUSE\$		01	07	06099999
BETWEEN\$		01	07	06309999
BEHIND\$		01	06	06199999
BESIDE\$		01	06	06149999
BEYOND\$		01	06	06619999
BELOW\$		01	05	06079999
BENEATH\$		01	07	06299999
BEGG\$		01	03	03172799
BESS\$		01	03	03171499
BENN\$		01	03	03349999
BENE\$		01	03	03349999
BEDD\$		01	03	03439999
BELL\$		01	03	03170799
BEAU\$		01	03	03029999
BE\$	L	06	02	06999999
BLESS\$		01	05	03401499
BLE\$		04	03	60999999
BLINDE\$		01	04	03072099
BLINDI\$		01	04	03072099
BLINDA\$		01	04	03072099
BLIND\$		01	05	03079999
BY AND BY\$		06	01	03610047
BY THE BY\$		06	08	52460003
BY \$		07	03	52999999
BRAILLE\$		01	07	03230799
BIGH\$		01	04	03102719
G\$		15	01	48279999
GHAM\$		01	04	27190113
GH\$		01	02	35999999
GOOD\$		01	04	27259999
GO\$	P	06	02	27999999
GG\$	L	04	02	54999999
GREAT\$		01	05	27233099
GREA\$		01	04	27230299
KNOWLEDGE\$	P	06	09	05999999
KNOW\$		01	04	16059999
VERY\$	P	06	04	39999999
J\$		15	01	48269999
JUST\$	P	06	04	26999999
QUICK\$		01	05	31059999
QUITE\$	P	06	05	31999999
QUESTION\$		01	08	16319999
X\$		01	01	45999999
Z\$		01	01	53999999

-ENOUGH \$	01 07 32189999
-IN \$	01 03 32209999
-WAS \$	01 04 32529999
-WERE \$	01 05 32549999
-HIS \$	01 04 32389999
-BE \$	01 03 32069999
-US\$	P 01 03 32323799
) \$	01 02 50009999
ooo \$	03 03 04040499
o \$	04 01 50999999
: \$	14 03 18000099
-COM\$	01 02 36099999
-BY\$	01 03 36036199
--\$	14 02 36369999
1\$	02 01 60019999
0\$	02 01 60269999
8\$	02 01 60199999
6\$	02 01 60119999
2\$	02 01 60039999
5\$	02 01 60179999
3\$	02 01 60099999
4\$	02 01 60259999
9\$	02 01 60109999
7\$	02 01 60279999
'COM\$	01 02 04099999
\$L\$	03 02 00659999
\$PG\$	03 03 00699999
\$P"\$	08 03 67389999
\$P\$	08 02 67389999
\$P\$	09 02 67404099
\$P\$	03 02 67999999
\$G\$	05 02 99999999
\$TAB\$	03 04 00689999
\$T\$	01 02 32049999
\$'R\$	03 03 52049999
\$' \$	03 02 32389999
\$"R\$	03 03 52999999
\$" \$	03 02 38999999
/\$ \$	01 02 99999999
\$B\$	03 02 00999999
__\$	12 02 40409999
_ENOUGH \$	01 07 40189999
_IN \$	01 03 40209999
_WAS \$	01 04 40529999
_WERE \$	01 05 40549999
_HIS \$	01 04 40389999
_BE \$	01 03 40069999
&#" \$	08 03 67389999
&# \$	08 02 67389999
&# \$	09 02 67404099
&# \$	03 02 67999999
"\$	10 01 38999999
"\$	11 01 52999999
%" \$	03 01 20209999
%\$	03 01 18159999
<\$	03 01 32549999
>\$	03 01 54049999
END OF TABLE	99 00 99999999
	02

P 14030202
L 01131313 NSV

NIC
RS-R-RRRRRR--RR
SRRS-SRRRRRRRS
----T-----
-----SR---
-----SR--

08 NDTC

Y-G-G-----GGG---N--
---Y-----YN--
-Y-----NN--
---YY-----G-NN--
-----Y-----YN
-----Y-----NY
-----Y-----N-
-----Y-----Y-

o	A	1
o	,	2
o	B	3
o	'	4
o	K	5
o	;	6
o	L	7
o		8
o	C	9
o	I	10
o	F	11
o	ST	12
o	M	13
o	S	14
o	P	15
o	5	16
o	E	17
o	:	18
o	H	19
o	IN	20
o	D	21
o	TD	22
o	R	23
o	45	24
o	D	25
o	J	26
o	G	27
o	AR	28
o	N	29
o	T	30
o	Q	31
o	┘	32
o	CH	33
o	EN	34
o	GH	35
o	-	36
o	U	37
o	?	38
o	V	39
o	┘	40
o	SH	41

o o o OW	42
o o o ED	43
o o o ING	44
o o o X	45
o o o THE	46
o o o AND	47
o o o =	48
o o o WH	49
o o o	50
o o o OU	51
o o o "	52
o o o Z	53
o o o) (54
o o o OF	55
o o o 456	56
o o o TH	57
o o o W	58
o o o ER	59
o o o #	60
o o o Y	61
o o o WTH	62
o o o FOR	63
	64

REFERENCES

1. J. K. Millen, DOTSYS II: Finite-State Syntax-Directed Braille Translation, MTR-1829, The MITRE Corporation, (1970).
2. R. L. Haynes, "Computer Translation of Grade II Braille", Professional Conference on New Processes for Braille Manufacture, 1968, American Printing House for the Blind, Louisville, Kentucky, pp. 1-4.
3. English Braille, American Edition, 1959 (Revised 1966), American Printing House for the Blind, Louisville, Kentucky.
4. U. S. A. Standard COBOL, American National Standards Institute, X3.23-1968.

DISTRIBUTION LIST

INTERNAL

C-01

C. W. Farr

D-06

P. R. Vance

D-07

J. H. Burrows

A. J. Roberts

D-11

J. J. Croke

C. E. Duke

J. F. Jacobs

D-12

C. A. Zraket

D-53

F. Engel

D-63

R. S. Nielsen

D-73

W. Amory

N. A. Anschuetz

E. H. Bensley

J. A. Clapp

T. L. Connors

C. G. Crothers

M. T. Gattozzi

W. R. Gerhart

R. A. J. Gildea (25)

J. B. Glore

O. R. Kinney

E. L. Lafferty

Dr. J. K. Millen (25)

D-73 (cont.)

J. Mitchell

C. M. Sheehan

J. E. Sullivan

N. B. Sutherland

L. M. Thomas

Dr. D. E. Walker

E. W. Williamson

PROJECT

G. Dalrymple, M.I.T.

M. Leonard, M.I.T.

Prof. R. W. Mann, M.I.T.

V. A. Proscia, M.I.T. (50)

EXTERNAL

H. Bassler

Colonial Penn Insurance Co.

112 South 16th Street

Philadelphia, Pennsylvania 19102

Dr. M. P. Boyles (3)

Director, Computer-Braille Project

Instructional Services Center

Atlanta Public Schools

2930 Forrest Hill Drive, S.W.

Atlanta, Georgia 30315

L. L. Clark (2)

Director, IRIS

American Foundation for the Blind

15 West 16th Street

New York, New York 10011

E. L. Glaser

Computation Center

Case Western Reserve University

University Circle

Cleveland, Ohio 44106

DISTRIBUTION LIST (cont.)

EXTERNAL

Dr. C. E. Hallenbeck
Department of Psychology
University of Kansas
Lawrence, Kansas 66044

R. Haynes
American Printing House for
the Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

Dr. K. R. Ingham
Room 20-B-207
Massachusetts Institute
of Technology
77 Massachusetts Avenue
Cambridge, Massachusetts
02139

R. E. LaGrone
IBM Corporation
Federal Systems Division
Department PC4, Room 2P25
18100 Frederick Pike
Gaithersburg, Maryland 20760

Dr. L. Leffler
Applied Mathematics Division
Argonne National Laboratories
9700 South Cass Avenue
Argonne, Illinois 60440

R. J. McNaughton
RCA
Aerospace Systems Division
Burlington, Massachusetts
01801

Prof. A. Nemeth
Mathematics Department
University of Detroit
4001 W. McNichols
Detroit, Michigan 48821

Dr. B. Perella
Department of Defense
Fort George Meade, Maryland

A. Schack
Schack Associates
127 West 12th Street
New York, New York 10011

J. Siems
American Printing House for the Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

Dr. E. J. Waterhouse
Director
Perkins School for the Blind
175 North Beacon Street
Watertown, Massachusetts 02172

V. Zickle
American Printing House for the Blind
1839 Frankfort Avenue
Louisville, Kentucky 40206

FOREIGN

P. W. F. Coleman
4 George Street
Eastleight, Hants
SO5 4 BU, United Kingdom

C. W. Garland
Royal National Institute for
the Blind
224-6 Great Portland Street
London W. 1, England

R. House
Research Group for Quantitative
Linguistics
Sodermlmstorg 8, Stockholm
Sweden

DISTRIBUTION LIST (conc.)

FOREIGN

Dr. G. Lamprecht
Wistfalische Wilhelms Universitat
44 Munster, Germany
Roxeler Strasse 64

B. Lindqvist
De Blindas Forening
Utvechlingsavdelningen
Gotlandsgatan 46
116 65 Stockholm
Sweden

J. Lindstrom
Handikappinstitutet
Pa Iach, 161 03 Bromma 3
Sweden

V. Møkleby
Husby OFF Skole for Blinde
Hovseterveien 3
Oslo 7
Norway

W. Sorke
Deutsche Blindenstudienanstalt
355 Marburg/Lahn
Am Schlag 8
West Germany

J. Vinding
Statens Institut for Blinde
og Svagsynede
Rymarksvej 1
2900 Hellerup
Denmark

Prof. H. Werner
Institut fur Numerische und
Instrumentelle Mathematik
Universitat Munster
Schlossplatz 5
44 Munster, West Germany



