BRAILLE TRANSLATION SYSTEM: 704
~~[illegible]~~. by the
INTERNATIONAL BUSINESS MACHINES
CORPORATION.

IBM

Data Systems Division

Mathematics and Applications Department

Print
655.38
In
International Business Machines
  Corporation.
Braille translation system--M&A-10.

4-5-71 ~~Mike Kabala~~
~~Kohle~~ ~~2900 Forest~~ D.M.
~~271-2480~~

~~JAN 10~~
JUN 1

International Business Machines Corporation
Data Systems Division

MATHEMATICS AND APPLICATIONS DEPARTMENT
1271 Avenue of the Americas
New York 20, New York

Program decks and additional copies of this write-up are available from
Library Services Department
IBM Data Processing Division
590 Madison Avenue
New York 22, New York

BRAILLE TRANSLATION SYSTEM FOR THE IBM 704

produced by
Ann S. Schack
and
R. T. Mertz

assisted by
Fred Brooks

* * * *

Preliminary write-up, 1961

M&A-10

BRAILLE TRANSLATION SYSTEM

A program which converts an ordinary punched card text into the special "shorthand" form of Braille II according to complex rules for contractions and abbreviations, producing an edited output suitable for mechanized Braille publishing.

AVAILABLE TECHNICAL WRITE-UPS OF COMPUTER PROGRAMS PRODUCED BY MATHEMATICS AND APPLICATIONS DEPARTMENT, DATA SYSTEMS DIVISION

Copies of these write-ups are available on request from the

Library Services Department
IBM Data Processing Division
590 Madison Avenue
New York 22, New York

Also available from the same source are Technical Information Bulletins describing some other applications under development or completed by the Mathematics and Applications Department (including the Auto-Promt Numerical Control System, Weather Forecasting, and the Solution of Einstein's Field Equations).

JOB SHOP SIMULATION APPLICATION. This program is designed to pre-test, on a computer-simulated basis, various plans for reducing in-process inventory, for improving order completions, for increasing the utilization of labor and machines, for adjusting the shop to handle anticipated changes in volumes and types of work, etc.

LIMITED INFORMATION ESTIMATION. This program uses advanced statistical techniques for forecasting and estimating purposes, permitting mathematical models structured with many dependent variables interrelated by simultaneous linear regression-like equations.

NON-LINEAR ESTIMATION. Another tool for forecasting and estimating purposes, this computer program allows the use of various non-linear forms in structuring a problem model.

THE MACHINE LOADING PROBLEM. This computer program is designed to allocate products to machines or plants in a way which will satisfy certain types of production requirements at minimum cost or maximum profit.

INTERVAL ARITHMETIC. Developed as an aid for error analysis, this computer technique determines the range within which the real answer is to be found, considering round-off and other types of errors resulting from lengthy calculations.

IBM TAXONOMY APPLICATION. This new experimental technique uses a computer for automatically comparing and classifying great masses of qualitative information by statistical methods.

MULTI-COMPONENT DISTILLATION PROGRAM. This program extends the ability of process engineers to calculate efficient designs and operating plans for fractionating towers.

THE BENEDICT EQUATIONS PACKAGE. These programs evaluate the Benedict-Webb-Rubin equations of state and their derivatives as required in refinery calculations.

FORECASTING BY ECONOMETRIC SYSTEMS. This program provides for the stepwise application of increasingly advanced statistical techniques (including Full-Information Estimation) to mathematical models structured with many dependent variables interrelated by simultaneous linear regression-like equations.

BRAILLE TRANSLATION SYSTEM. For converting an ordinary punched-card text into the special "shorthand" of Braille II, this program handles complex rules for contractions and abbreviations, producing an edited output suitable for mechanized Braille publishing.

SCHEDULING WITH ARBITRARY PROFIT FUNCTIONS. This is a computer method for scheduling the activities of a given facility according to the relative importance or profit potential of each job or task to be handled.

BRAILLE TRANSLATION SYSTEM

TABLE OF CONTENTS

INTRODUCTION

BRAILLE TRANSLATION SYSTEM

This report describes a system for producing Braille books using the IBM 704. The computer has been programmed to translate any text into Braille codes, to edit the resultant Braille, and to produce output in a form suitable for the automatic production of master embossing plates. The two-year period required to train transcribers is dramatic evidence of the complexity of the Braille system, and one of the major reasons for the development of this program. The following section describes the shorthand nature of Braille and the difficulties of the translation task.

The Braille system of raised dot 'printing' which enables the blind person to 'read' using his sense of touch, was first developed by Louis Braille late in the 19th century. Although there have been other systems designed for the same purpose, the Braille system has been generally accepted as a world-wide standard. Each Braille character occupies a fixed space, called a cell, in which dots may be placed in any of six positions. These dots are arranged and numbered as illustrated below:

A portion of the Braille alphabet is shown below to illustrate that there is no relation between the dot configuration and the shape of an ink-print character.

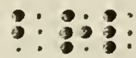
A B C X Y Z

Originally, the Braille code set included only characters for each letter of the alphabet, numbers, and punctuation marks. Books were transcribed letter for letter. In the years since its introduction, many additions and changes have been made to the system, each with a view to simplifying the blind reader's task. Today, the letter-for-letter system, called Grade I Braille, is seldom used. Most books are published in Grade II Braille -- a highly contracted system of representation which utilizes all of the 63 possible dot combinations, many of which have multiple meanings. In the present system (English Braille, American Edition, 1959) there are 189 whole words and letter combinations which may be represented in contracted form. For example, the word PEOPLE is represented by a single code; the syllable ATION, by a double code:


PEOPLE


ATION.

Certain words are always abbreviated: BRAILLE is always written BRL; BLIND, BL.


BRAILLE


BLIND

While the shorthand nature of this system facilitates the reading process, it also makes the transcription task extremely difficult.

The transcription difficulties can best be illustrated by examining a few of the rules governing the use of these contraction codes. A contraction may <u>only</u> be used to represent a given letter combination, such as THE, SOME, etc. However, it may not always be used. There are restrictions based on the position in which these letters occur within a word, the pronunciation of the word, and, in some cases, the meaning of the word or phrase. (In the illustrations which follow, those letter combinations which can be represented by contraction codes are capitalized.)

One of the simpler rules, because the restriction is based on position alone, is the following:

> The contraction COM may only be used at the beginning of a word ... but it need not be a syllable. It must never be used in contact with a hyphen, dash, or apostrophe.
>
> Ex: COMe BEcome home-comING

This is one of the few completely unequivocal rules.

A more typical, and more complex, rule is the following:

> The contractions for BE, CON, and DIS may be used only as syllables at the beginning of a word.
>
> Ex: CONcept cone DISturb disc BErate bell

Additional difficulties are imposed by rules which are based on meaning:

> One-cell whole word contractions (CAN, FROM, YOU, etc.) when separated by a space from other letters or contractions will be read as a word. They may be used when followed by the apostrophe in familiar combinations. However, they should not be used in rare or colloquial forms.
>
> Ex: YOU'll yOU'n

Even more complex is the following rule which determines the translation of a group of words:

> The word signs A, AND, FOR, OF, THE, and WITH should follow one another without a space between if there is no natural pause.
>
> Ex: He walked WITH AND talked WITHTHE boy.

Then some rules which appear relatively straightforward may be contradicted by other rules based on meaning. For example,

> Final letter contractions (FUL, LESS, etc.) should be used in the middle or at the end of the word. They should never begin a word.
>
> Ex: careFUL fulfill

However, even though the FUL combination occurs in the middle of the word UNFULFILLED, its use is prohibited by the following rule:

> A contraction must not be used where the usual Braille form of the base word would be altered by the addition of a prefix or suffix.

And, finally, there is the blanket restriction which contributes most heavily to the transcription problems:

> Contractions forming parts of words should not be used where they would obscure the recognition or pronunciation of a word.

In addition to these rules governing translation, there are rigid format specifications about the number of characters per line, the number of lines per page, numbering of pages, centering of chapter titles, etc. These and other rules will be discussed in detail in Section II of this report. They are quoted here to illustrate the nature of the Braille system and the difficulties facing the transcriber. The editor of The American Printing House for the Blind, the largest Braille publisher in the world, estimates that two years are required to train a skilled Braillist -- that is, someone who works full time producing the metal plates which are used to produce multiple copies of Braille books. After this period, the qualified transcriber can produce about 30 correct pages a day, or about 12 words a minute. Though there are many devoted volunteers who produce single copies, there is a serious problem in hiring and training full-time transcribers for the mass production of Braille texts.

Another important problem facing the Braille publisher is the need to produce books in the shortest possible calendar time. Last-minute decisions about school texts cause particular difficulty. Since proper pagination cannot be established until the entire text has been translated, it is difficult to distribute the work among several transcribers. A third, although less serious, problem is that of error correction. The omission of a single Braille code on the metal master plate requires that at least one whole line of dots be hammered out in order for the correct code to be inserted -- a tedious and time-consuming task.

In cooperation with The American Printing House for the Blind, IBM has designed an automatic system which utilizes the 704 computer for translation and editing Braille books. This work was guided by the following aims:

1. To limit the manual work to those operations requiring little specialized training.
2. To minimize the calendar time required to publish a Braille volume.
3. To simplify the detection and correction of errors.

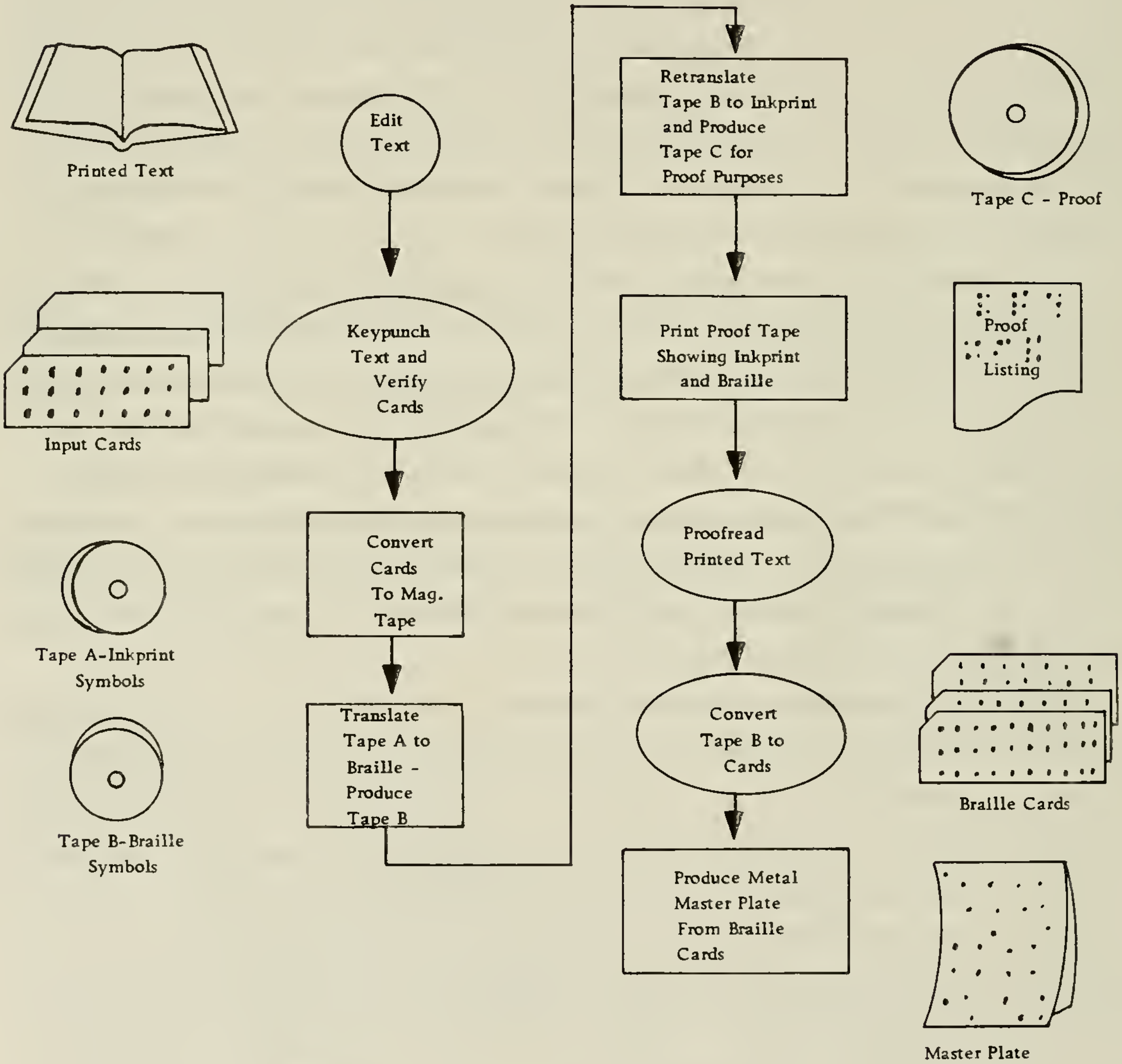
This system represents the most recent effort by IBM to solve some of the problems of Braille publishing. In 1954, IBM engineers designed a Braille writer which produces a paper tape instead of the Braille master plate. This tape, after being key verified, is used to produce the plate automatically, thus eliminating the time-consuming tasks of proofreading and making corrections directly on the metal plates. In the following year, a demonstration program was written for the IBM 701 which indicated the feasibility of machine translation.

While other investigators may also have considered the problem of machine translation of Braille, the only published work known to the writers is that of John P. Cleave, Birkbeck Institute, University of London. Dr. Cleave describes several techniques which would effect machine translation, although he concludes that it would be uneconomical to put these into effect because of the limitations of the computer available to him at that time, and the time required to prepare input for the computer. It is interesting to note that some of the methods proposed by Dr. Cleave are similar to those used in this program, though we were not aware of his work until our program was nearly completed. To our knowledge, the program described here is the only working program for Braille translation. Its efficiency has already been demonstrated in the transcription of 12 Braille books. The system is currently being established as part of the regular procedure at The American Printing House for the Blind. The program, which translates 1000 words a minute, can be run on any 704 with 8192 words of core storage and six tape units. The Printing House personnel will be responsible for preparing the input, editing and producing the master plates.

Though this publication will describe the total system for producing Braille automatically, the emphasis will be on the translation program itself. In particular, the authors will demonstrate how a set of translation rules, designed to be interpreted by a person familiar with English pronunciation, have been restated in a form suitable for machine interpretation. While this is simply a word-for-word translation, many of the problems of editing and dictionary searching are not unlike those which occur in the larger task of mechanical translation from one language to another. The solutions described here may offer some insights to workers in that field.

II. THE SYSTEM

The procedure for transcribing a printed English text into Braille is schematically illustrated below. Manual operations are described within a circle; machine operations, within a rectangle. Each of these will be described in detail in this section.



A. Editing procedure

The manual procedure for publishing Braille requires that the printed text be edited before it is given to the transcriber. The same kind of pre-editing is required for the automatic system. The editor must decide how outlines and diagrams are to be presented, how special type faces are to be represented, what text changes are necessary where references are made to pictures, which portions of the text are in foreign languages, and many other such questions. These decisions are not based on a formal set of rules; they require an understanding of content and the ability to judge what will be easy for the blind reader to comprehend. For this reason, they cannot be made by the computer. It is expected that editing for the automatic system will be no more difficult and require no more time than editing for the manual system.

B. Keypunching and verifying:

The operators who prepare the input cards need know nothing about the Braille system. They simply copy the edited text letter for letter. Some special codes are used to represent those print symbols for which there is no equivalent on the punch keyboard. (For example, a semicolon ; is punched $, .)

Figure A, on the following page, illustrates the card form, or rather the lack of form. Columns 1-72 are used for the text, and words are continued from one card to another without an intervening space or hyphen. The cards may be thought of as discrete pieces of a continuous record, rather than as unit records. (The = sign indicates that the succeeding character is capitalized.)

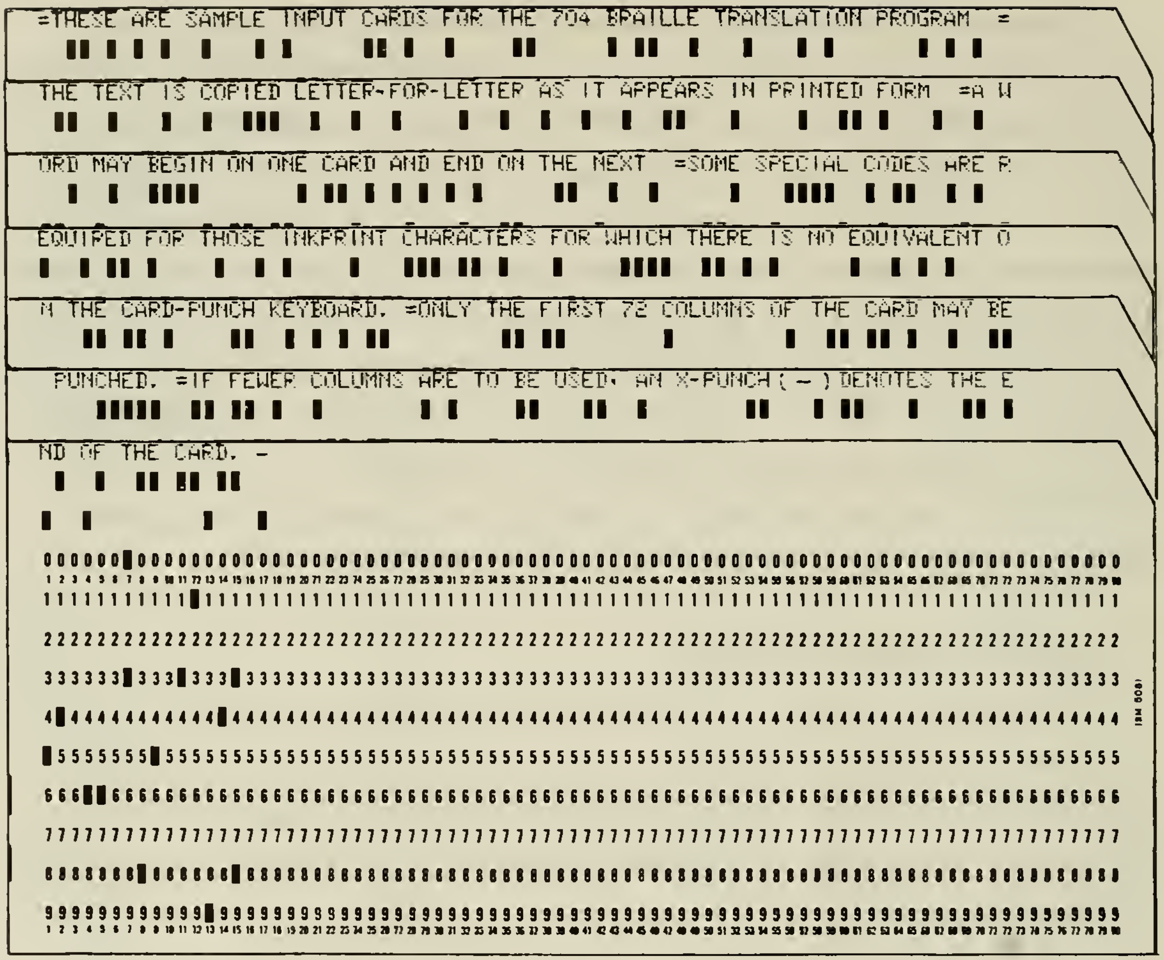

=THESE ARE SAMPLE INPUT CARDS FOR THE 704 BRAILLE TRANSLATION PROGRAM =
THE TEXT IS COPIED LETTER-FOR-LETTER AS IT APPEARS IN PRINTED FORM =A W
ORD MAY BEGIN ON ONE CARD AND END ON THE NEXT =SOME SPECIAL CODES ARE R
EQUIRED FOR THOSE INKPRINT CHARACTERS FOR WHICH THERE IS NO EQUIVALENT O
N THE CARD-PUNCH KEYBOARD. =ONLY THE FIRST 72 COLUMNS OF THE CARD MAY BE
PUNCHED. =IF FEWER COLUMNS ARE TO BE USED, AN X-PUNCH (-) DENOTES THE E
ND OF THE CARD. -
IBM 5081

Figure A

C. Translation Program

This program performs two important functions: the translation of ink print to Braille, and the insertion of special codes which control the make-up of the Braille page. While this latter function is the less dramatic, it is extremely important; without it, this would be an interesting demonstration program rather than an operational system. The program insures that there will be the correct number of characters in a line and the correct number of lines on a page. It provides for centering chapter titles, inserting page numbers, and also handles the special formats required by poetry and outline material.

The translation, one ink-print word at a time, is accomplished by reference to a dictionary stored in memory. This dictionary lists alphabetic and numeric characters, punctuation marks and special symbols, contractible letter combinations and exception words, along with their Braille equivalents. Associated with each item in this list is a set of 'rules' codes which provide information about the limitations on the use of these Braille symbols. Each portion of the ink-print word for which there is a matching dictionary item is called a 'bite.' After each bite is located in the dictionary, the rules codes are tested to determine the legality of that bite. When one ink-print word has been translated, the Braille codes are moved to an output area; and during this transfer, format codes (end of line, end of page, etc.,) are inserted wherever appropriate.

The details of the dictionary arrangement, the techniques used to search for a bite and to evaluate its legality will all be discussed in Section III. A general flow chart will be found in Appendix F.

D. Retranslation Program

This program helps to detect obvious translation errors, to check the validity of Braille codes, and to provide a list of Braille and ink-print symbols in a form suitable for proofreading. (See Figure B.) Unlike the translation from ink print to Braille, the reverse process is unambiguous. For this reason we will not describe this program in detail. (A general flow chart will be found in Appendix F.)

As the sample listing indicates, three lines are required to print the dot patterns for a single Braille line. After these have been printed, each Braille code is translated to ink print. Since the same code may have different meanings depending on its position in a word, there are several dictionaries -- one for initial characters, one for final characters, one for single character words, etc. For example, each of these dictionaries contains a different equivalent for the following Braille code:

⠂

The final character table translates this as a comma; the middle character table, as EA. Should it occur as the first character of a word, it would be translated as XX indicating that it is illegal in that position. As the sample output shows, the ink-print equivalent is printed directly beneath its Braille code. This format calls attention to contraction codes and makes it easy to check on the correctness of their use. Note also that words which are abbreviated in Braille are translated letter for letter (e.g., BRL for BRAILLE, LR for LETTER, etc.). Since additions can only be made to short-form words if the combination could not be mistaken for another word, any incorrect additions should be immediately obvious. (For example, if the abbreviation BL were incorrectly used in the word BLINDED, the resultant translation would be BLED.)

In addition to this translation check, there is also a validity check of each Braille code. The Braille codes produced by the translation program are in seven-bit form and each code contains an even number of bits. The retranslation program checks to see that each code satisfies the even-bit count requirement and prints an error message to indicate invalid codes.

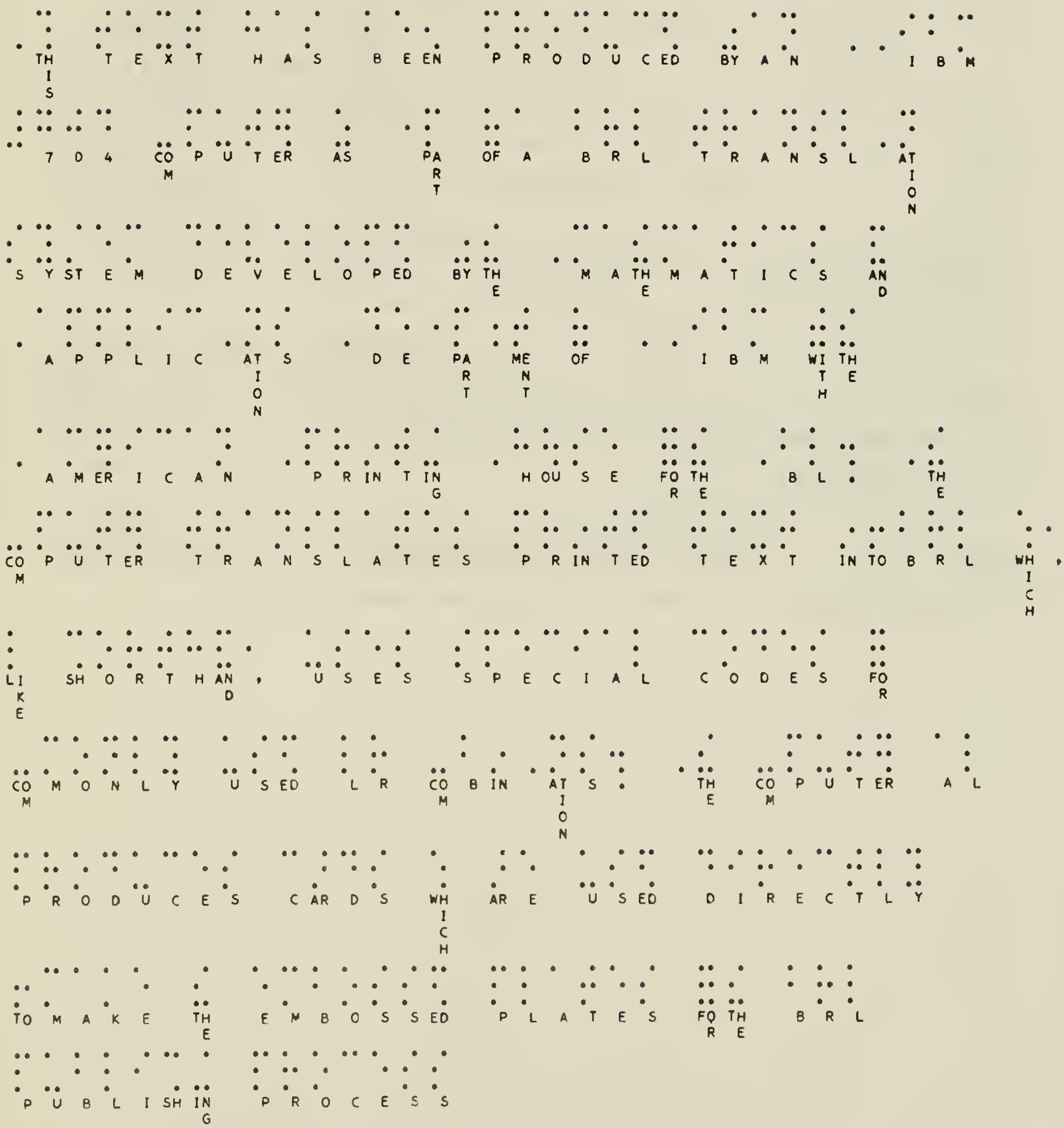

THIS TEXT HAS BEEN PRODUCED BY AN IBM
704 COMPUTER AS PART OF A BRL TRANSLATION
SYSTEM DEVELOPED BY THE MATHEMATICS AND
APPLICATIONS DEPARTMENT OF IBM WITH THE
AMERICAN PRINTING HOUSE FOR THE BL. THE
COMPUTER TRANSLATES PRINTED TEXT INTO BRL WHICH,
LIKE SHORTHAND, USES SPECIAL CODES FOR
COMMONLY USED LR COMBINATIONS. THE COMPUTER AL
PRODUCES CARDS WHICH ARE USED DIRECTLY
TO MAKE THE EMBOSSED PLATES FOR THE BRL
PUBLISHING PROCESS

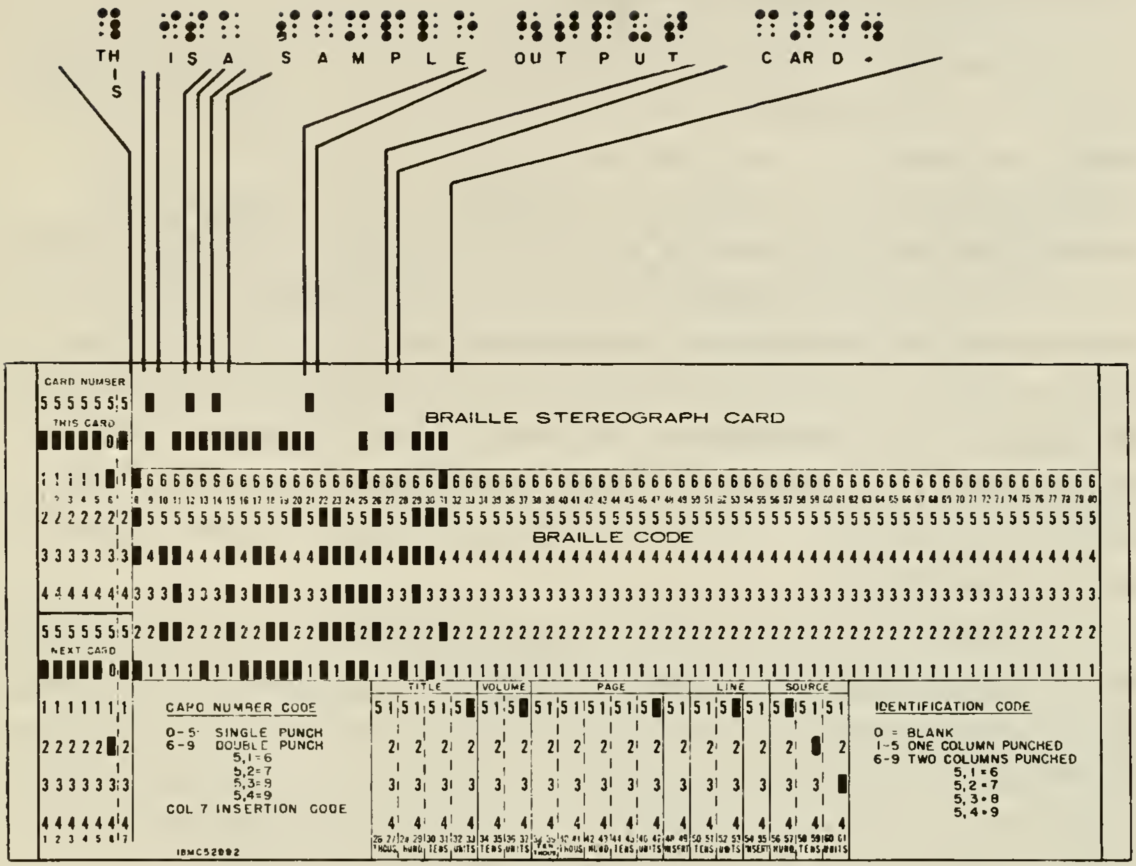
Figure B

E. Proofreading

Although the retranslation program does detect certain errors, it cannot determine whether the use of a contraction 'obscures recognition or pronunciation.' This must be discovered by a proofreader. While the dictionary in the present program has proved adequate to translate the twelve books published to date, it is entirely possible that unusual words will occur in future books for which new dictionary entries will be required. Under the present procedure, a preliminary translation of a portion of the text is printed and proofread before the entire book is translated on the computer. This practice is based on the assumption that each writer uses a characteristic vocabulary and that a sample translation will provide the chance to make any required dictionary additions before translating the entire text.

Additions to the dictionary are made automatically using another 704 program. While it is possible that translation runs may be made at many different locations it is expected that the dictionary up-dating will be handled at a central location.

When a correct proof-listing has been obtained, the output tape produced by the translation program is converted to cards. These Braille cards (see Figure C) are used to produce the master plates.



TH I S
I S A
S A M P L E
OUT P U T
C AR D
CARD NUMBER
THIS CARD
NEXT CARD
BRAILLE STEREOGRAPH CARD
BRAILLE CODE
TITLE
VOLUME
PAGE
LINE
SOURCE
CARD NUMBER CODE
0-5 SINGLE PUNCH
6-9 DOUBLE PUNCH
5,1=6
5,2=7
5,3=8
5,4=9
COL 7 INSERTION CODE
IDENTIFICATION CODE
0 = BLANK
1-5 ONE COLUMN PUNCHED
6-9 TWO COLUMNS PUNCHED
5,1=6
5,2=7
5,3=8
5,4=9
IBMC52092

Figure C

III. THE BRAILLE TRANSLATION PROGRAM

The non-formal quality of the Braille rules, as illustrated in Section I, might suggest at first that they are not adaptable for computer interpretation. While the tests for position validity can be easily programmed, the rules based on pronunciation present a much greater problem. The technique used to solve this problem is an imitation of the one used by the human transcriber. Since there is no set of rules which govern the pronunciation or syllabication of a given word, the Braillist must rely on a dictionary, either in memorized or printed form. The computer has been programmed to follow a similar procedure: a set of programmed tests determines whether the contraction satisfies the position restrictions; a dictionary stored in memory provides information about pronunciation.

The basic unit of translation is one ink-print word, which we define as a string of characters between two spaces, and including the second space. The translation proceeds left to right, and each group of letters for which there is a dictionary entry is referred to as a 'bite.' In some cases, one 'bite' will suffice to translate the entire word; in others, there will be as many 'bites' as there are letters.

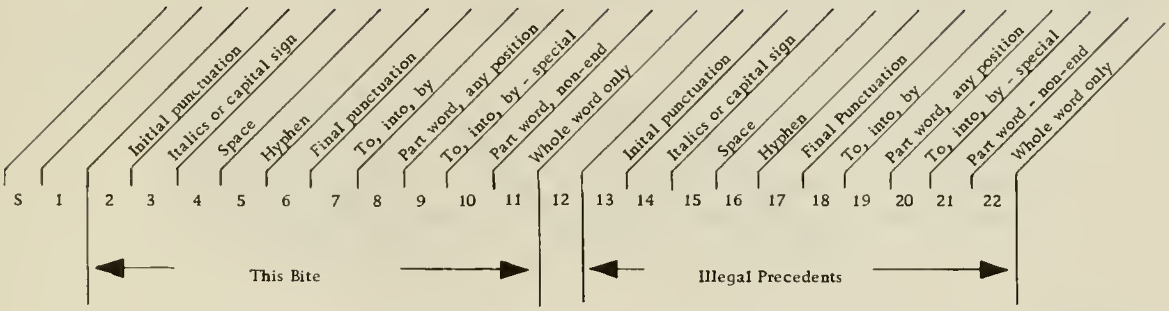
Each dictionary entry, whether it be a single alphabetic character or an exception word, contains three kinds of information: the ink-print symbols, their Braille equivalent, and a set of rules codes. The rules codes occupy one 704 word (36 bits); the space required for the ink-print and Braille codes depends on the length of the particular letter combination. The tests for position validity utilize the rules word, which is functionally divided into several segments. In the following section, they will be discussed in the order in which they are used by the program.

A. Position restrictions

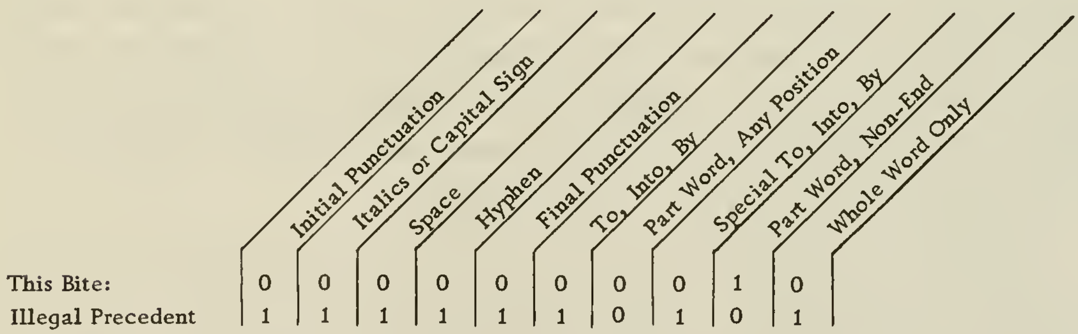
Each bit position of the rules word represents a particular characteristic of the entry. A (1) indicates that the entry has this characteristic; a (0) that it does not. The first segment, bits 2-11, describes the characteristics of this entry, or bite; the second, the characteristics

bits 13-22

of all bites which may not precede this one. (Positions S, 1, and 12 are not used.) The significance of each bit in these two segments is illustrated below:



Thus, each table entry is described in terms of one of ten characteristics. Only one bit is necessary to describe this bite; several bits may be required to describe the illegal precedents. For example, the EA contraction, which may only be used in the middle of a word, is described by the following rule bits:



In other words, the EA contraction may not end a word, and must be preceded by another part word.

To illustrate how these rule bits are utilized, let us consider how the program translates the word SEA. First, the BCD codes for S, E, A, and Space are extracted from the input area and moved to a work region.

Next, the S section of the dictionary is searched to find the largest possible bite of this word. Since there is no entry between SAID and SH, the largest bite in this case is the single letter S. The rule bits for S are:

	Initial Punctuation	Italics or Capital Sign	Space	Hyphen	Final Punctuation	To, Into, By	Part Word, Any Position	Special To, Into, By	Part Word, Non-End	Whole Word Only
This Bite:	0	0	0	0	0	0	1	0	0	0
Illegal Precedent	0	0	0	0	0	0	0	0	0	1

The only illegal precedent is a whole word contraction to which no additions can be made. Since S is the first bite of the word, it is, by definition, preceded by a Space. Therefore, the illegal precedent bits of the S entry are compared with the description of a Space by means of a logical AND instruction. This instruction compares each bit position, substituting a 1 wherever both words contain a 1, and a 0 wherever either word contains a 0. A non-zero result indicates that the preceding bite is one of the illegal ones. Comparing the illegal precedent for S (0000000001) with the description of a Space (0010000000) results in zero, indicating that this is a legal sequence.

The result of the next table search is the EA contraction, with the rule bits indicated above. Comparing the illegal precedents for EA (1111110101) with the S description (0000001000) yields a zero, indicating that this sequence is legal. However, when the illegal precedent bits for the final bite, Space, (0000000010) are compared with the EA bits (0000000010), the result is non-zero, and the position of the non-zero bit indicates that the EA contraction is illegal. Therefore, the EA bite is erased, and the E section of the table is re-entered at the entry following the EA entry. This re-biting process results in E, A, and Space as separate bites, and the rules comparisons indicate that this is a legal sequence. In this case, it was necessary to examine three bites before determining the legality of the EA bite. Had the word been EASY or EAST, the first comparison would indicate that EA is illegal in that position.

While the significance of most of the rule bits is obvious, a word is in order about the bits labeled TO, INTO, BY, and SPECIAL TO, INTO, BY. The rulebook states

> There should be no space between the contractions TO, INTO, and BY and the word which follows if there is no natural pause.... They should not be used as part words or in compound words. They may not be used before any punctuation sign.

	Initial Punctuation	Italics or Capital Sign	Space	Hyphen	Final Punctuation	To, Into, By	Part Word, Any Position	Special To, Into, By	Part Word, Non-End	Whole Word Only
This Bite:	0	0	0	0	0	1	0	0	0	0
Illegal Precedent:	0	0	0	1	1	1	1	1	1	1

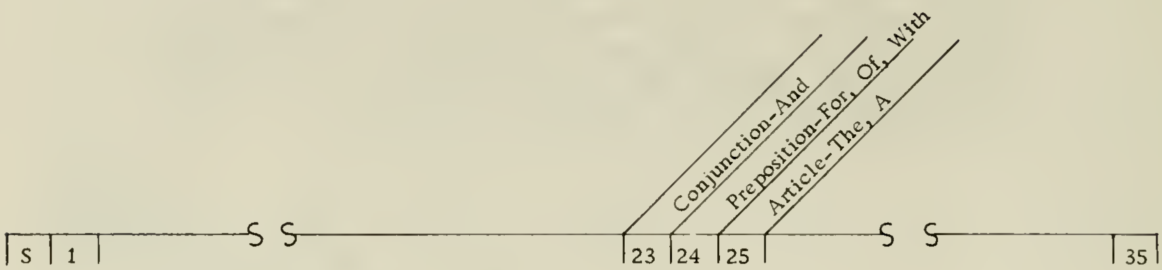
The space elimination is handled very simply: in the dictionary, the Braille equivalent for the sequence T-O-Space is the single code for the TO contraction. Thus, the first character of the succeeding word will be written next to the TO contraction with no intervening space, provided that the rule bits do not indicate that a TO-type bite is an illegal precedent. Since the rule bits for punctuation marks indicate that TO words are illegal precedents, the following sentence will be correctly translated: THEY came to (vERbal) blOWs. Had there been no parenthesis, the TO contraction would have been written next to the V with no intervening space.

The SPECIAL TO, INTO, BY bit identifies these words when they are italicized or capitalized, and is used to effect the following rule:

> The contractions for TO, INTO, and BY should not be used when they are preceded and followed by a capital sign, or when they are preceded or followed by an italic sign.

The rule bits for the italics sign and capital sign indicate that the SPECIAL TO words are illegal precedents.

The remaining segments of the rules word describe other characteristics of the table entry. The first of these consists of three bits which are used to describe only six entries -- the whole words A, AND, FOR, OF, THE, and WITH.



These contractions may be used either as part words, or as whole words. When used as whole words they should

> follow one another without a space, if there is no natural pause between them.

Since we cannot program tests to determine the presence of a 'natural pause,' it is necessary to restate this rule in terms of some formal properties of these words. Analysis of the combinations of the three parts of speech represented (conjunction, preposition, article) reveals that certain sequences cannot be included in one phrase, and therefore should not be written without a space between them. For example, if the conjunction AND is the second word in the series, it is clear that the first word ends a phrase or clause, and that in most cases there should be a 'natural pause' between these two words. (A similar analysis is made to determine whether there is a 'natural pause' between TO, INTO, BY, and the word which follows.)

Some examples of legal and illegal sequences are listed on the following page.

SPACE MAY BE ELIMINATED	SPACE MAY NOT BE ELIMINATED
1. CONJUNCTION, PREPOSITION	1. PREPOSITION, CONJUNCTION
Ex: He looked for pencils AND FOR paper.	Ex: He looked FOR AND found the pencils and paper.
2. CONJUNCTION, ARTICLE	2. ARTICLE, CONJUNCTION
Ex: The boy AND THE girl laughed.	Ex: THE AND an are articles.
3. PREPOSITION, ARTICLE	3. ARTICLE, PREPOSITION
Ex: He walked WITH A cane.	Ex: Put A WITH T to make AT.

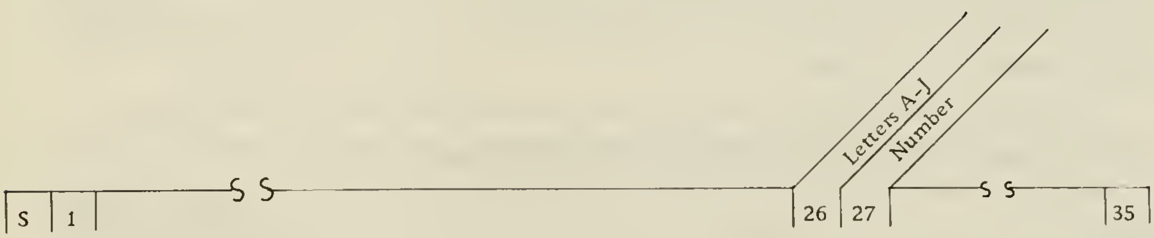
Three bits of the rules word are used to identify the part of speech of the AND-type words. In bit positions 23-25, a conjunction is coded 100, a preposition, 010, and an article, 001. When shifted into the appropriate position, these codes will be interpreted as the decimal numbers 4, 2, and 1. A simple arithmetic test is made to determine whether any pair of these words may be written together without a space:

IF Word 1 minus Word 2 = zero or negative, space is required

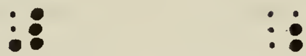
IF Word 1 minus Word 2 = positive, space is eliminated

Longer sequences of these words will be correctly translated by applying this test to each pair within the series.

The next segment, bits 26 and 27, identify the letters A through J and the numbers 0-9.



In ink print, each of these characters has a unique representation; in Braille, however, the first ten letters of the alphabet are also used for numbers. A special Braille code indicates that the characters which follow it are numeric; another indicates that the succeeding characters are alphabetic.



NUMBER SIGN LETTER SIGN

While the number sign is always required before a group of numbers, the letter sign is only required in such ambiguous combinations as 4F, 23A, etc. Since the effect of the number sign is terminated by a space, there is no ambiguity in the following:

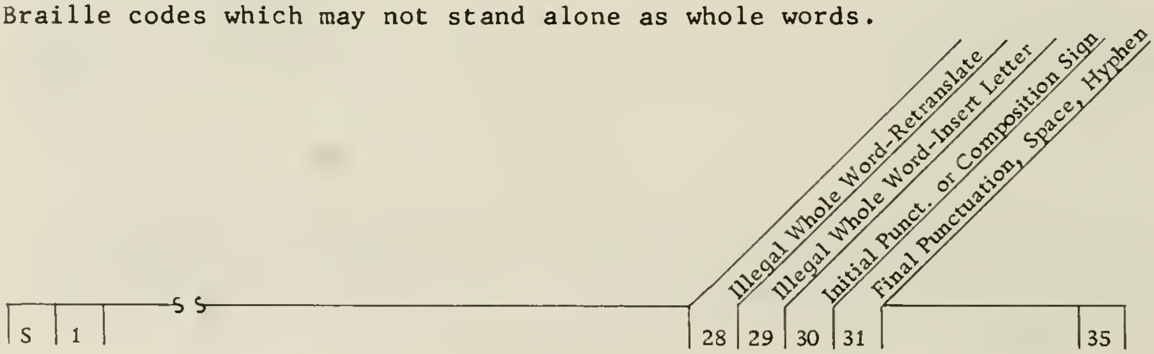
The report card showed 3 A's and 2 B's.

If the first test for illegal precedents is successful, a test is then made to determine whether a letter or number sign must be inserted. In some cases, a 'pseudo' number sign is inserted. This is a special code which has an extra, eighth, bit to distinguish it from real Braille codes. This pseudo-code is used to effect the following rule:

> Although numbers joined by a hyphen do not require a second number sign, if the number is divided at the end of the line after the hyphen, the number sign should be repeated at the beginning of the following line.

Whenever a hyphenated number is encountered, the program inserts the pseudo number sign after the hyphen. After the word has been fully translated, the Braille codes are moved to an output area. A count is made to determine whether an end-of-line or end-of-page code should be inserted. If the pseudo number sign falls in the middle of a line, it is eliminated; if it falls at the beginning of a line, it is converted to the real Braille code.

The next set of bits (28-31) is used to discover the two types of Braille codes which may not stand alone as whole words.



While the information in bits 30 and 31 is also available in the first segment of the rules word, it is summarized here to simplify the programming. Three bites must be examined to determine whether a single bite stands alone as a whole word. Whenever a final punctuation bite is encountered, the program searches backwards and examines bite N-2. If that is initial punctuation, the intervening bite is examined to see whether it is a legal whole word. The first class of illegal whole words, identified by a bit in position 28, includes the part-word contractions for EN, SH, etc. Since the EN contraction standing alone is read as ENOUGH, it may not be used to designate the printers' measure, but must be spelled in full.

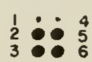
The second class of illegal whole words identified by a bit in position 29, includes all the letters of the alphabet. Each letter, when it stands alone, represents a whole word. The Braille B is read as BUT, C as CAN, etc. If the single alphabetic code is intended to represent the letter rather than the whole word, the rules require that it be preceded by the letter sign. If the three-bite test described above indicates that a single letter stands alone, an additional test is made to satisfy the following exception:

> The letter sign is *not* required when the letter is an initial or abbreviation followed by a period.

Thus, when the final punctuation is a period, the letter sign is not inserted. Of course, since the period represents an abbreviation point as well as a final stop, a problem is presented by sentences of the following type:

> This is the letter B.

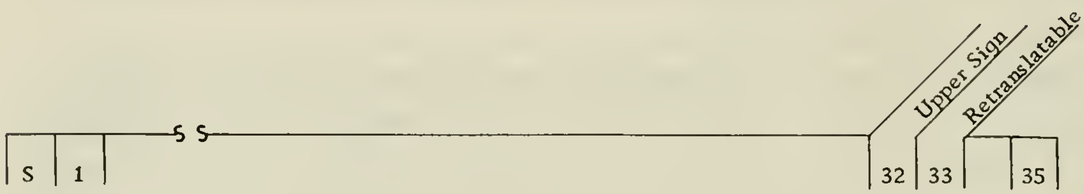
Our assumption is that this condition will occur infrequently in the majority of books to be translated. In the case of childrens' first readers, or mathematical texts where such constructions would be expected, a simple parameter change will accomplish the correct translation.

The final rule handled by the rules word is concerned with lower sign codes. Lower sign is the term applied to Braille codes which occupy only the lower four dots in the cell: 

If many lower signs occur in sequence, it is difficult for the blind reader to maintain the proper orientation within the Braille cell. For this reason the rule states:

> Two or more lower signs must not follow one another when they are not in contact with an upper sign containing a dot 1 or 4. When two or more lower sign contractions follow one another without being in contact with an upper sign, the final lower sign must not be used.

A bit in position 32 identifies all upper signs and position 33 indicates whether this is a retranslatable contraction.



When the legality of each bite has been established, the upper sign bit is stored using a logical OR command. When the word has been completely translated, this location is tested. If there is at least one upper sign in the word, this location will contain a 1. When the location is zero, a backward search is made for the first retranslatable bite. This last bit is necessary since some of the lower signs are punctuation symbols which, obviously, cannot be retranslated.

B. Pronunciation rules

In the preceding section we have shown how the rules word is used to apply those translation rules which are based on the formal characteristics of a given word or letter combination. Pronunciation problems are solved more simply. We have said that the dictionary includes an entry for every letter combination which can be represented by a Braille contraction or abbreviation. These represent only 189 entries. The rest of the dictionary (at present, approximately 600 entries) contains those words or parts of words whose pronunciation prohibits the use of a contraction or abbreviation.

The time required to look up a word is kept to a minimum by using a second, small table which serves as an index to the main dictionary. This index. called the Grade I table, contains a two-word entry for each letter of the alphabet, number, and punctuation sign. (This is the only table which would be required for a Grade I, or letter-for-letter, translation.) The first word of each entry contains the rules pertaining to that character. The second contains the Braille equivalent, and an index used to compute the address of the first entry in the main table which begins with this character. The address of a given character in this index table is computed according to a simple formula which uses the numeric equivalent of the character as a factor.

Each entry in the Grade II table contains the following information:

1. Number of BCD characters in this entry
2. BCD characters for contraction, abbreviation, or exception
3. Rules pertaining to this letter combination
4. Braille equivalent

The rules word is the only portion of the entry which occupies a fixed amount of storage space. The number of 704 words required for the BCD and Braille codes depends on the size of the word or letter combination.

The example below demonstrates the method used to look up a word in the table. If the word to be translated is CHAIR, the C entry in the Grade I table is located and the index stored in the second word of this entry is then used to compute the first address in the C section of the main table. The end of each section in the Grade II table is designated by a -0 entry. Thus, since the first C entry is computed, and the last is indicated by the -0s, it is not necessary to include the C character in the BCD representation of the word or contraction. For example, the entry for CAN contains only the characters A and N. This, of course, saves space as well as look-up time.

Within each section the entries are arranged in alphabetical order which differs from the ordinary alphabetical ordering only in that the blank is considered higher than the letter Z. Shown below is a sample of the C section where the search will be made for the largest possible bite from the word CHAIR. The circles indicate which characters in the table are compared with the input word.

Number of Characters	BCD characters	
5	A N N O T	(CANNOT)
2	A N	(CAN)
1	C	(CC)
8	H A R A C T E R	(CHARACTER)
7	H I L D R E N	(CHILDREN)
4	H I L D	(CHILD)
1	H	(CH)
2	O M	(COM)

The first comparison made is between the number of characters in the word to be translated and the number of characters in the dictionary entry. Only if the input word is larger than or equal to the table entry is a letter-for-letter comparison made. (As illustrated above, the number of characters does not include the C.) For this reason the first entry, CANNOT, is rejected after the numeric test. Since the next entry, CAN, is smaller than CHAIR, the H is compared with A, and since the table character is lower than H, this entry is rejected and the search proceeds to the next entry. A similar result is obtained when the H is compared with C. The next two entries, CHARACTER and CHILDREN are ignored because of their size. The match between the H of CHILD and the H of CHAIR causes a comparison of the next letters. Since the I of the table entry is <u>higher</u> than the A of the search word, this indicates that there are no further entries in the table beginning with CHA. Therefore, a match can only be found for the characters CH. For purposes of continued table search, the word being translated is now truncated to CH, and the search continues looking only at table entries of 1 character. In this case, a perfect match is found in the next table entry, and this is taken as the first 'bite' of the word.

In order to translate the remainder of the word -AIR, the address of the A entry in the Grade I table is computed, and the index found there is used to locate the first A entry in the Grade II table. As it happens, the first entry encountered is the word AND. The comparison of I with N indicates that there is no possibility for a match on more than the first character A. The address of the A entry in the Grade I table, which has been saved, is now referred to, and the Braille equivalent for A is taken as the second 'bite' of the word. The same procedure is followed for the rest of the word, now IR, and, in this case, I and R become the third and fourth bites of the word. (As described earlier, the rules routine evaluates the legality of each bite.)

The portion of the table used to illustrate the translation of CHAIR included only standard Braille contractions and abbreviations. However, most of the table entries represent special cases in which the contractions may not be used.

Some of these special cases are clearly defined by the Braille rules. For example,

> Always use any alternative one-cell contraction in preference to EA and the double letter signs.
>
> Prefer AR to EA: neAR instead of nEAr
> Prefer BLE to BB: bubBLE instead of buBBle
> Prefer ED to DD: pEDdle instead of peDDle

Note that the first two examples require the special table entries EAR, and BBLE. However, since the ED contraction is the first one encountered in the left-to-right translation, it is not necessary to include an entry EDD.

Another clearly defined special case is the following:

> Whenever D, R, or N follows ONE or HERE, the contractions for ED, ER, and EN should be used in preference to ONE and HERE.

This is easily handled by table entries ONED, HERED, HEREN, etc.

Other exceptions are less clearly defined. For example,

> ... a contraction may not be used where it would violate the primary division between a prefix or suffix and the base word.

Since it is not possible to set up specific rules to define 'prefix,' 'suffix,' or 'base word,' we must resort to the use of the dictionary to indicate the words in which contractions may not be used.

For example, let us consider the contraction ER. This symbol may be used in any portion of the word provided that the E and R are in the same syllable. Examination of a dictionary (Webster's New World Dictionary, 1958) indicates that there are many words beginning with ER followed by a vowel,where the E represents a prefix meaning OUT. Rather than put each such word as an exception in the dictionary, we have tried to handle the translation of as many words as possible with a single dictionary entry. In only one case is there a word beginning with ERA which can be translated using the ER contraction. Therefore, our dictionary includes one entry ERA, with rules indicating that when this letter combination occurs at the beginning of a word it should be translated into the three Braille symbols E, R, and A. Another entry, ERATO (the Greek muse), has the Braille equivalent ER, A, T, and O. Thus, the ERA entry takes care of such words as ERADIATE, ERADICATE, ERASE, ERASMUS, etc. The ERATO entry can be considered an exception to the exception. Similarly, the entry ERO will insure the correct translation of such words as ERODE, EROGENOUS, EROSION, EROTIC. Among the ERE words, only those which begin EREC may not use the ER contraction (ERECTILE, ERECT, etc.), while the others (EREMITE, ERETHISM, etc.) may use the ER symbol. The two entries ERUD, and ERU will correctly translate such words as ERUDITE, ERUDITION, and ERUPT, ERUCT. The Braille equivalent for the former is ER, U, D, and for the latter, E, R, U.

These entries by no means exhaust the cases in which the ER contraction may not be used. They cover only those cases in which ER occurs at the beginning of a word. Another entry is required to translate a word such as ANTEROOM. (As a matter of fact, the single table entry ANTE handles such words as ANTEROOM, ANTEDATE, ANTENATAL, in which the ER, ED, and EN contractions may not be used.)

Another kind of exception is the following:

> A contraction must not be used where the usual Braille form of the base word would be altered by the addition of a prefix or suffix.

In other words, since the EA contraction may only be used in the middle of a word, it cannot be used in the word EASY. According to this rule, then, it may not be used in the word UNEASY, either. A single entry, UNEA will govern the translation of UNEASY, UNEARTHLY, etc.)

No claim is made that the present dictionary will translate perfectly any word which will be encountered; nor could such a claim be made even if there had been a detailed analysis of the most comprehensive dictionary. The English language is constantly changing and it is used very freely by many authors. Hence, any type of machine dictionary will have to be updated periodically. A portion of the table is shown in Appendix G. Since it is expected that the American Printing House for the Blind will be responsible for updating the dictionary, this report does not include a description of the Table Generator program.

C. Format Control

The table serves a very important second function -- that of format control. Unlike ink-print books, the composition of Braille books is very rigidly specified. There are detailed rules about the number of lines on a page, characters on a line, the treatment of chapter headings, tabular material, poetry, etc. Technical books and childrens' books present special format problems. The decisions about where to insert end-of-line and end-of-page codes are programmed; initialization cards provide information about the maximum number of lines and characters for a given volume. Other format decisions, however, must be based on information provided by the editor. (It is not possible, for example, to program the computer to distinguish poetry from prose except by editorial codes.)

Wherever the editor notes that special spacing is required, the keypunch operator punches a special code which functions as a signal to the computer program. For example, chapter titles are preceded by the symbol $CHAP and followed by the symbol $CHEND. The computer is programmed to translate the chapter title into Braille and then insert the spaces required to center it properly. Foreign language passages (which must be translated letter for letter) are indicated by the symbols $IG1 (Initiate Grade I) and $TG1 (Terminate Grade 1).

The technique used to implement these format rules provides great flexibility, since a change in rules requires only a simple table change. For each format code there is a dictionary entry, identical in form to those described in the preceding section. However, in format entries the Braille equivalents are 'pseudo' codes which are interpreted by the program as signals to insert special format control codes. When an input word has been completely translated, the Braille codes are moved from a temporary list to the output record area. As each code is transferred, it is examined to determine whether it is a 'real' or 'pseudo' code. (As we have indicated earlier, 'pseudo' codes contain an extra, eighth, bit.) Real codes are converted to a form suitable for writing; pseudo codes are interpreted by the program and the appropriate functional codes are inserted.

For example, the code $SKIP is punched by the operator to indicate that a blank line should be left between two lines of Braille text. (It should be noted that there is no relation between the end of an ink-print line and the end of a Braille line.) There are two 'pseudo' Braille codes stored with the $SKIP entry: a conditional end-of-line Type 1 and Type 2. These are interpreted by the program as follows:

Type 1. If the Braille line preceding the line to be skipped has not been filled, insert an end-of-line code. If the preceding line has been filled and an end-of-line code already inserted, ignore this.

Type 2. If the line to be skipped turns out to be the first line of a page, ignore this code. If it occurs any place else, insert an end-of-line code.

In other words, the first code insures that the present line will be ended; the second, that a line will be left blank providing it occurs in the middle of a page.

Tables and diagrams are also handled by table entries. As we have indicated in the Section II, these must be hand transcribed since they require decisions which cannot be programmed. The editor notes, at the appropriate point in the ink-print text, how many lines or pages must be reserved for a table or diagram. The keypunch operator punches $RTL or $RTP (Reserve Table Lines, or Pages) followed by the number indicated in the text. By convention, all tables begin at the top of a page. Therefore, the pseudo Braille codes stored with these entries cause the required space to be reserved, starting at the top of the page following the point at which the code is first encountered. Because diagrams are hand drawn using a stylus, the reverse side of the page may not be used. Therefore the $RFL code (Reserve Figure Lines) becomes effective at the beginning of the first odd-numbered page after it has been encountered, and the following page is left completely blank.

The Table method of handling format problems permits revisions to be made simply. For example, the paragraphing procedure for most Braille books requires that a paragraph begin on a new line and be indented two spaces. For certain volumes, where space saving is important, three blank spaces are left within a line to indicate a new paragraph. A simple table change enables the program to handle either form. Format entries are added or deleted by means of the table updating program described earlier.

REFERENCES:

1. Cleave, John P., Braille Transcription, Mechanical Translation, Vol. 2, No. 3, (December, 1955), pp. 50-54.

2. Cleave, John P., The Mechanical Transcription of Braille, Mechanical Resolution of Linguistic Problems, Booth, A., Brandwood, L., Cleave, J.P., Academic Press, N.Y., 1958, pp. 97-109.

Appendix A
PRE-EDITING PROCEDURE

The Braille editor, or someone else thoroughly familiar with Braille, must read the ink-print text thoroughly and annotate it for the keypunch operator. The following decisions must be made:

a. Are there pictures which are to be included? Such pictures will be hand drawn. The editor must determine how much space they will occupy and make note of the number of lines to be left available on the page.

b. What text changes are necessary where references are made to pictures which have been deleted?

c. How are tables to be represented? In some cases, information presented in tabular form in ink print should be written in paragraph form in Braille. In other instances, the columnar form should be retained. The editor decides which form shall be used, manually transcribes the information and indicates on the ink-print copy how many lines or pages should be left blank to accommodate the table.

d. Are there special format considerations? Although, in most cases, space-saving is desirable in transcribing Braille, there are some books in which it is necessary to leave extra lines or spaces blank. In addition to noting where lines should be left blank, the editor should also indicate whether sub-headings in the text should be treated as separate lines, how titles and subtitles are to be capitalized, where italics occurring in ink print should be retained in Braille, where poetry passages begin and end, etc.

e. How should the book title be represented at the top of each odd-numbered page? In those cases where the Braille equivalent of the book title takes up more than one line, the editor must decide how the title is to be abbreviated and should transcribe this manually.

f. How should the Braille edition be divided into volumes? Since it is desirable to end each volume at some logical breaking point in the text, the editor must indicate where this is to be done.

g. Are there foreign words or phrases in an English text? These should be noted so that the program will cause them to be translated on a letter-for-letter basis.

h. Are footnotes to be included in parentheses within the text, or should they appear separately at the end of the book? This decision is based on the length of the particular footnote.

Appendix B

KEYPUNCHING INSTRUCTIONS

1. GENERAL INSTRUCTIONS

The printed text and editorial annotations should be copied letter for letter. There are special codes for those ink-print symbols for which there is no equivalent on the keypunch keyboard. These are listed at the end of this section.

Columns 1-72 of the card are used. Identification information will be gangpunched in columns 73-80 after the cards have been verified. If fewer than 72 columns are to be used, an X-punch following the last significant column denotes the end of the card.

The cards are read by the translation program as if they form a continuous record. Therefore, a word may begin on one card and end on the next with no space or hyphen in between. If a word ends in column 72, a space must be keyed in column 1 of the following card.

Hyphenated words which are genuine compound words should be copied exactly as they appear in the text. However, the hyphen which is used to continue a word from one line to the next should be ignored.

2. CAPITALIZATION

When an entire word is capitalized, precede the word by a double capital sign: = =. There should be no space between the capital sign and the first letter of the word.

When only the initial letter of a word is capitalized, precede the word with a single capital sign: =.

When a portion of the word is capitalized, the double capital sign is used to indicate the beginning of the capitalized portion. The end of the capitalized portion is indicated by $TC (Terminate Capital).

When the capitalized portion of the word begins in the middle, the hyphen should precede the double capital sign. When the final portion of the word is capitalized, the $TC is not required.

Examples:	DIStinguish	==dis$TCtinguish
	disTINguish	dis-==tin$TCguish
	distinGUISH	distin-==guish

Note: Words which are capitalized in the printed text are not always so represented in Braille. Follow the editor's instructions.

3. ITALICS

When 1, 2 or 3 italicized words occur in sequence, precede each with a single italics sign: $I.

When 4 or more italicized words occur in sequence, precede the first with a double italics sign ($II), and the last with a single italics sign ($I).

When several italicized paragraphs occur in sequence, precede each paragraph with a double italics sign, and the last word of the last paragraph with the single italics sign.

When a portion of the word is italicized, the single italics sign ($I) is used to indicate the beginning of that portion, and $TI (Terminate Italics) is used to indicate the end. When the italicized portion is at the end of a word, the $TI is not required. (See examples under CAPITALIZATION.)

Note: Words which are italicized in ink print are not always so represented in Braille. Follow the editor's instructions.

4. PARAGRAPHING

Begin each paragraph on a new card.

End the current card. Key a space following the last character, and then an X-punch if fewer than 72 columns are used.

Begin the new card with $PAR followed by one space. Then punch the text of the paragraph.

Some ink-print texts capitalize the first word or words to indicate the beginning of a paragraph. In these cases, follow the procedure indicated above, and capitalized only the first letter of the first word.

5. CHAPTER TITLES

Begin each chapter title on a new card.

End the current card. Key a space following the last character, and then an X-punch if fewer than 72 columns are used.

Begin the new card with $CHAP followed by one space. Punch the chapter title. Follow the title with $CHEND followed by one space and an X-punch if fewer than 72 columns are used.

Note: When chapter title and number are on different lines in the ink-print text, treat as two separate titles.

6. SPECIAL FORMATS

a. Subheadings:

The editor will note which subheadings are to be centered. In those cases:

End the current card.

Begin the next card with $HEAD. Punch the heading and follow it with $HDEND followed by one space.

End the card with an X-punch if fewer than 72 columns are used.

b. Blank lines:

The editor will note where blank lines should separate portions of the text. In those cases:

Key a space following the last character of the line.

Punch $SKIP followed by one space.

End the card with an X-punch if fewer than 72 columns are used.

c. Poetry:

Poetry should be preceded by $IPO followed by one space.

The end of each line of poetry should be indicated by $EL followed by one space.

The end of the poetry passage should be indicated by $TPO followed by one space.

d. Tables and diagrams:

Tables and diagrams will be transcribed manually. The editor will indicate how many pages or lines they will occupy.

If a table occupies less than one page, the following symbols should be used:

5 lines or less	$RTL5
6 to 10 lines	$RTL10
11 to 15 lines	$RTL15
16 to 20 lines	$RTL20

If more than 20 lines are required for a table, one page should be reserved using the symbol $RTP.

In the case of diagrams, the code is $RFL (Reserve Figure Lines) followed by one of the following numbers: 5, 10, 15, 20 or 25.

e. Foreign language passage:

The editor will note which passages are to be so treated. Precede the passage with $IG1 (Initiate Grade 1) followed by one space. Follow the passage with $TG1 (Terminate Grade I) followed by one space.

KEYPUNCH SYMBOLS - BRAILLE TRANSLATION PROGRAM

Ink Print	Fortran Keyboard	Standard Keyboard
Capital (single letter)	=	#
Capital (whole word)	= =	##
Italics (three words or less)	$I	$I
Italics (more than three words)	$$I	$$I
Accent	$A	$A
Poetry	$IPO	$IPO
Period	.	.
Comma	,	,
Semicolon	=,	#,
Colon	=.	#.
Dash	--(8-4)	@@
Question mark	$Q	$Q
Exclamation point	$X	$X
Double quotes (left)	=(	#%
Double quotes (right)	=)	#¤
Single quote (left)	+(	&%
Single quote (right)	+)	&¤
Left parenthesis	(	%
Right parenthesis	)	¤
Left bracket	$(	$%
Right bracket	$)	$¤
Ampersand	AND	AND
Apostrophe	$,	$,
Asterisk	*	*
Degree symbol	$DG	$DG
Dollar sign	$DOL	$DOL
Decimal point	$DEC	$DEC
Ellipsis	...	...
Equals	$	$#
Hyphen	-(8-4)	@
Number sign	$N	$N
Paragraph symbol	$P	$P
Percent sign	$PC	$PC
Section symbol	$S	$S
Slash, fraction mark	/	/

Format Control Codes:

$EL	End line
$SKIP	Skip one line
$PAR	Start new paragraph
$CHAP	Start new chapter; this code indicates the beginning of the chapter title
$ECH	Indicates the end of the chapter title
$IPO	Initiate poetry format
$TPO	Terminate poetry format
$HEAD	Indicates the start of a subheading
$HDEND	Indicates the end of subheading
$RTLn	Reserve n lines for table
$RTPn	Reserve n pages for table
$RFLn	Reserve n lines for figure or diagram
$IG1	Initiate Grade I (letter-for-letter) translation
$TG1	Terminate Grade I translation
$PAGNO	Indicates page number references in text; should be punched in place of the number shown

Appendix C

OUTPUT CARD FORM

The output card form, (see page 41), has been designed to correspond to the paper tape which is currently used to control the stereograph machine. For this reason, as with the input cards, the output card should not be considered as a unit record, but as a piece of a continuous tape.

Three different coding systems are used for this card:

1. Columns 1-7: These columns are used to check the sequence of the cards as they are fed through the card-controlled stereograph equipment. The upper half of these columns contains the Number of This Card; the lower half contains the Number of the Next Card. The card numbers produced by the 704 will usually have a zero in column 7, thus permitting the insertion of as many as nine cards while maintaining the correct sequence. Whenever a card is inserted, the preceding card must be changed so that the Number of Next Card will correspond to the number of the inserted card.

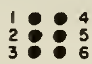
   The coding system used in this section of the card requires that the numbers 0-5 be represented by a single punch; the numbers 6-9, by a double punch. The numbers referred to in this section refer to the numbers printed on the face of this card form, and do not correspond to the numbers on the standard keyboard. Obviously, it will be necessary to exercise great care when making manual corrections.

2. Columns 26-61, lower four rows: These columns contain identification information and are included primarily for the operator's convenience. The numbers 1-5 are represented by a punch in a single column; the numbers 6-9, by one punch in each of two columns; zero, two blank columns. The page and line numbers

punched in the card refer to the first Braille code in the card. Thus, if both lines 24 and 25 are included on one card, the end of page message which prints on the proof listing will show the line number as 24. Here, too, there is no correspondence between the numbers on the face of the card, and the standard keyboard.

3. Columns 8-80, upper 8 rows: These columns contain the Braille code representation, plus a validity check bit, and several functional codes which control the operations of the stereograph. Row X, unlabeled on this form, is used as a validity check bit for those Braille codes which have an odd number of bits. (The stereograph will stop if there is an odd number of punches in a column.) Row 12, also unlabeled here, is used for functional codes, e.g., end of line, end of page, end of card. The space code, represented in Braille by a blank cell, is punched as an X-12 combination in order to preserve the validity check.

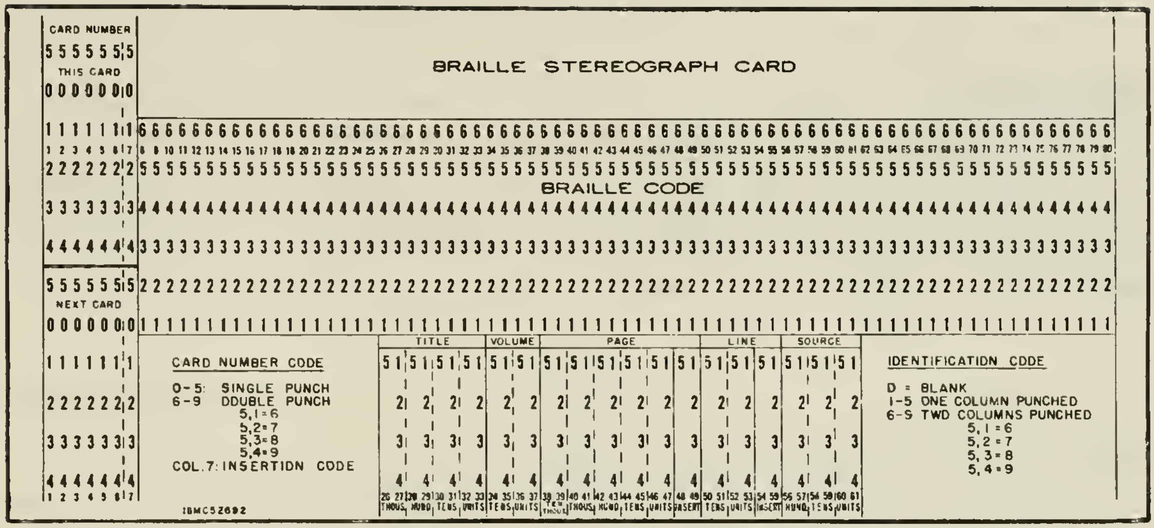
The rows labeled 1-6 correspond to the traditional numbering scheme for the Braille dots:



Someone familiar with Braille should be able to read this section of the card with little trouble.

Although there is room on the form, the output cards presently produced will be blank beyond column 72. An end-of-card code is the last code in each card. In order to facilitate error correction, each paragraph and each page begins on a new card.

It is important to note that these cards cannot be duplicated on the standard keypunch.


CARD NUMBER
THIS CARD
NEXT CARD
BRAILLE STEREOGRAPH CARD
BRAILLE CODE
TITLE
VOLUME
PAGE
LINE
SOURCE
CARD NUMBER CODE
0-5: SINGLE PUNCH
6-9 DOUBLE PUNCH
5,1=6
5,2=7
5,3=8
5,4=9
COL.7: INSERTION CODE
IDENTIFICATION CODE
0 = BLANK
1-5 ONE COLUMN PUNCHED
6-9 TWO COLUMNS PUNCHED
5,1=6
5,2=7
5,3=8
5,4=9
IBMC52692

Appendix D

PROOFREADING AND ERROR CORRECTION PROCEDURES

PROOF LISTING:

The proof listing (Fig. D) shows the ink-print representation of the Braille codes, and the BCD characters directly beneath the corresponding Braille. The ink-print equivalent of a contraction code is spelled in full and may occupy several lines:

Where abbreviated forms appear in the text, the ink print shows a letter-for-letter representation of the Braille.

This, it is hoped, will help call attention to any incorrect use of the short-form words. For example, if the abbreviation for BLIND (BL) were incorrectly used in the word BLINDED, the text would read BLED.

If a contraction were used in the wrong position of a word (such as the use of the EA contraction in the word EAST), the ink-print translation of that contraction will be XX.

Should there be an invalid Braille code or format code, an error message will print below the line containing the error, and the particular code will be underlined. (See line 5.)

A message is printed at the end of each Braille page, indicating the page number, and the number of the card on which the page ends.

While this retranslation program does detect invalid codes, and certain misuses of contractions, there are other errors which can only be detected by someone familiar with the Braille system. Therefore, this listing should be carefully proofread. After this automatic system for translating Braille has been in production for a while, it is reasonable to assume that most of the problem words will have been encountered and provided for in the dictionary. At that point it may be possible to proofread sample chapters rather than the whole book. It is too early to use this procedure now.

ERROR CORRECTION:

Certain errors can be corrected manually with a minimum of effort; others may require a rerun of a portion of the text. An example of an error which can be easily corrected is shown on line 6 of the sample listing. (Fig. D.) Apparently, there is an error such that the validity check bit was not included for the Braille code OF. Since that same Braille code appears correctly in other lines of the listing, we know that there is no error in the table of Braille codes. The error message which prints below line 6 indicates the code in error. (The card number which appears in the error message is the card which contains the last character of the line. Examination of the cards shows that the error actually occurred in column 67 of card 3.) As shown in Fig. E, the card can be corrected by adding an X-punch in column 67.

Another easily corrected error is shown on the last line. Because of a keypunching error, the word ERRORS is incorrectly spelled ERROORS. In this case, no contraction code is involved, and the extra O (column 20, card 4) can be eliminated by punching the delete code in that column. (See Fig. F.) The delete code (all 8 rows punched) causes the stereograph to space over that column.

Because a contraction code is involved, correcting the error on line 3 is slightly more difficult. Because the word PROGRAM contains an extra

G, the contraction for GG was punched in column 53 of card 2. The delete code cannot be used in this case. Rather the code for a single G must be substituted for the GG code. As it happens, the error on line 2 (TRASLATION instead of TRANSLATION) also occurs in card 2. Since this involves the addition of a character, the information on card 2 must be reproduced into two cards. As shown in Fig. G, the first of these contains the Braille codes up to the error point, column 44. The Braille N has been punched in column 45, and an end-of-card code in column 46. The second half of card 2 is reproduced into another card, and the G code substituted for the GG. Note that these two cards have been numbered so that the sequence is not interrupted. The extra card is numbered 2.1.

In cases where a number of words have been omitted, it may be possible to make a manual correction, provided that the correction does not interfere with the pagination established by the program. Where page numbers will have to be adjusted to permit the insertion of a number of words, it will probably be desirable to rerun a portion of the book. In some cases, words which have not been provided for in the dictionary will be mistranslated. Where their frequency is low, it may be possible to make manual corrections. However, where there is more than one such word, or when it is used frequently throughout the text, it will probably be necessary to rerun. At the outset, it is suggested that one or two chapters be translated first to see if the particular author's vocabulary will be handled by the existing dictionary. This procedure should cut down the number and size of the reruns required.

```
                              S  A  M  P  L  E                                                    1
 TH   I S    A    S A M P L E   OF TH   P R O OF   L I ST IN   P R E P AR ED
 I                                    E                        G
 S
BY TH      7 0 4   CO P U T ER   AS    PA   OF TH   B R L   T R A S L   AT
   E               M                   R       E                         I
                                       T                                 O
                                                                         N
P R O GG R A M .   ER R O R S   HA   B E EN   D E L I B ER A T E L Y
                                V
                                E
IN T R O D U C ED   TO D E M O N ST R A T E   TH   U S E   OF   TH
                                              E                 I
                                                                S
R E P O R T   AN TH   M E TH O D   OF   C O R R E C T IN   S CH
              D  E                 --                  G
  CARD NUMBER     4.0     PAGE NUMBER     1.0     LINE NUMBER     6.0     VALIDITY CHECK
ER R O O R S .
```

Figure D

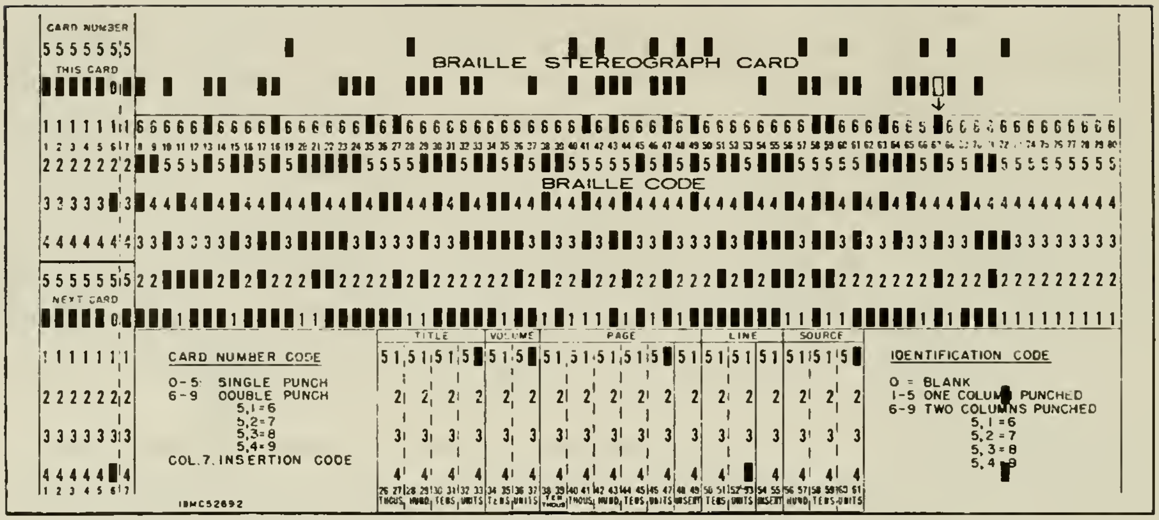


Figure E

Validity check error in column 67 can be corrected by adding an X-punch.

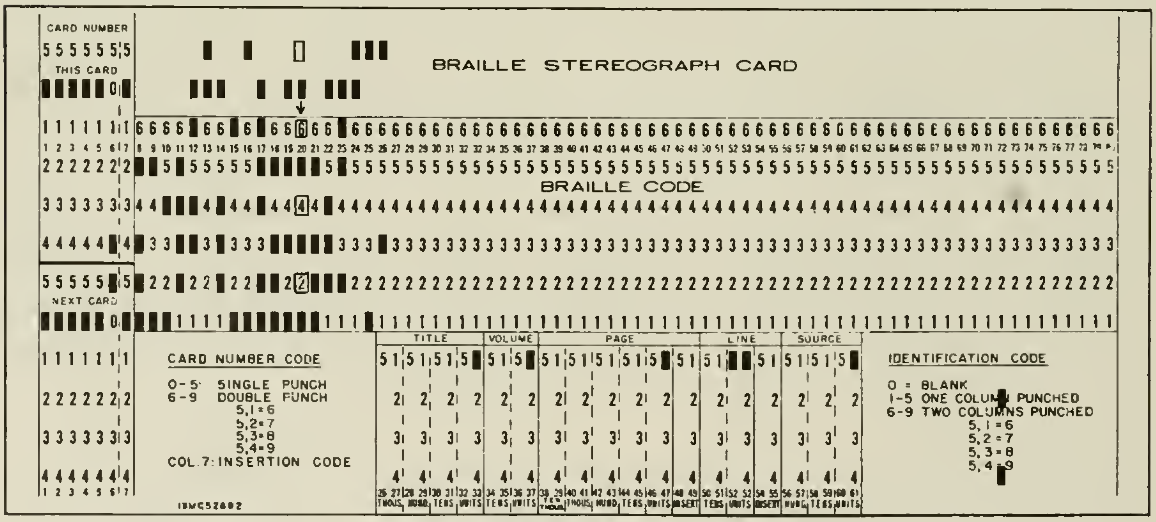
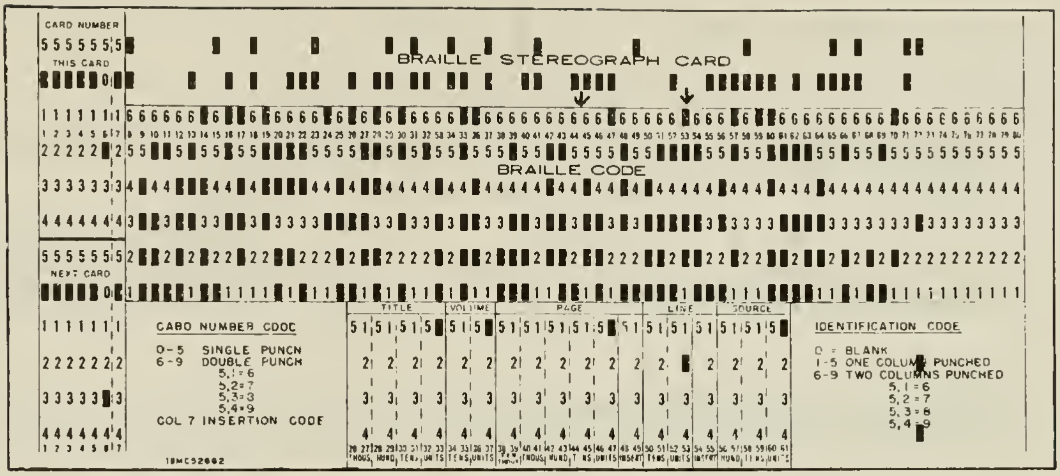


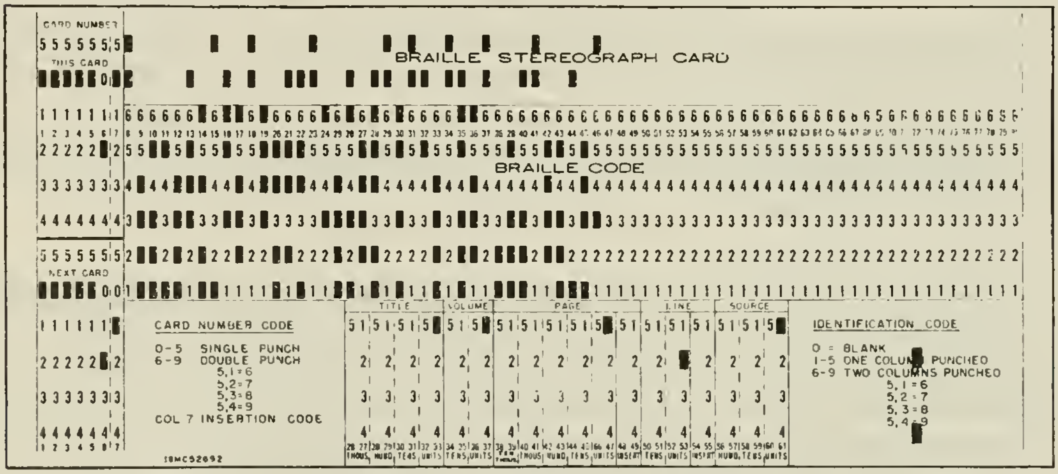
Figure F

The extra 0 in column 20 can be eliminated by punching the delete code.

Error
Card 2



Corrected
Card 2



Corrected
Card 2.1

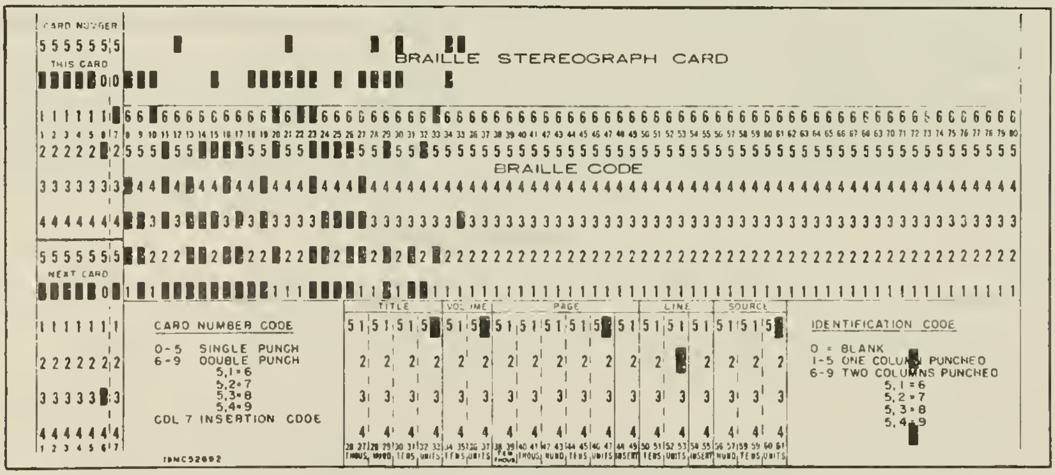
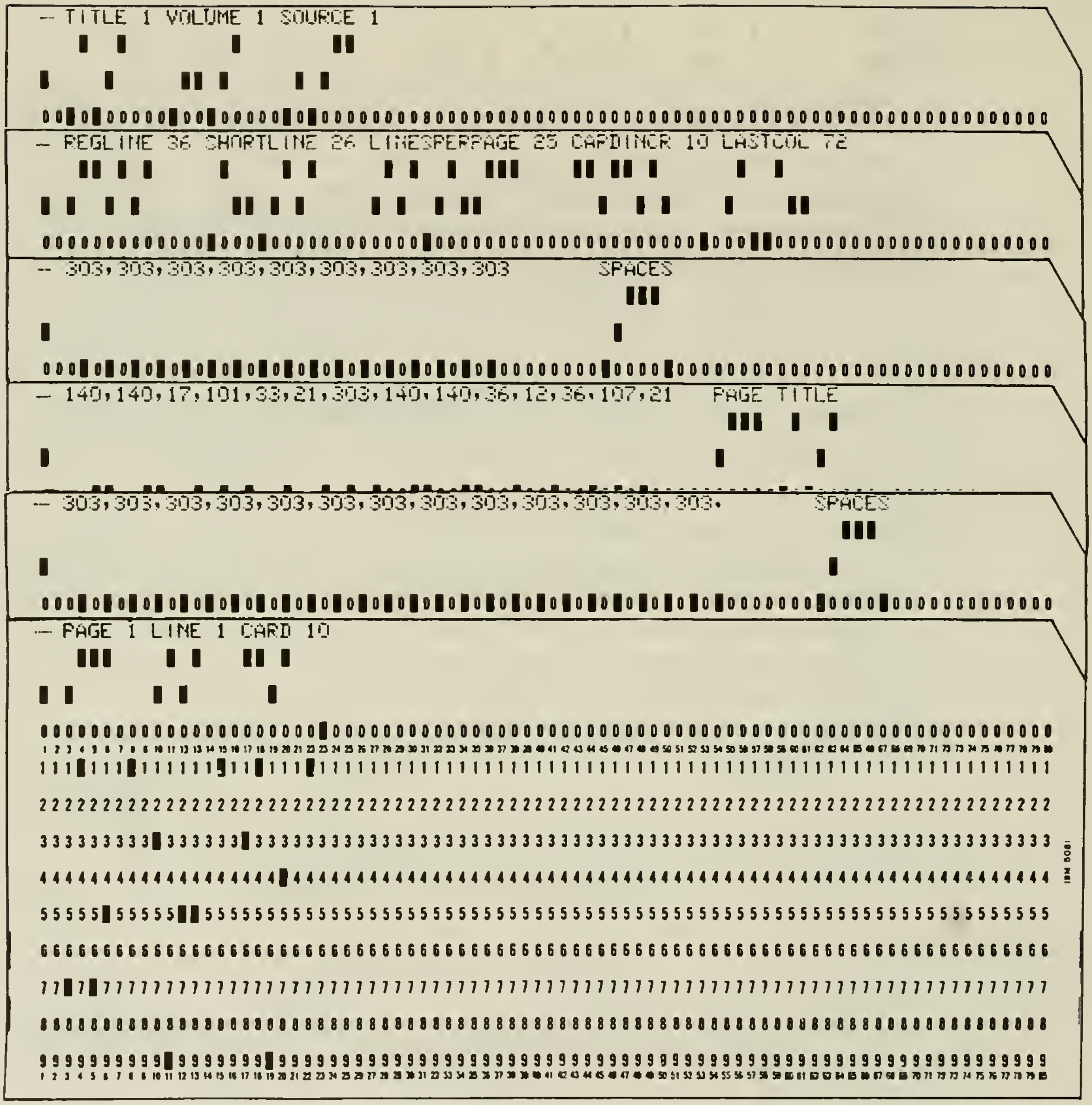


Figure G

The errors in Card 2 are corrected by transferring the information to two cards.



Initialization Cards

Appendix E

OPERATING INSTRUCTIONS

1. BRAILLE TRANSLATION

GENERAL INSTRUCTIONS:

The Braille translation system consists of five phases of operation. Phases 1, 4, and 5 are peripheral machine operations; phases 2 and 3 are 704 runs. Phase 1 will only be executed if input is from tape; phase 5 will only be executed if output is on tape.

PHASE 1: The input data is put on tape using the peripheral card reader. The text to be translated is punched in columns 1-72 of the card. The text cards must be preceded by 6 initialization cards and 2 end cards.

Initialization cards:

Each initialization card must contain an X-punch in column 1.

Card 1:	(Decimal)	Title number, volume number, source number
Card 2:	(Decimal)	Number of characters per regular Braille line, number of characters per short line, number of lines per Braille page, Braille card number increment (*), number of last column in Braille card to be punched.
Card 3:	(Octal)	Braille space codes (303) which should precede page title.
Card 4:	(Octal)	Braille codes for page title
Card 5:	(Octal)	Braille space codes which should follow title.

NOTE: The total number of characters on card 3, 4, and 5 must equal the number of characters per regular line as indicated on card 2. The number of spaces preceding the title may not be less than 3; the number of spaces following the title may not be less than 7.

(*) In order to provide for inserts, Braille cards are normally stepped by 10, starting at 10. If a particular run is to produce insert cards, the increment will be 1.

Card 6: (Decimal) Number of first Braille page, number of first Braille line, number of first output card.(*)

In the sample cards (Fig. G) alphabetic information has been punched to identify the codes for the benefit of the operator. However, it is only necessary to separate these numbers by some non-numeric code.

<u>End cards</u>:

Card 1: $END in column 1-4; X-punch in column 6

Card 2: X-punches in columns 1 and 2

PHASE 2: The BCD tape created in Phase 1 (or the cards) is read into 704 and translated to Braille codes. The Braille output may be written on tape, or punched into cards on line. Tape output requires that the Tape to Card equipment have the column binary device.

If necessary, the translation run can be interrupted under the control of Sense Switch 5. The program can be dumped, onto tape or cards, and the dump output used to restart the translation. The dump may also be initiated when certain error conditions are encountered.

At the end of translation (STOP 0004) the operator must write End-of-File on the two output tapes 4 and 5. They must be rewound manually. These are duplicate tapes and both should be saved until Phase 3 is completed. If Phase 3 indicates that there are no tape errors on Tape 4, then Tape 5 may be discarded.

PHASE 3: The binary output from Phase 2 (either tape or cards) serves as input to the re-translation program. The output from this run is a BCD tape which is used to create a proof listing. The input to this phase (tape or cards) must be saved for use in the final phase.

(*) In order to provide for inserts, Braille cards are normally stepped by 10, starting at 10. If a particular run is to produce insert cards, the increment will be 1.

PHASE 4: The BCD tape created in Phase 3 is printed off-line using either the high- or low-speed printer. This report should be single spaced, eight lines to the inch. The carriage brushes should be raised to eliminate overflow skipping.

PHASE 5: If output from Phase 2 (and input to Phase 3) is on tape, this tape is converted to cards off-line. The Tape-to-Card equipment must have the column binary device. This phase should only be executed after the listing produced in Phase 4 has been proofread and found to be correct.

PERIPHERAL OPERATIONS

Card to Tape: (Phase 1)

1. No more than 2000 cards should be written in one file.
2. The end of the book or section to be translated is marked by two end cards. If more than one book is written on one tape, initialization cards for the second book should follow the two end cards.
3. 80-84 SHARE board.

Tape to Printer: (Phase 4)

1. Single space, eight lines per inch
2. Release carriage brushes to eliminate overflow skipping

Tape to card: (Phase 5)

1. Ready card hopper with Braille cards (Form No. IBMC52692)
2. Standard column binary board

To load program:

From tape:
1. Sense switch 1 Up
2. Ready system tape on unit 2
3. Place BRAL loader in card read (5 cards)
4. READY card reader, then LOAD CARDS

From cards:
1. Sense switch 1 Down
2. Place system deck in card reader: BRAL loader
Program deck
Table deck
Transfer card
3. READY card reader, then LOAD CARDS

Input:

From tape:
1. Sense switch 2 Up
2. Ready BCD input tape on unit 3

From cards:
1. Sense switch 2 Down
2. Ready card reader with initializing cards, followed by text cards

Output:

On tape:
1. Sense switch 3 Up
2. Ready output tapes on units 4 and 5

On cards:
1. Sense switch 4 Up
2. Ready card punch with Braille cards (Form No. IBMC52692)

Note: Both card and tape output can be produced in a single run. To eliminate tape output, set Sense switch 3 Down; to eliminate card output, set Sense switch 4 Down. If both switches 3 and 4 are down, the program will stop at 0101.

CONSOLE OPERATIONS

Set switches 1, 2, 3 and 4 appropriately.

Sense switch 5 is used to interrupt the program. When Sense switch 5 is down, the computer will stop before a new input record is read, and before a new output record is written. At either point, the operator can dump the program (onto tape or cards) in a form suitable for reloading and re-start.

To dump: Set switch 6 Down

SS2 - Up: Row binary card output
Dn: Omit row binary card output

SS3 - Up: Output on Tape 6
Dn: Omit Tape output

SS4 - Up: If Sense switch 3 is also up, column binary card output
Dn: Omit column binary card output

Note: It is possible to get tape output only. However, if column binary cards are to be punched, it is also necessary to produce the tape.

If switches 2, 3 and 4 are all up, column and row binary cards will be interspersed.

If both switches 2 and 3 are down, the program will stop at 0102. The operator can then correct the switch settings, then START.

In addition to these interrupt points, the program can also be dumped at several other stops in the program. Detailed instructions are included in the list of programmed stops.

PROGRAMMED STOPS -- NON-ERROR CONDITIONS

1300: HPR 0001: Program has been loaded. Check to see that switches are set properly; then START

3470: HPR 0002: A. If in Dump status, this stop indicates that Dump has been completed. Reset switches to Run status, then START to continue as before.

B. If restarting, this stop indicates that the program has been loaded for restart. Push START to continue translating.

3417: HPR 0004: End of current translation. If there is more than one book to be translated during a single run, push START to initialize for next run.

Note: Output tapes 4 and 5 have no end of file and are not rewound by the program. This must be done by the operator.

3170: HPR 0011: If Sense switch 5 is Down, the program will stop here before the next input record is read.

To continue, Push START.

To dump, set Sense switch 6 Down, set other switches in appropriate Dump status, then START.

3203: HPR 0012: End of file in card reader. To continue, push START.

To dump, set Sense switch 6 Down, set other switches in appropriate Dump status, then START.

3270: HPR 0014: End of file on input tape. If the file count in the AC indicates that there are more files to be translated, push START. The AC contains the number of files completed.

If more than one input tape is to be translated, mount new tape, RESET, then START. (File count will be zeroed.)

To dump, at end of file or end of tape, set Sense switch 6 Down, set other switches to appropriate Dump status, then START.

3354: HPR 0020: If Sense switch 5 is Down, the program will stop here before the next output record is written.

To continue, push START.

To dump, set Sense switch 6 Down, set other switches in appropriate Dump status, then START.

PROGRAMMED STOPS -- ERROR CONDITIONS

1324: HPR 1001: No X punch in column 1 of initialization card. To ignore error card, push START.

1341: HPR 1002: Illegal BCD character on initialization card. To ignore error card, push START.

1370: HPR 1004: Invalid Braille character in initialization title card. (Character code in AC; character number in XR 2.)

1654: HPR 1010: X punch in column 1 of text card. To ignore error card, push START.

3253: HPR 1020: Ten consecutive errors in reading single tape record. (File count in AC; record count in MQ.)

To try again, push START.

To dump, set Sense switch 6 Down, set other switches to appropriate Dump status, push START. (File and record counts will be zeroed.)

3221: HPR 1040: Invalid end of file.
To ignore error record, push START.

3364: HPR 0101: Sense switches improperly set. Set switches, then START.

3477: HPR 0102: Occurs only when in Dump status. Indicates that sense switches are improperly set. Correct settings, then START.

OPERATING INSTRUCTIONS -- RESTART

The output from the dump (whether on cards or tape) consists of the following:

1. Program
2. Table
3. Transfer instruction

To reload cards:

1. Precede dump output with BRAL loader (5 cards)
2. Set Sense switches for appropriate loading, input, output. (See instructions for normal run.)
3. READY card reader, then LOAD CARDS.

To reload tape:

1. Ready dump tape on unit 2.
2. Set sense switches appropriately for loading, input, output. (See instructions for normal run.)
3. Ready card reader with BRAL loader (5 cards), then LOAD CARDS.

If the program was dumped in the middle of translating an input tape, remount that tape. The program will skip over those tape records which have already been translated and will continue translation at the correct point.

If the program was dumped in the middle of translating cards, only those untranslated cards should be read in when restarting. Initialization cards are <u>not</u> required.

All stops indicated for the normal run can also occur when the program has been restarted. An additional stop can only occur when restarting:

1276: HPR 1100: Invalid end of file, encountered while skipping over tape records.

Either push START or rewind tape, and START

SUMMARY: SWITCH SETTINGS

Run Status:

Loading:	SS1	Up -	Load from Tape 2
		Dn -	Load from row binary cards
Input:	SS2	Up -	Input on Tape 3
		Dn -	Hollerith card input
Output:	SS3	Up -	Output on Tapes 4 and 5. (Each record is Braille card image.)
		Dn -	Omit tape output
	SS4	Up -	Braille card output
		Dn -	Omit card output
Pause:	SS5	Up -	No pause
		Dn -	Pause before input and output

Dump Status: (Sense switch 6 Down)

Output:	SS2	Up -	Row binary card output
		Dn -	Omit card output
	SS3	Up -	Output on Tape 6 (binary card images)
		Dn -	Omit tape output
	SS4	Up -	(Only if SS3 Up) column binary card output
		Dn -	Omit column binary card output

2. RETRANSLATION PHASE

To Load:	From tape:	1.	Ready program tape on unit 1
		2.	LOAD TAPE
	From cards:	1.	Ready card reader with RETRAN deck
		2.	LOAD CARDS
Input:	From tape: SS1 - Up	1.	Mount output tape from Translation Phase on unit 4. (Be sure end of file has been written on that tape.)

Note: This tape must be saved to produce Braille cards in final phase.

From cards: 1. Ready card reader with Braille output cards
SS1 - Dn from Translation Phase.

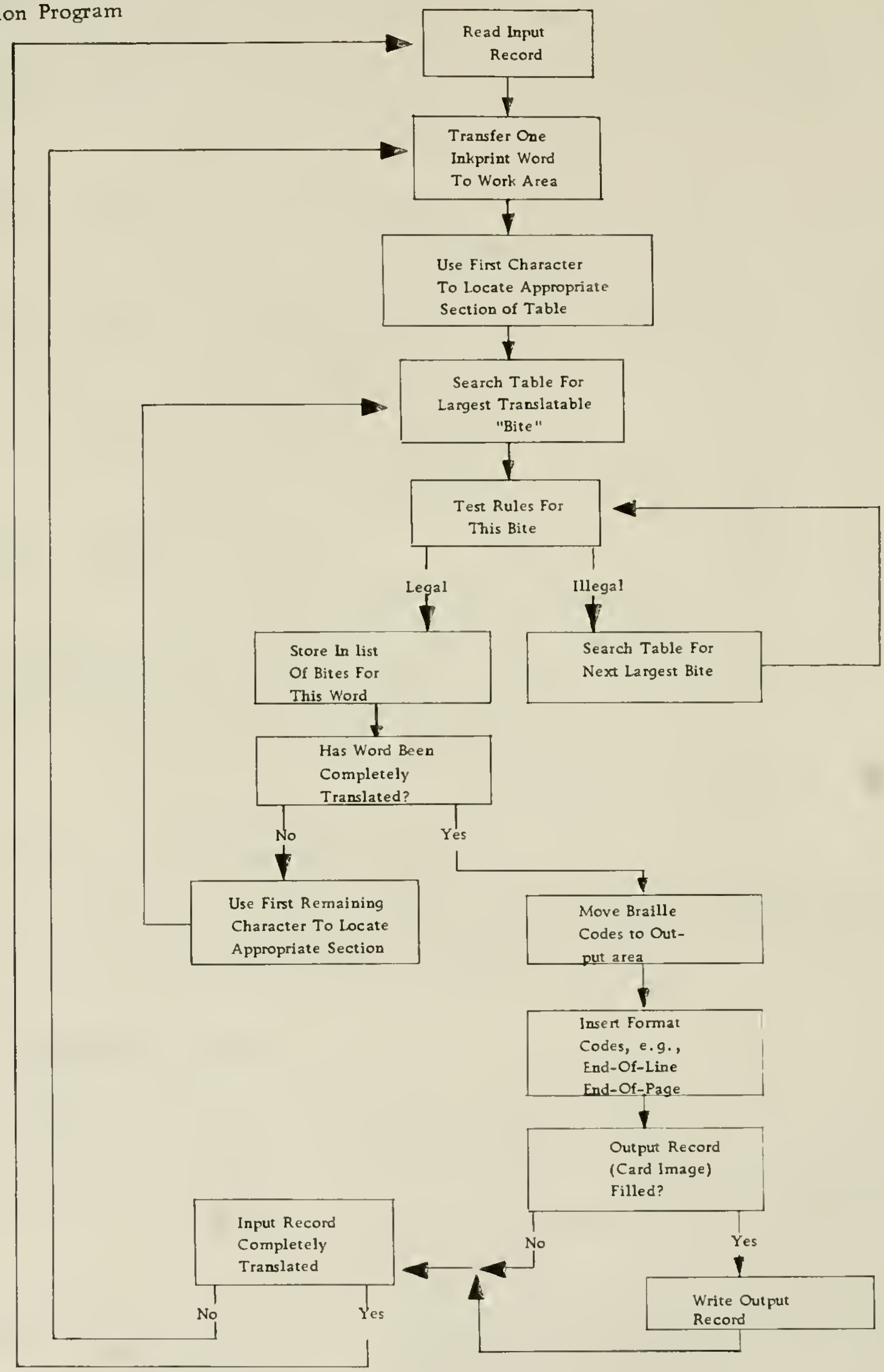
Output: Can only be on tape: Mount output tape on unit 2.

Final Stop: 2411 - HTR 7777

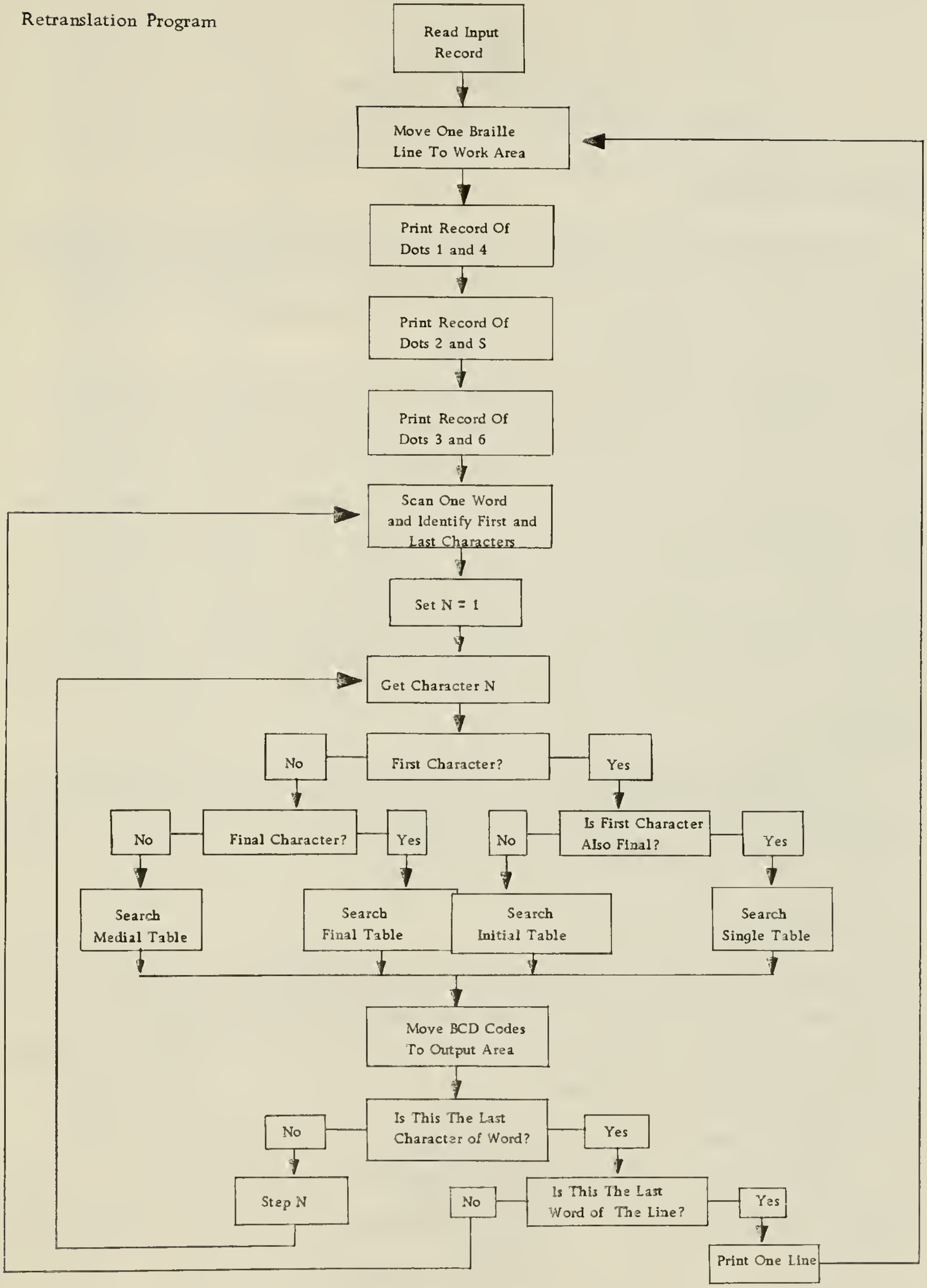
Appendix F

FLOW CHARTS

Translation Program



Retranslation Program



Appendix G
TABLE GENERATOR PROGRAM

GENERAL DESCRIPTION:

Table entries may be added or deleted using this program. During each updating run, the new indices to the Grade II table are computed and stored in the appropriate locations in the Grade I table. Both of these tables are included in the output.

The program writes up to 500 new entries on Tape 5, the current table on Tape 3, and merges these two onto Tape 4. If there are more than 500 entries, the first group is processed, Tape 5 is rewound and the second group of 500 entries is written on Tape 5. These entries are then merged with Tape 4 onto Tape 3. If there is a third group of 500, a new Tape 5 is written and merged with Tape 3 onto Tape 4, and so on. It is necessary to know how many entries have been processed in order to determine whether Tape 3 or 4 contains the updated table. A great deal of time can be saved by using the appropriate tape as input to the next updating run.

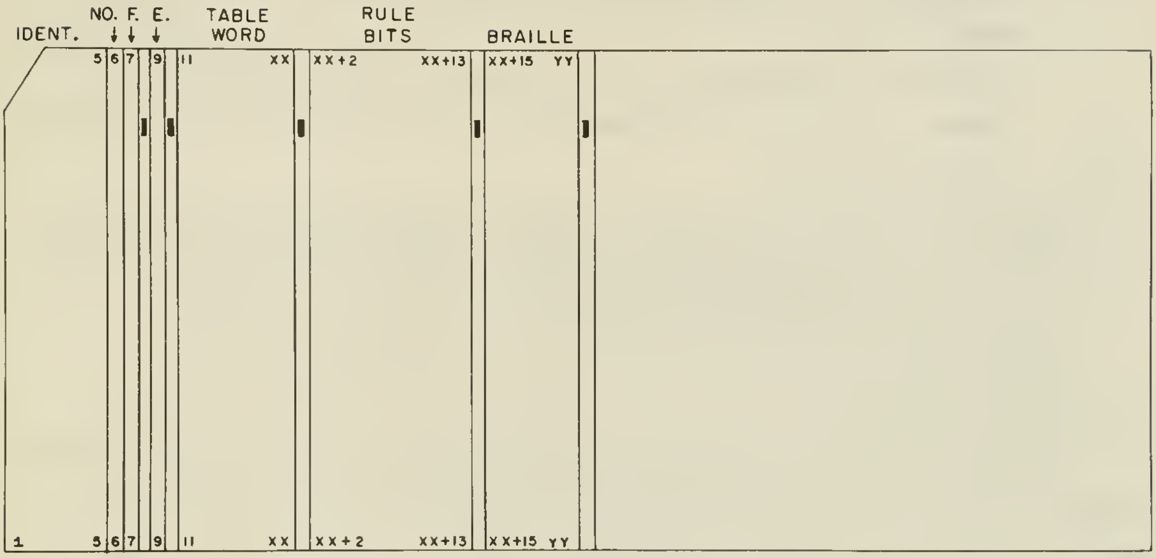
The program always produces column binary card records on Tape 6, and row binary cards on line. It is also possible to punch column binary cards on line under control of Sense Switch 2.

INPUT CARD FORM:

Columns 1-72 of the card may be used. Beginning in column 11 the field lengths are variable. Fields are separated by an 11-punch. Detailed instructions for punching each field are given below, and a sample cards forms ~~are~~ is shown on page 61

Columns	Field Name
1-5:	Identification
6:	Card number
7:	Function number
8:	11-punch
9:	Sub-entry number
10:	11-punch

Columns	Field Name
11→XX: (Variable)	BCD characters for table entry
XX + 1:	11-punch
XX + 2 → XX + 13:	Octal representation of rules word
XX + 14:	11-punch
XX + 15 → YY: (Variable)	Octal representation of Braille
YY + 1:	11-punch



1. Identification:

Any alphabetic or numeric codes may be punched in this field. It is not necessary to fill all five columns, and the code used need not be unique. The entry cards used to produce the present table are punched with the letter combination which necessitated that entry. For example, since the DIS contraction may not be used in the word DISC, that entry is identified DIS. Similarly, since the FF contraction should not be used in the sequence FFOR, that entry is coded FF. (It may be interesting to tabulate the number of entries required by each contraction code.)

2. Card Number:

Most table entries will require only one card, punched 1 in this field. For very long entries which require two or more cards, the identification code should be repeated, and the succeeding cards punched 2, 3, etc.

3. Function number:

1: Add entry to table. If there is an entry with the same BCD characters and sub-entry number, delete that entry and substitute this one. If the sub-entry number is different, this entry will be inserted in the correct position. (See paragraph 4 below.)

2. Delete the table entry with the same BCD codes and sub-entry number. If there is no matching entry in the table, this card will be ignored.

<u>Note</u>: If any other number appears in this column, the program will stop at location 552. If push START, the card will be treated as if coded 2.

4. Sub-entry number:

In some cases it is necessary to have more than one entry with the same BCD codes. The rules or the BCD equivalents will be different, however. For example, the rules for the whole word AND and the part word AND are different. The cards should be numbered in the order in which they should occur in the table. Since sub-entry numbers are not stored in the table, it is not possible to insert a third entry for the same letter combination between two already in the table. To effect this, the two entries should be deleted, and new cards with the proper sub-entry numbers added to the table.

5. BCD characters for table entry:

The BCD characters should be punched beginning in column 11, and an X should be punched following the last significant character of the entry. In the case of space elimination words (to, into, by) and format entries, the last significant character is a space. However, in all other cases, there should be no space before the X-punch.

6. Rules bits:

The 36 bits of the rules word are represented by 12 octal digits. That is, each set of three bits, starting with the Sign position, is represented as follows:

000 - 0	100 - 4
001 - 1	101 - 5
010 - 2	110 - 6
011 - 3	111 - 7

The significance of each bit in the rules word and some examples are shown in Fig. H.

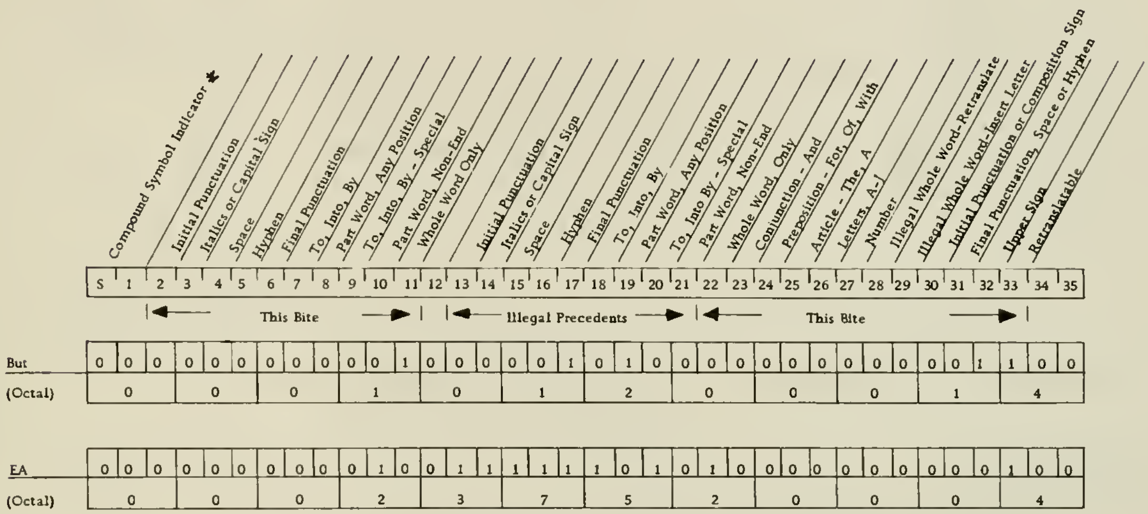


Figure H

*This bit is used to identify Grade I entries which require more than one Braille cell (e.g. asterisk) and should not be used for any Grade II entries.

7. Octal representation of Braille:

The Braille code, plus the validity check bit, is represented in octal form as illustrated below. Note that multiple Braille codes are separated by commas.

Braille Code		Braille Dots V 6 5 4 3 2 1	Octal
BUT	⠃	0 0 0 0 0 1 1	3
AND	⠯	1 1 0 1 1 1 1	157
EA	⠂	1 0 0 0 0 1 0	102
ALLY	⠠⠽	1 1 0 0 0 0 0 1 1 1 1 1 0 1	140, 175

Listed below are the "pseudo" codes used for format entries.

<u>Pseudo Codes</u>:

300: Conditional space #1; may not begin or end a line

303: Unconditional space

305: Conditional space #2; may not occur after line beginning

311: Unconditional page title

312: Conditional page title; may only occur on odd-numbered page

317: Initiate centering

321: Terminate centering

341: Reserve n pages for table, beginning at top of following page

342: Reserve n lines for table, beginning at top of following page

344: Reserve n lines for diagram, beginning at top of next odd-numbered page

350: Conditional end-of-line #1; may not begin a line

353: Conditional end-of-line #2; may not begin first line of page or first line after title

355: Conditional end-of-line #3; may not begin first line of page

356: Unconditional end of line

360: Unconditional end of card

365: Initiate Grade I translation

366: Terminate Grade I translation

372: Conditional number sign; may only occur at beginning of line

374: Hyphen; may end line

377: No operation

TABLE GENERATOR PROGRAM

Programmed Stops:

<u>Non-Error Conditions</u>

0125:	HPR 1	Program has been loaded. Check Sense Switches, Tapes, etc., then START.
0552: 0561:	HPR 3 HPR 4	Incorrect function number. If push START, will be treated as a delete card.

<u>Error Conditions</u>

0272:	HTR 214:	Illegal BCD code; push START to ignore
0514:	HTR 474	Two entry cards have the same entry number and alphabetic information. If push START, first card will be ignored.
0667:	HTR 621	Normal stop if SS 2 UP
0677:	HTR 621	Normal stop if SS 2 DN. Additional entries can be processed at this point. Ready card reader, th then START.
0674:	HTR 703	Only occurs if SS 2 DN. Output tape record too long. Push START to ignore extra words.
0712:	HTR 670	Only occurs if SS 2 DN. Checksum error while reading tape record to produce column binary card. Push START to backspace and reread.
1037:	HPR 5	Checksum error while reading tape record while merging. Push START to backspace and reread.
1060:	HPR 6	Current table out of order. Dump appropriate tape (3 or 4) to investigate.
1111:	HTR 1110	New entry records on Tape 5 out of order. Dump Tape 5 to investigate.
1221:	1215:	Only occurs when SS 2 DN. Checksum error in last card punched. Push START to re-do card.
1231:	HTR 1234	Tape write error. Push START to backspace and rewrite.

A PORTION OF THE DICTIONARY TABLE

Identification
Card number
Function number
Sub-entry number
Table entry
Rules bits
Braille codes

```
ING  11-1-INGLASS-001001020014-24,33,107,101,116,116-
ING  11-1-INGRAD-001001020014-24,33,27,101,131-
ING  11-1-INGU-001001020014-24,33,145-
ING  11-1-ING-001037520014-154-
INTO 11-1-INTO -002001760004-24,126-
IN   11-1-IN-000123760004-24-
IN   11-2-IN-001001020004-24-
XF   11-1-ITSELF-001001020014-55,113-
IT   11-1-ITS-000101260014-55,116-
ITY  11-1-ITY-001037520214-60,175-
IT   11-1-IT-000101260014-55-
JUST 11-1-JUST-000101260014-132-
OW   11-1-KILO-001001020010-5,12,107,125-
KNLDG11-1-KNOWLEDGE-000101260014-5-
KNOW 11-1-KNOW-001001020014-120,5-
LESS 11-1-LESS-001037520214-50,116-
LR   11-1-LETTER-001001020014-107,27-
LIKE 11-1-LIKE-000101260014-107-
LL   11-1-LITTLE-001001020014-107,107-
LORD 11-1-LORD-001001020014-120,107-
MANY 11-1-MANY-001001020014-170,115-
MENT 11-1-MENTION-001001020014-115,42,60,35-
MENT 11-1-MENT-001037520214-60,36-
SH   11-1-MISH-001001260010-115,12,116,123-
ST   11-1-MISTA-001001260010-115,12,116,36,101-
ST   11-1-MISTRESS-001001020010-115,12,14,27,21,116,116-
ST   11-1-MISTR-001001260010-115,12,116,36,27-
ONE  11-1-MONETARY-001001020014-115,120,125,36,101,27,175-
MORE 11-1-MORE-000101260014-115-
MOTHR11-1-MOTHER-001001020014-120,115-
MCH  11-1-MUCH-001001020014-115,41-
MUST 11-1-MUSTN$,T-000101260014-115,14,35,104,36-
MUST 11-1-MUST-000101260014-115,14-
MYF  11-1-MYSELF-001001020014-115,175,113-
NAME 11-1-NAMENT-001000020014-35,101,60,36-
NAME 11-1-NAME-001001020014-120,35-
NEC  11-1-NECESSARY-001001020014-35,21,11-
NEI  11-1-NEITHER-001001020014-35,21,12-
NESS 11-1-NESS-001037520214-60,116-
NOT  11-1-NOT-000101260014-35-
OW   11-1-NOWAY-001001020010-35,125,72,101,175-
OW   11-1-NOWI-001001020010-35,125,72,12-
OW   11-1-NOWH-001001020014-35,125,161-
ED   11-1-OED-000101260014-125,21,131-
EN   11-1-OEN-001001020014-125,21,35-
OF1  11-1-OF -020003264034-167,300-
OF2  11-1-OF-001001020014-167-
ITY  11-1-OITY-001037520214-125,12,36,175-
ONE  11-1-ONEA-001001020014-125,35,102-
ONE  11-1-ONED-001001020014-125,35,53-
ONE  11-1-ONEER-001001020014-125,35,21,173-
ONE  11-1-ONEE-001001020010-125,35,21,21-
ONE  11-1-ONEI-001001020010-125,35,21,12-
ONE  11-1-ONEN-001001020014-125,35,42-
ONE  11-1-ONER-001001020014-125,35,173-
ONEF 11-1-ONESELF-001001020014-120,125,113-
ONE  11-1-ONESE-001037520010-125,35,21,116,21-
ONE  11-1-ONESS-001001020010-125,60,116-
ONE  11-1-ONET-001037520014-125,35,21,36-
ONE  11-1-ONEUM-001001020010-125,35,21,145,116-
ONE  11-1-ONE-001001020014-120,125-
ONG  11-1-ONG-001037520214-60,33-
TIME 11-1-OPTI-001001020010-125,17,36,12-
OUGHT11-1-OUGHT-001001020014-120,63-
OUND 11-1-OUND-001037520214-50,131-
OUNT 11-1-OUNT-001037520214-50,36-
OURVS11-1-OURSELVES-001001020014-63,27,47,116-
OUT  11-1-OUT-000101260014-63-
OU   11-1-OU-001001020214-63-
OW   11-1-OW-001001020014-152-
OC   11-1-O$,CLOCK-001001020014-125,104,11-
PD   11-1-PAID-001001020014-17,131-
```