

## **Braille Transcription and Mechanical Translation**

John P. Cleave, Birkbeck College, University of London, London, England

TRANSCRIBING romanized print into Braille suitable for reading by the blind is a problem which has similarities to those arising in mechanical translation. The theoretical problem of mechanical translation is to construct an operational syntax - a set of formal rules of translation prescribing operations to be performed on the text to get the output text - entirely in terms of patterns of input words and types of words and such information as may be contained in the dictionary. And this is simplified already, firstly by the small vocabulary (consisting of a definite number of letters, capitalized letters, punctuation marks, etc.) and the absence of ambiguity and, above all, the existence of explicit rules for transcription which are already partly formalized.

### The Braille Systems

Braille is a system of embossed characters formed by six dots arranged and numbered as in Fig.1(a). In the project outlined here the output of the computer presents the Braille characters as a series of six "1's" or "0's" corresponding to the six Braille dots. Thus the Braille character of Fig.1(b) is represented by the binary number of 1(c).

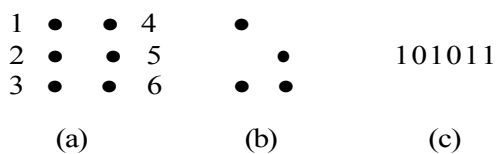


Figure 1

While to each letter-press character there corresponds one Braille sign, there are Braille characters (single-cell contractions) and pairs of Braille characters (double-cell contractions) which under various conditions represent groups of inkprint letters. Thus, the Braille character of Fig.2 represents the group "wh" in that order.

The rules of Braille largely concern the conditions under which contractions can be made.

There are four grades of Braille: Grade I, uncontracted; Grade "one-and-a-half"; Grade II, moderately contracted; Grade III, highly contracted. The latter grade is rarely used. Grade I presents no problem to the computer. Grades "one-and-a-half" and II are the more profitable lines of inquiry,



Figure 2

The problem to be dealt with is that of constructing a program by which an electronic computer will do the work of making the contractions correctly. We envisage an input organ to the electronic computer with a keyboard with keys for all the characters used in inkprint (including punctuation marks). The output from this organ is in the form of binary numbers (machine characters) on which the computer operates and finally obtains from each such number a six digit binary number representing the six Braille dots. (Fig.1) An output mechanism, similar to an ordinary teleprinter (it could in fact be such a piece of equipment fitted with a mechanical device), will convert this number into the Braille characters as actually used.

The Braille signs used in this project are as shown in Fig.3. These characters are divided into classes called "lines." Line 1 is formed by dots 1-2-4-5. Line 2 is formed by adding dot 3 to each of the characters of line 1, and line 3 by the addition of dots 3 and 6 to line 4. Line 4 is formed by the addition of dot 6 to line 1 signs. Line 5 is obtained by repeating line 1 in a lower position. This classification has no significance as far as the Braille rules are concerned.

A further classification of Braille signs, which cuts across the "line" division, is the classification into "lower signs" and "non-lower signs"; a lower sign is a Braille sign which does not

contain dot 1 or dot 4. The lower signs are all those of line 5 together with "com" of line 6. This again is a formal property of the Braille

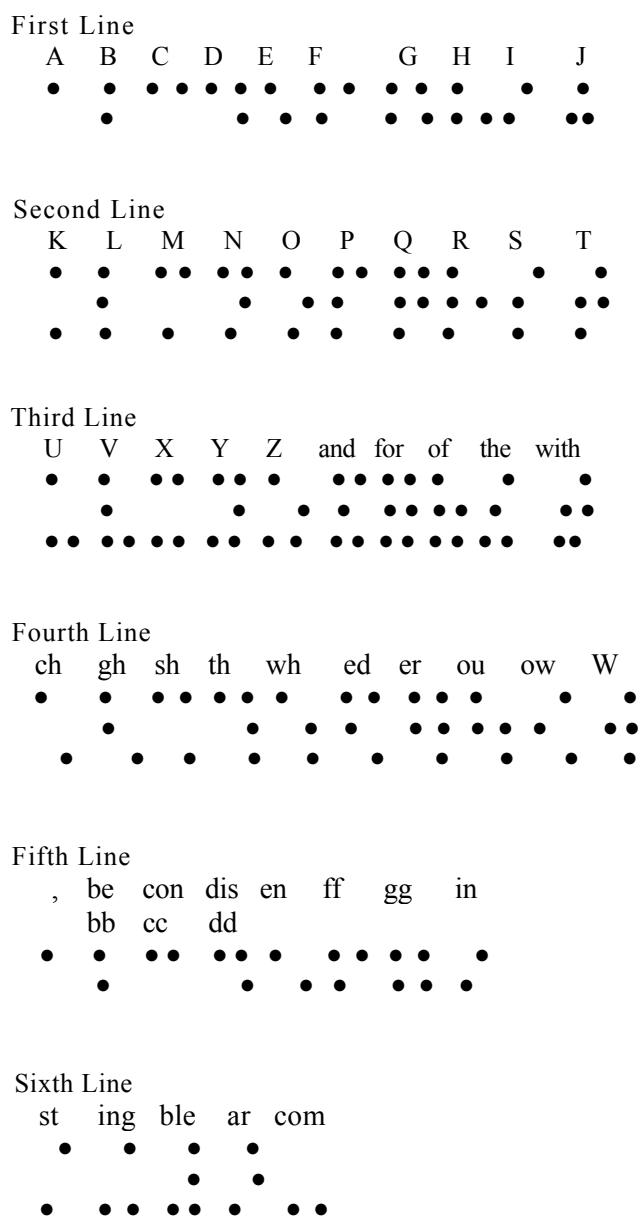


Figure 3

sign, but for technical convenience it is explicitly represented by a code digit attached to the coded Braille. The rule concerning the contraction of double letters requires explicit mention of the lower sign property.

Formalization of the Rules

The rules followed in this work are those printed in Standard English Braille.<sup>1</sup> The rules as expressed in the booklet are not all usable for a mechanical transcription of inkprint characters into Braille as they stand, though they are perfectly satisfactory for a human agent. To be put in a form suitable for the construction of a machine program the rules must be formalized. That is, all reference to terms which cannot be given an extensional definition in terms of the machine characters, or a definition in terms of their formal properties, must be eliminated. For instance, rule 34 reads:

Contractions forming parts of words should not be used when they are likely to lead to obscurity in recognition or pronunciation and therefore they should not overlap well-defined syllable divisions. Word signs should be used sparingly in the middle of words unless they form distinct syllables. Special care should be taken to avoid undue contraction of words of relatively infrequent occurrence.

The principal term in this rule is "syllable." It would be possible to formalize this term if a complete list of syllables could be compiled. This would be a clumsy procedure and would require comparison of incoming words with a large dictionary for recognition of syllables. Similar difficulties arise with "pronunciation," though the problem is largely solved when the "syllable" question has been resolved. The most simple way to resolve the issue is to ignore the restrictions imposed by this rule. Another, which includes a non-formal restriction, is rule 21:

The word signs and, for, of, the, with, a, may follow one another without a space where the sense permits. . .

The condition "where the sense permits" is impossible to formalize fully except by constructing a list of phrases in which the elimination of the space between these "and-words" may be effected without destroying the sense. However, the sense may not be determined by the phrase but by the whole sentence. The task of including this condition in its entirety in a machine program is now immense. Confusion could arise when a space is eliminated between and-words where at least one is part of a word.

1 Published by the "National Institute for the Blind," London, 1932.

The restriction could then be formalized to read:

" . . unless at least one of the and-words is part of a word"

It is simpler to ignore the wide restriction and to base the space-elimination entirely upon the occurrence of the words. More will be said of this rule later.

On the other hand some of the rules are already adequately formalized. For instance, Rule 27:

The contractions bb, cc, dd, ff, gg, may only be used when they occur between letters and signs of the same line of Braille.

Since "word" and "line" can be given formal definitions the rule as it stands is sufficient though it is more explicit (ignoring the complication caused by "line") if we simply say:

Use the contractions bb, cc, dd, ff, gg if the sign preceding and the sign following b b, c c, d d, f f, g g are neither spaces nor punctuation marks.

An important principle in formalizing the rules is the explicit representation in the machine characters of the properties used for the operation of the program. For instance, a word can be defined formally as the series of signs lying between signs each of which is either a space or punctuation mark. We therefore require that the computer recognize the punctuation marks. It would obviously be possible to define the punctuation marks extensionally as "either the comma or full stop or exclamation mark or.." The process by which the machine recognizes the punctuation mark is then quite complicated, involving comparison of the incoming letter with each punctuation mark in turn, which is slow and wasteful of storage space. The simplest procedure is to indicate membership of this group of words by a digit of the machine character. Several other properties, either of the Braille characters or the letter-press characters, and membership of various other classes are best represented by digits of the machine characters.

#### The Structure of the Machine Characters

The machine characters must bear the six digits representing the Braille dots. It is technically convenient to represent the membership of the various classes of sign by a set of three digits (the code-digits) preceding the six Braille digits, so that the machine character is a number with nine binary digits. Thus the machine character has the following structure:

1st position punctuation digit  
2nd position "and"word digit  
3rd position "lower sign" digit

These are the code digits. The 4th – 9th positions represent the Braille dots: these digits are the machine representation of Braille.

The first digit, showing whether the letter is a punctuation mark, presents explicitly a property of the alphabetic letter rather than of the structure of the corresponding Braille sign, for a Braille sign may be used either as a contraction or as a punctuation mark (see the signs of line 5). Since some of the Braille rules concern the occurrence of punctuation marks, it is necessary that the machine characters corresponding to such signs carry that information explicitly. Thus the machine can determine the presence of a punctuation mark in the accumulator by shifting left one place and then using the conditional transfer order to discriminate on the sign digit.

#### Pattern Sensing

A method of detecting patterns of signs is to delay the final printing while sending the last several characters in turn through a series of memory locations. The context of any machine character can then be searched. An illustration of this process is provided by the following method of operating Rule 21 mentioned above. The series of machine characters, after having been modified by the contraction program to produce the and-word characters, is sent serially through five memory locations. If the conditions for space elimination are not present, the character in the fifth position is sent to the "print routine" which removes the code digits and prints the six digits representing the Braille sign. The characters in the remaining positions are then shifted one place by the "shift routine" leaving the first place to be occupied by a new character from the contraction routine. Rule 21 in the form required by the machine program now reads:

- (i) if there are either punctuation marks or spaces in locations (1) and (5) go to (ii); if not go to the print routine.
- (ii) if there is a space in (3) go to instruction (3); if not go to the print routine.
- (iii) if there are and-words in both positions (2) and (4) shift the character in (2) to (3) and that in (1) to (2) (space-elimination); if not go to the print routine.

This version of the rule is in fact weaker than the original since it permits only pair-wise juxtaposition of "and"words. But it does deal ade-

quately with the majority of cases. It would be possible to construct a routine for effecting the space-elimination in all the circumstances demanded by the formalized version:

"the 'and' words may follow one another without a space unless at least one of them is part of a word"

This, however, would be rather long and would not be justified by the frequency with which three or more consecutive and-words occur, compared with the relatively large frequency of pairs of and-words.

More complicated procedures of a similar nature are necessary to operate the rules concerning numerical expressions, ellipsis, compound lower signs and capital letters.

### The Dictionary

In Grade 'one-and-a-half' it is unnecessary to have a dictionary for the contractions; incoming letters may be compared on arrival with possible members of contractions by means of a "contraction routine." Thus, if an "a" is detected, the contraction routine compares the following character with "r". If an "r" is found, the "ar" contraction is subjected to the next part of the program; if not, "a" is sent to the next part of the program after which the letter following "a" is examined to determine whether it could be the initial letter of a group which could be contracted.

Grade II Braille, on the contrary, contains so many contractions that it is necessary to use a "dictionary" of groups which can be contracted. Characters must then be fed in serially and stored in a set of temporary locations - the Initial Word Store - until a whole word has been received. The dictionary matching mechanism then takes the first letter in the Initial Word Store and finds the longest dictionary entry which is part of that word. The appropriate contraction is selected and sent to another set of storage locations - the Final Word Store - after which

the remainder of the word is treated in the same way. Should no entry be found, the first letter is sent to the Final Word Store and the matching procedure started with the second letter.

There may be several ways of contracting a word. The choice between the methods of contraction is governed by considerations of length. That way must be chosen which gives the shortest transcription. The case where two different methods of contraction yield words of equal length is governed by rule 35:

In cases where a word may according to the above rules be contracted in two or more ways, each saving the same amount of space, that way should be selected which produces the most readable combination of dots. If the same space is saved, simple contractions are better than two-celled word-signs.

Avoid using Double Letter Signs where there is an alternative single cell contraction.

The dictionary is so constructed that the shortest set of contractions is automatically chosen. For instance, "themselves" precedes "the" in the dictionary so that if "themselves" occurs in the Initial Word Store it is compared with the appropriate entry before being compared with "the". If, however, "them" occurs in the text, the longest dictionary entry occurring which is part of that word is "the". The priority rule for single-cell contractions is solved by including in the dictionary those phrases which provide a double-"translation." For instance, the phrase "oner" occurs in the dictionary and precedes "one". "Oner" may be contracted in two ways - "one r" and "o n er". In the first case "one" is a two-cell contraction so that "one r" occupies three cells. In the second case the translation occupies three cells since "er" is a single-cell contraction. By rule 35 "o n er" is the correct translation of "oner" so the dictionary includes o n er as the dictionary entry. Thus, Rule 35 does not appear explicitly in the machine program but is implicit in the construction of the whole program and, in particular, of the dictionary.