THE BRAILLE TRANSLATION PROGRAM

OF MÜNSTER UNIVERSITY

by

Bernd Eickenscheidt

Let ms first give a rough summary of the German Grade 2 system. In this I can be very brief with those kinds of contractions and rules that exist in Braille Grade 2 for every language, but it might be useful to give some details of the typical German problems.

The contractions in German Grade 2 can be grouped into seven classes:

1) For simplicity of speaking let us say that the ordinary letters are contractions, too.

2) Letter contractions, such as "in", "ll", are represented by one Braille sign.

3) Prefixes, such as "ex", "pro", etc., are represented by one Braille sign if they appear at the beginning of a word.

4) Suffixes, such as "nis", "ung" (cf. English "ness" resp. "ing"), are represented by one sign. Other suffixes, such as "ation", "ativ", are represented by dot 5 plus one sign.

5) So-called "two-forms word-contractions", such as "dt" for "demokrat", "bl" for "blind", are used for isolated words, and most of them also in compound words.

6) So-called "one-form word-contractions", such as "g" for "gegen" (=against), "n" for "nicht" (=not) are used for isolated words, and about half of them are also used in compound words, where they are indicated by a dot 2 preceding them.

7) So-called "comma-contractions", such as dot 2 + "e" for "setz", dot 2 + "i" for "sitz" (stems of the verbs setzen = to set, sitzen = to sit) can be regarded as a special kind of two-forms contractions. (Of course, the second sign of a comma-contraction can be only a sign which, regarded as a one-form word-contraction, would not be allowed to appear after a dot 2, i.e., in compound words.)

Additional contractions are derived from word-contractions and comma-contractions for words or word stems containing one of the vowels a, o, or u (or the diphtong au), by the following rule: If in a derived form of a word this vowel changes to the corresponding (Umlaut" ä, ö, ü respectively, then the contraction is also used, but preceded by dot 5. If a dot 2 is involved in the contraction, the dot 5 replaces the dot 2 instead of preceding it.

Example: ⠟ is a one-form word-contraction for the word "voll" (=full)

in the compound word "vollkommen" (=perfect)

part "voll" is represented by ⠂ ⠟ ;

in the derived word "vollig" (=fully) the part "voll" is represented by ⠐ ⠟

Of the other rules that govern the use of word-contractions I shall mention only the following one:

If a one-form word-contraction appears at the beginning of a word and is followed by a dot 2 or dot 5 then the dot 2 preceding this word-contraction is cancelled.

Example: In the word "vollzumachen" (= to fill or filling, appropriate grammatical context), since the part "zu" is represented by a contraction beginning with dot 2, no dot 2 preceding ⠟ is needed.

This rule is the reason why one-form word-contractions for two-letter words, which do exist in German Grade 2 Braille, are not useless even in compound words where they must be preceded by dot 2.

The rules governing the use of the letter-contractions will be similar in most languages:

Those which coincide with punctuation signs are, of course, not applicable at the beginning or at the end of a word, respectively depending on the "left" or "right" nature of the punctuation sign. But in German Garde 2 Braille also the Braille signs for the letters x, y, q, c which are rare letters in German language (note: ch and ck, are not rare but they are contracted), are given a meaning as letter contractions for mm, el, ll and en, respectively. If one of the letters x, y, q, c appears, a dot 6 must precede the Braille sign to indicate that the following sign is to be read as Grade 1. (Note: dot 6 also serves as what is known as the "letter sign" in English Grade 2 Braille. Capitalization would be indicated by dots 4, 6 where necessary, but normally is not used since in German texts too many words are capitalized.) Other signs are provided to indicate that a complete word or more than one word must be read as Grade 1.

That is why it is not true in German Grade 2 Braille that "it is never wrong to write Grade 1". Switching to Grade 1 always must be indicated.

After this short introduction to German Grade 2 Braille let me describe how it is handled by our program, which was originally developed by Prof. Werner and Winfried Dost.

The contractions and most of the rules governing their usage are not part of the program but are the contents of a so-called "dictionary" (i.e., a card deck given to the program as input) which is designed to be easy to write and even easier to change. This seems to be the only method of making the program flexible enough to accept updates or to switch to different grades or to different languages, and this method has been chosen by later Braille translating programs in other countries, too. But the disadvantage of this method usually is a rather poor processing speed. This situation may be compared to the problems concerned with interpreting high-level programming languages, and therefore our approach is similar to how compilers process their source programs: The dictionary is transformed into a form more suitable for quick processing, and the result is kept on a magnetic disk or tape. This step need be done only once when a dictionary has been newly written or has been updated. For actual Braille translations only this "transformed" version of the dictionary is used.

The "source" dictionary consists of a set of entries, each of which corresponds to a different contraction or "pseudo-contraction" (the latter will be explained later). Remember that we call an ordinary letter a contraction, too. Every such entry is one or two punched cards long. It contains

1) the inkprint character string to be translated,

2) a (possibly empty) sequence of conditional translation instructions to be "tried" sequentially,

3) an unconditional translation instruction to be used "else".

Each translation instruction contains

a) a code giving the class of the contraction used, indicating e.g., whether the translation begins with the Braille sign for one of the letters A to J, or whether the translation is to be regarded as ending with a vowel, etc. (Special codes are used for digits, punctuation signs, format control, etc.)

b) the Braille signs for the translated left part of the inkprint character string

c) the length of the (right) part of the inkprint character string which remains untranslated.

The conditional translation instructions have a set of codes added that indicate the conditions all of which must be satisfied in order to allow the use of this translation. Conditions that can be tested are: beginning of a word, ending of a word, following a vowel, preceding the letter E, etc.

Examples: The entry DEMOKRAT #64, 131036 has the unconditional translation instruction only, where 64 is the code for a two-form word-contraction (not ending with a vowel) and 131 and 036 represent the Braille signs ⠙ and ⠞ respectively, in octal coding preceded by a parity bit. The length of the untranslated part is zero and is not written.

The entry EM #+WA, +WE # 42, 021115#48, 167. contains one conditional translation instruction. WA means that the condition Wort-Anfang (beginning of a word, more precisely: we are just behind punctuation and/or blanks) is to be tested, the plus sign means that the condition must be satisfied to make the test result positive (a minus sign would indicate that the condition must not be satisfied), the condition WE (Wort-Ende = end of a word) is tested analogously. 42 means ordinary letter(s) the first of which is between A and J, and the last of which is not a vowel. 021 and 115 represent the Braille signs ⠑ and ⠍, 48 means letter contraction, and 167 represents ⠷ .

The whole entry briefly reads: EM is contracted except when isolated.

But this is not completely correct, and that leads us to the "pseudo-contractions" mentioned earlier. For example in the word "Ehemann" (=husband) the contraction for "mann" takes precedence over the one for "em". Although, of course, an appropriate entry for "mann" is in the dictionary, the program (which scans the input from the left to the right) would no longer find "mann" after "em" had been translated. Therefore an entry is added to the dictionary for the pseudo-contractions EMANN:

EMANN#43, 021, 4.

This reads: When EMANN is found, the translation result is the vowel (43) ⠑ , and 4 inkprint characters remain untranslated. The program selects this entry, where applicable, with higher priority than the one for EM, because the string EMANN is longer.

Let me now give a rough description of the program logic. The translation algorithm is as follows:

1) Initialize status variables, etc.

2) Find the dictionary entry with the longest inkprint character string matching the beginning of the input queue.

3) Set those variables which are concerned with the inkprint image to the right of the matched string, e.g., the variable "end of a word", to their correct values.

4) Find the first valid translation instruction of the selected entry.

5) If the last translation was a one-form word-contraction at the beginning of a word, and the new one does not begin with dot 2 or dot 5, then send a dot 2 to the output queue, followed by the contents of the "waiting queue" (cf. paragraph 7).

6) Test the current status against the class of the incoming contraction whether delimiters are required; e.g., a number sign before a digit (if the last translation was not a digit, too), a proper delimiter before the letters A to J after digits, or a dot 2 before a one-form word-contraction not at the beginning of a word, etc. Send these delimiters to the output queue.

7) If the new translation is a one-form word-contraction at the beginning of a word, and not isolated, then send it to a "waiting-queue" and keep this fact in mind.

8) If paragraph 7 does not apply, then send the new translation to the output queue.

9) Update the status variables according to the contraction code that was in the translation instruction.

10) Cancel the proper number of characters from the input queue, and resume processing at paragraph 2.

Of course, the program provides sections which deal with necessary operations like shifting and filling the input queue, shifting and clearing the output queue, starting new output lines and pages, centering headings, numbering pages, etc.

But the only thing important for the translation to be correct is the dictionary, and the most important thing for the translation to be fast is paragraph 2. Therefore I shall describe now the "transformed" version of the dictionary. It has a tree structure, i.e., a system of "records" or "nodes" chained together by several pointers. The records can be considered to be named by character strings of varying length. And every proper left substring of the inkprint character string of any entry to the (source) dictionary occurs as a name of such a record. The root of the tree is the record named by the empty string, e.g., the string of length zero. Each record contains a table of pointers; one pointer for each possible character. Regarding such a record, for each possible character let us call the "new name" the name of this record with the "new" character concatenated to it on the right. Then every pointer in this table either points to the record with the "new name" (if such a record exists), or gives the address of the stored sequence of translation instructions for the dictionary entry having the longest inkprint character string matching a left substring of the "new name".

Examples:

root record:

⋮	⋮
D	——> record D
E	——> record E
⋮	⋮
X	——> transl X
⋮	⋮

(No entry in the dictionary except the entry for "X" itself has an inkprint character string beginning with "X".)

record "D":

⋮	⋮
D	——>transl D
E	——>record DE
⋮	⋮

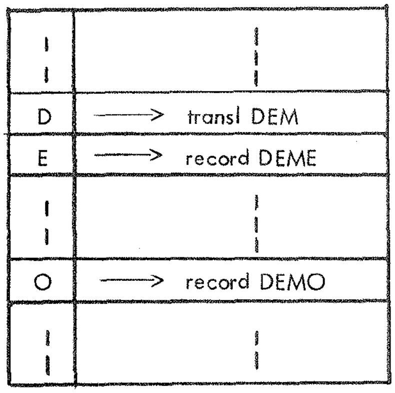
(No strings beginning "DD" occur)

record "DE":

⋮	⋮
D	——>transl D
⋮	⋮
M	——> record DEM
⋮	⋮

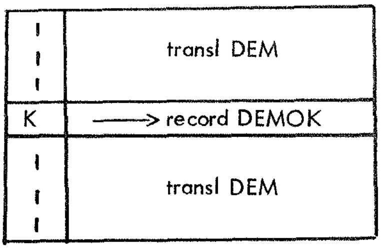
(No strings beginning "DED" occur, and "DE" does not occur.)

record "DEM":



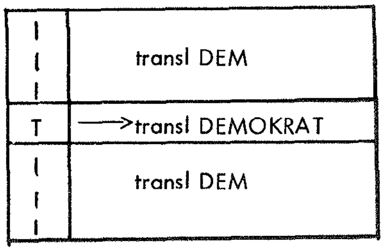
(No strings beginning "DEM" occur, but an entry for "DEM" exists.)

record "DEMO"



(No strings beginning "DEMO" occur, except "DEMOKRAT".)

record "DEMOKRA":



When the program now has to "find the dictionary entry with the longest inkprint character string matching the beginning of the input queue", all it has to do is to start at the root record, scan the input queue, character by character, and with every new character follow the correct pointer until it arrives at a translation.

For the selection of the correct pointer as a function of the next character to be processed, two different kinds of such records are provided: one uses the character's internal coding as an index to the table, this method is particularly fast; the other one uses a search technique, with the corresponding records requiring as little storage as possible. The first one applies to the root record and to those records which either are frequently used or contain many different pointers; the second one is useful especially for such records as "DEMO", "DEMOK", "DEMOKR", "DEMOKRA" which contain only few or even only one pointer besides the one which one might call the "else"-pointer. The decision for which records the fast (and storage consuming) system is used is given as input data to the program which "compiles" the dictionary into the format just described.

Statistics

The current version of the program, which is written in PL/I, with the current version of the German Grade 2 dictionary, which contains about 750 entries and which uses the storage-consuming system in 17 records, operates in 100K bytes of storage of an IBM 360/50 with a speed of about 3,000 - 4,000 words per minute. (German words are long words!)

More precisely: The translation of one page of 28 lines with 36 signs per line takes 4 seconds of CPU time. This is about the same time which the IBM 1403 line printer with an appropriately loaded UCS buffer needs to print this page. (Note that a Braille line is three lines to the printer.) If the Braille output is punched on cards, which are to control the automatic embossing machine, then this page takes about 16 cards. This is somewhat less than a modern card punch can punch in those 4 seconds. The embossing machine takes 5 minutes for a page.

The current version of the dictionary produces about 0.9 serious miscontractions per page, plus about 2 deviations from the rules caused by not using some contractions under certain conditions. With three tiny changes in the dictionary these numbers 0.9 and 2 could easily be changed to 1.5 and 0.1, respectively.

A new version of the dictionary will soon be implemented.