[From: *Information theory*: papers read at a symposium on 'Information theory' held at the Royal Institution, London, September 12th to 16th, 1955.; edited by Colin Cherry (London: Butterworths, 1956).]

20

A PROGRAMME FOR BRAILLE TRANSCRIPTION*

J. P. CLEAVE
Birkbeck College, London

THE TRANSCRIPTION of ordinary alphabetic text into Braille is essentially a coding problem. As will be seen later, there is not a simple one-one correspondence between a printed character and the corresponding Braille sign: the transcribed character employed depends upon context. The problem to be dealt with is the use of an electronic computing machine to do the work of determining the appropriate transcription-work that is now done by a human agent with the aid of a mechanical or electro-mechanical Braille machine.

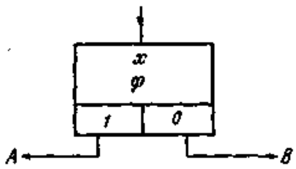


Figure 1. The machine instructions.

In the schematic diagrams below, we use the notation of *Figure 1* to indicate the part of a programme which recognizes a particular character, or one of a certain type. This symbolizes the complex instruction 'if the machine character is of the type characterized by Φ go to the programme A, if not go to B'.

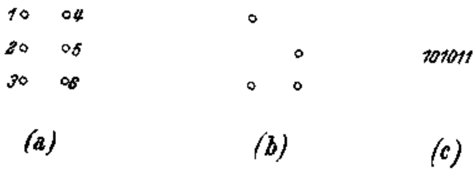


Figure 2.

Braille is a system of embossed characters formed by six dots arranged and numbered as in *Figure 2a.* 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 *Figure 2b* is represented by the binary number of *2c.*

Production of a Braille text as actually used is now simply a problem of fixing a mechanical device to the output organ of the computer.

However, the computer actually operates on a modified version of the Braille signs—the 'machine characters' which consist of a group of digits, the

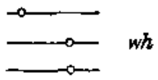
* This is a shortened version of the original paper—ED.

'code digits' and the six Braille digits following. The latter are finally obtained in the accumulator or register by a left shift operation on the machine characters.

Braille systems differ in their complexity, that is, in the extent to which 'contractions' are used. A contraction is a Braille character which under certain conditions is a transcription of a series of alphabetic characters. Thus, the Braille character of *Figure 3* represents the group 'wh' in that order.

The rules of Braille largely concern the conditions under which the contractions can be made. All the systems are based upon Grade I Braille which is uncontracted. Grade 'one and a half' has only simple contractions, whilst Grade II is moderately contracted. Grade III is highly contracted and rarely used. The system programmed in this paper is a simplified version of Grade one and a half. The simplification is seen in the omission of rules concerning capitals and the restrictions on contractions at the end of a line of Braille. The representation of numbers has not been considered. Owing to the limited facilities of the standard teleprinter input device, some of the rules concerning punctuation marks have been left out. In spite of the simplifications, the problem of interpreting Braille rules of the higher grades in terms of a machine programme is no more complex than the limited problem considered here.

Figure 3.



Braille signs are classified into 'lower signs' and 'non-lower signs'; a lower sign is a Braille sign which does not contain dot 1 or dot 4. This again is a formal property of the Braille sign, but for technical convenience 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.

The number attached to the machine representation of Braille consists of three digits, so that the machine character is a number with nine binary digits. Thus the machine character has the following structure:

first position, punctuation digit
second position, 'and'-word digit
third position, 'lower sign' digit

These are the code digits. The fourth-ninth 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. 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.

There is a Braille rule concerning the 'and'-words which will be explained below. The 'and'-words are the signs for 'a', 'and', 'for', 'of', 'the', 'with'. It is not logically necessary that the 'and'-words be specially marked machine characters: an extensional definition of this class of signs could be given so that the schema of *Figure 4* could be used to detect the presence of an 'and'-word.

A much simpler process uses a certain code digit to signify membership of the 'and'-word class. Detection of a member of this class requires merely the use of a left-shift operation and a conditional transfer order. This process is clearly more economical on storage space, and reduces the probability of error.

The rules quoted below are taken from 'Standard English Braille' published by the National Institute for the Blind. Only those parts relevant to the simplified version of Grade 'one and a half' are stated here.

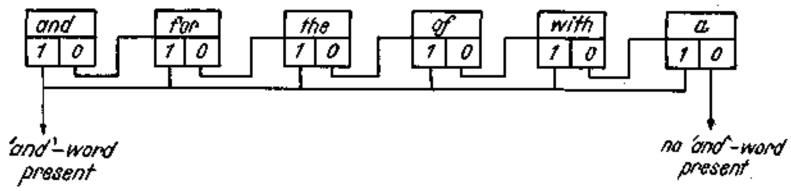


Figure 4.

Rule 21. The word-signs *and, for, of, the, with, a,* may follow one another without a space between them, where the sense permits. These word signs should be used as parts of words wherever possible (see Rule 34) in preference to any other contraction, unless their use entails waste of space.

Rule 22. The contractions . . . *of and, for, with, be* and *in* . . . may be used for the whole words for which they stand ....

Rule 24. The contraction *ea,* dot 2, may be used only when these letters occur between two letters (or contractions) of the same word in one line—it may never begin or end a word. It should always be used in preference to *ar* in such words as 'hear', 'dearth', *etc.* It should not be employed where the letters *e* and *a* belong to separate and well-defined syllables as in 'react', 'readdress', 'preamble' but it would be permissible in 'realize' (see Rule 34).

Rule 25. The contractions for *be, con, dis* (except when *be* stands alone), may be used only as syllables either at the beginning of a word, or at the beginning of a line in a divided word.

Rule 26. The contraction for *com* may be used only at the beginning of a word or of a line, and may not be used when it would be in contact with the hyphen or dash. It need not be a syllable.

Rule 27. The contractions for *bb, cc, dd, ff, gg,* . . . may only be used when they occur between letters or signs of the same word and in the same line of Braille.

Rule 28. Any number of lower signs may follow each other if they stand for separate words written with a space between . . .

Rule 29. One lower sign may not follow another without a space, unless one of them is in contact with a sign containing dot 1 or dot 4.

Rule 30. Not more than two lower signs may join each other.

Rule 31. Contractions for *ing* and *ble* must not begin a word but may begin a line in the case of divided words.

Rule 34. 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.

Rule 35. . . . Avoid using double letter signs where there is an alternative single cell contraction *e.g. m-ed-dle* not *me-dd-le.*

These rules require further explanation and, in some cases, modification to some extent before they can be framed in terms of the machine instructions.

Rule 21. This rule contains the phrase 'where the sense permits'. 'Sense' cannot be formalized for the machine unless a complete list of phrases can be given in which the proposed elimination of the space between 'and'-words would lead to confusion of the sense: this is equivalent to an extensional definition of sense. Likewise, the provisions of Rule 34 present difficulties of formal presentation. We see that the rule is framed in terms of 'obscurity of recognition' or 'pronunciation', and in terms of syllable divisions. It would be possible to operate these provisions by constructing a dictionary of frequently occurring words in which certain contractions or operation of certain rules were not permissible on the basis of syllable division or obscurity of recognition. However, such a procedure enormously complicates the problem and the programme requires a much greater storage space. The solution adopted has been to make the contractions of Grade one and a half, regardless of whether they cut across syllable divisions. Thus, the 'and'-word contractions are made indiscriminately, so making the space elimination of Rule 21 independent of the 'and' contractions. The operation of Rule 21 can thus be performed by a part of the programme following the contraction programme, which now includes the 'and'-contracting orders.

Printing of Braille characters is delayed by five signs by sending them through five storage positions—the 'double-letter filter'. These five positions enable the machine to detect certain sequences of types of signs for the operation of Rule 21. In the project outlined here, a space is eliminated between two 'and'-words when the pattern of signs of *Figure 5a* is present in the double-letter filter. The programme for the operation of the rule is represented schematically in *Figure 5b.* It leads, however, to a weakened version of the rule: the space-elimination leads to the pattern of characters of *Figure 5c.* If this is now followed by another free 'and'-word *(i.e.* one not part of a word) as in the phrase:

. . . and with . . .

the pattern of sign types in the filter positions prior to the operation of Rule 21 is as in *Figure 5d* and no space-elimination can take place here owing to the 'and'-word in 5 (see *Figure 5b).* Thus, by this method we achieve a pair-wise space-elimination: a series of alphabetic characters

. . . space	and-word	space space	and-word	space	and- ... word

has the Braille representation

. . . space	and-word	and-word	space	and-word	and- . . . word

which is adequate for most cases.

The complication arising from the possible occurrence of juxtaposed 'and'-words as bound forms *(i.e.* as parts of words) is not disposed of by using an extra filter position, but removed one stage further: confusion would result if three bound 'and'-words occurred at the beginning of a word.

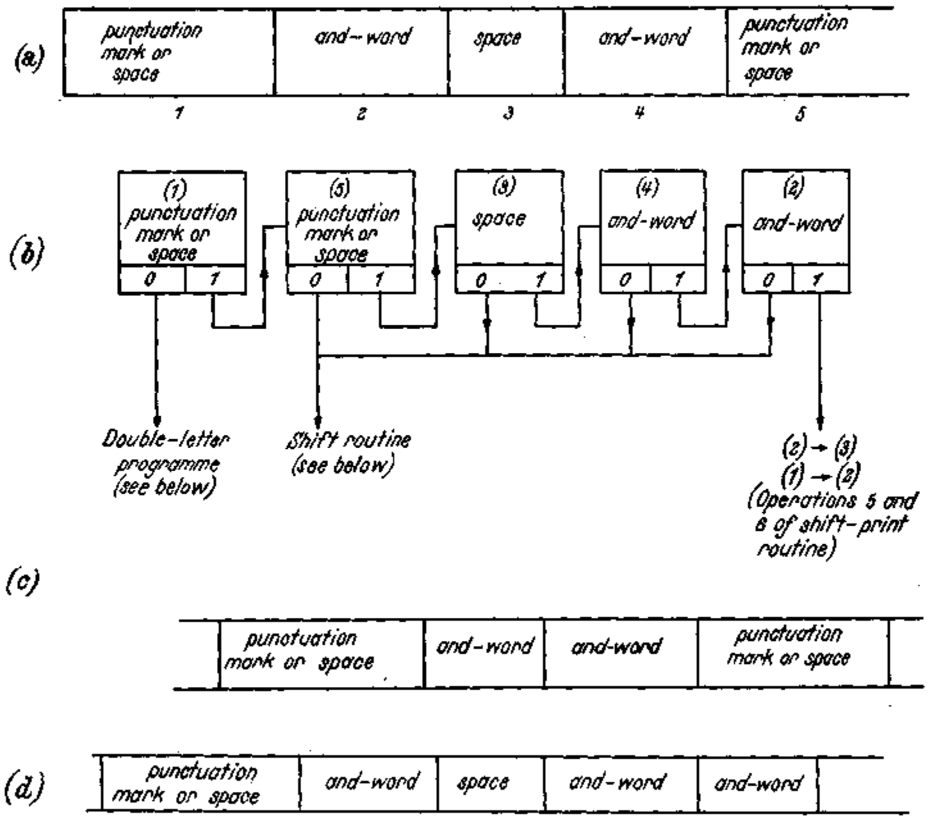


Figure 5.

Rule 24. Here again we have the difficulty of framing an extensional definition of 'syllable' (see remarks on Rule 21). In this project we have ignored the *ea* contraction. It could easily be incorporated in the programme, in which case (ignoring the difficulty of syllable definition) the priority rule would have to be programmed. This could simply be done by ensuring that wherever *e* is followed by *a* the contraction *ea* is formed. The following 'r' in the sequence of characters 'ear' could not possibly be incorporated in an *ar* contraction.

Rule 26. Since printing is always delayed five signs by the double-letter filter, position (2) always contains the sign immediately preceding that about to be sent to the filter. The programme determines whether an incoming character is the beginning of a word by examining position (2). Thus, if *c* is an incoming letter and (2) contains a punctuation mark or a space and 'o' and 'm' follow, then the *com* contraction can be made.

Rule 27 is complicated by the double-letter contractions being lower signs, so this rule will be further considered when the rules governing the lower signs have been examined.

Rules 28, 29, 30 govern the occurrence of lower signs. The interpretation given to these rules in our programme is that a lower sign contraction cannot be made if it would be preceded by two lower signs, that is, if positions (2) and (3) of the double-letter filter contain lower signs. This is represented on the programme schema of *Figure 6a.* The rules will be violated if a lower sign contraction is made which is preceded and followed by a punctuation mark. This, in practice, would only occur with the contractions *be, in,* standing as free forms. The circumstances in which the words 'be', or 'in' have punctuation marks both preceding and following are so rare as to be negligible. That any number of punctuation signs may follow one another is fulfilled automatically, since, not being recognized by the contraction programme, they go immediately to the double-letter filter.

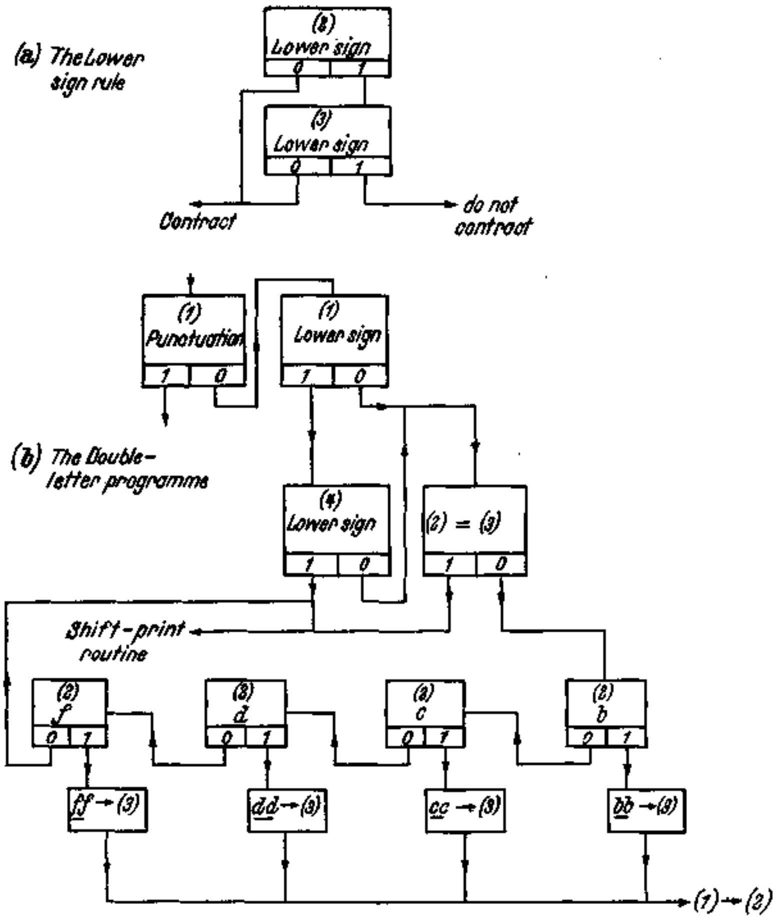


Figure 6.

In the light of these rules, double-letter contractions are made only if:

1. position (1) of the double-letter filter is not a punctuation mark.
2. either position (1) of the double-letter filter contains a lower sign or (4) contains a lower sign.

This is represented schematically by *Figure 6b.*

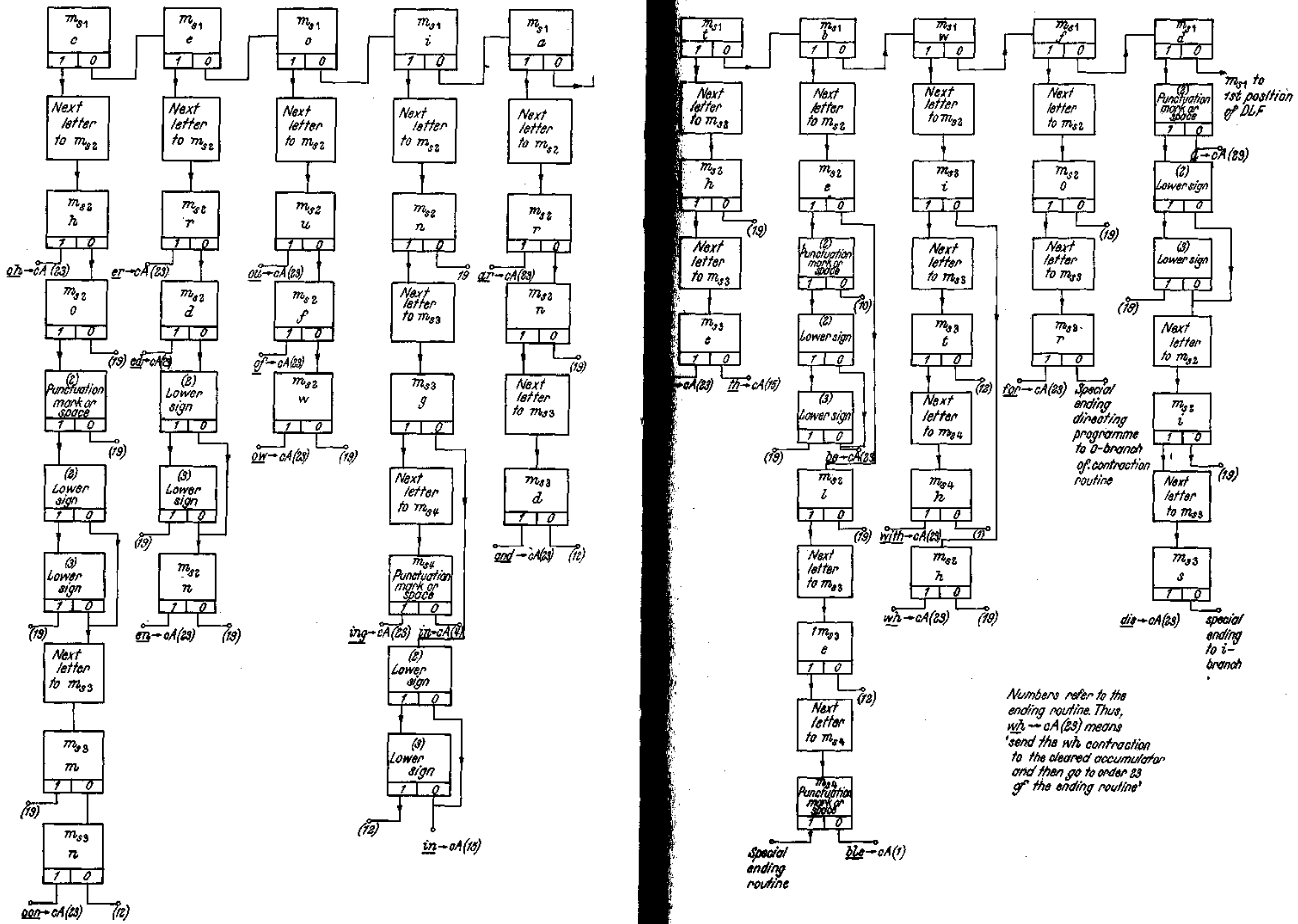
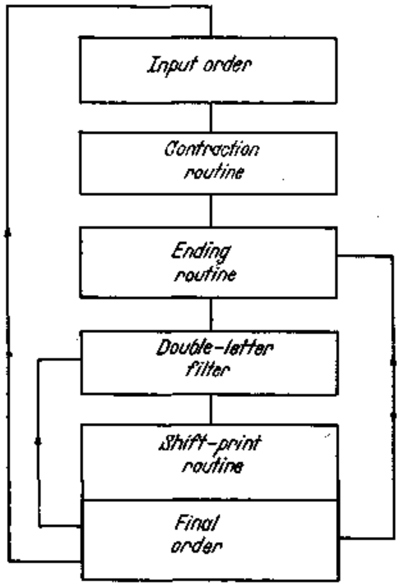

Numbers refer to the
ending routine. Thus,
wh → cA(23) means
'send the wh contraction
to the cleared accumulator
and then go to order 23
of the ending routine'

Figure 7.

Rule 35 is automatically fulfilled since the double-letter contractions are made by the double-letter filter routine after other contractions have been made. Thus, for instance, since *cc* is not a contraction in the contraction programme, the first *c* of the sequence '. . . cch . . .' is sent to the first position of the double-letter filter, from whence, after the operations of the double-letter filter programme and the shift-print routine, it is shifted to position (2). After this the second *c* is operated on by the contraction programme and the *ch* contraction is made and sent to position (1) of the double-letter filter. After further operations *c* is in (3) and *ch* in (2), and no double-letter contraction can be made. In fact, the repeated letter must reach positions (2) and (3), by which time the other contractions have been made. An overall picture of the total programme is given in *Figure 8.*

Figure 8.



The sequence of operations making up the contraction routine is seen in *Figure 7.* The figures at the conclusion of an operation refer to the part of the ending routine to which the sequence is then referred. As the incoming letters are identified they are stored consecutively in locations m_{s1}, m_{s2} m_{s3}, m_{s4}, though not all of these are used unless a contraction of four letters occurs. Thus, if 'the' occurs 't' is sent to m_{s1}, and identified by the contraction routine. The 't'-branch of this programme then identifies 'h', sending it to m_{s2}. The letter following is sent to m_{s3}. m_{s4} is not used in this case since the machine has only to deal with a 3-letter contraction.

The ending routine is concerned with setting the final order of the shift-print routine so that another complete sequence of operations can be performed automatically. Where a group of characters has been stored in the memory locations m_{s1}, m_{s2}, m_{s3}, m_{s4}, and no contraction has been found possible, the ending routine sheds these in turn to the double-letter filter, where the cycle begins again.

A few branches of the contraction routine require special ending routines.

This is the case where the second letter of a group which could be a contraction is also the first letter of another group which might be contracted. For instance, if the group 'fou' occurred, the first letter 'f' would be sent to m_{s1} and then identified by the contraction routine. The f-branch sends the second letter 'o' to m_{s2} and then identifies it. The third letter is not recognized by the f-branch, so the f-contractions cannot be made. A special ending-routine here does the following operations:

(1) (i) sends m_{s1} to double-letter filter; (ii) sets final order of shift-print routine so that after (1) (iii) the next sequence is 2; (iii) starts at first order of double-letter filter.

(2) (i) sends m_{s2} to m_{s1}; (ii) sends ms3 to ms2; (iii) goes to that part of the o-branch which identifies the second letter of an o-contraction.

Thus in the sequence '. . . fou . . .' the *ou* contraction will be made.

The shift-print routine is simply for printing the contents of position (5) of the double-letter filter and for shifting the remaining characters along one place. The sequence is as follows: 1. (5) to register; 2. left-shift 3 places to remove the code digits from machine representation of the Braille; 3. shift the character in (4) to (5); 4. shift the character in (3) to (4); 5. shift the character in (2) to (3); 6. shift the character in (1) to (2).

The final order of operation (6) contains the address of the first order of the next complete cycle of operations, and this order is set each time by the ending routine.

In dealing with grade one-and-a-half Braille, the incoming text letters are examined immediately by the contraction routine and any required operations such as contraction are delayed until the immediate context of the letter has been searched. Grade II Braille contains so many contractions that it is necessary to feed in the text-letters until a whole word is in the temporary store. This word is then subjected to a programme which compares it to a dictionary of groups which can be contracted. The whole problem then bears a closer resemblance to mechanical translation.

DISCUSSION*

R. A. FAIRTHORNE : Dr. Booth's retrieval of a dictionary word by successive dichotomy is in fact a demonstration of transcription from a constant word-length graphical number language (the integer n represented by n somethings followed by N-n non-somethings) into a variable-word-length binary radix language, *via* an operational language the two-character alphabet of which is 'inspect midway before' and 'inspect midway after'. The algebraic structures of all three languages are easily shown to be very similar.

A possible method of mechanical translation of languages, corresponding to iterative numerical computation of latent vectors, would be to make a rough translation, and from it an inverse translation back into the original language. Differences between the original and fed-back translation would initiate corrective procedures and the cycle would be repeated.

Actual mechanisms are not needed, in the first instance, to test such schemes. If they are not successful when done by human beings acting like machines they would not be successful on mechanisms.

* Discussion of this paper together with that of the preceding paper by D. Booth.

I heartily approve the Braille transcription task as being the type of small, self-contained, and not too difficult test-object that should be used for experiment, rather than grandiose projects.

Y. BAR-HILLEL: It is again a misleading extension to call what Cleave has been doing work on mechanical 'translation'. It is obviously no more, though also no less, than transcription or transliteration, on which he has been working since his output and input were not two different languages but rather two scripts of the same language. It is, of course, highly interesting that even in this task he was confronted *in noce*, so to speak, with problems that should have come up only in translation proper. In addition to the work on translation of languages restricted to limited fields, such as meteorological French, hence with restricted vocabularies, the dual project suggests itself of investigating languages with an artificially restricted syntax, such as Model English (see *Machine Translation of Languages,* Locke-Booth, 1955) or Interlingua.

Since learning to write in Interlingua for anyone who knows at least one Roman language (or English) is a relatively simple affair, the combination of writing, science and technology in Interlingua and having it mechanically translated into any other language one is interested in should prove to be a very powerful, perhaps the most powerful, approach towards effective multi-lingual propagation of scientific writing.

A. D. BOOTH in reply: The suggestion of Mr. Fairthorne is an extremely interesting one which will certainly be tried when the project of translating several times to and from several languages becomes feasible. As far as the remarks of Dr. Bar-Hillel are concerned, we can only direct his attention to the title of the paper by Mr. Cleave and observe that, whatever the niceties of the situation, it is better to have achieved an actual experimental result than to have added to the already confused screen of verbiage which surrounds the subject of mechanical translation.