[From: A.Zampolli, N.Calzolari (eds.) *Computational and mathematical linguistics*. Proceedings of the International Conference on Computational Linguistics, Pisa, 1973, vol.II (Firenze: Olschki, 1980)]

WOLFGANG A. SLABY

A UNIVERSAL BRAILLE TRANSLATOR

As we will attempt to formalize the translation process of inkprint into grade 2 braille, we first must explain what we mean by such notions as "inkprint" and "grade 2 braille". For this reason we first of all give a formal definition of the terms "alphabet" and "word over some alphabet".

DEFINITION 1:

Each finite nonempty set Σ is called an *alphabet*. In our problem we are dealing mainly with two alphabets:

1) the alphabet of Latin capital letters, arabic numerals, punctuation marks, and perhaps some other special characters

$$\Sigma_1 := \{A, B, C, ..., X, Y, Z, 0, 1, 2, ..., 7, 8, 9, .,; ,:, \text{etc.}\}$$

2) the alphabet of braille symbols

$\Sigma_2 := \{$ ⠁, ⠂, ⠄, ..., ⠿ $\}$

consisting of $2^6 - 1 = 63$ symbols, where each braille symbol is composed of one to six *dots* organized as a matrix of three rows and two columns).

Each finite (possibly empty) sequence

$$w = x_1\ x_2\ ...\ x_{n-1}\ x_n$$

of symbols $x_1, x_2 ..., x_{n-1}, x_n$ of an alphabet Σ is called a "word over Σ".

The "empty word", i.e. the word (over Σ) consisting of zero symbols, is denoted by ε. The set of all words over an alphabet Σ, including ε, is denoted by $\Sigma^\star$, that is

$$\Sigma^\star := \{w \mid \exists n \in \mathbb{N}_0 : \exists x_1, ..., x_n \in \Sigma : w = x_1 \ ...\ x_n\}.$$

Each subset L of $\Sigma^\star$ ($L \subseteq \Sigma^\star$) is called a " formal language over Σ ".

If $v = x_1 x_2 \ldots x_r$ and $w = y_1 y_2 \ldots y_s$ are words over Σ one defines the " concatenation of v and w " by

$$vw := x_1 \; x_2 \; \ldots \; x_r \; y_1 \; y_2 \; \ldots \; y_s$$

(In formal language theory $\Sigma^\star$ together with the operation of concatenation is called the *free semigroup generated by* Σ).

Let x and y be any words over some alphabet Σ. We define:

$$\text{" } x \text{ part-word of } y \text{ ": } \Leftrightarrow \; \exists u, \; v \in \Sigma^\star : y = uxv$$

After these preliminary definitions we are able to explain what is meant by the terms " inkprint " and " braille ".

If we consider some suitably chosen formal language L_1 over the alphabet Σ_1 we call any word $w \in L_1$ (resp. any text built up of words of L_1) an " inkprint word " (resp. text), whereas each word over the alphabet Σ_2 is called a " braille word ". Now when we translate an inkprint word into braille character by character there are no translation problems at all if there is an one-to-one transformation C

$$C : \Sigma_1 \rightarrow \Sigma_2$$

which associates each symbol of Σ_1 with a unique welldefined symbol of Σ_2.

But you can easily imagine that the one-by-one translation of a normal inkprint book into braille consisting of about 300,000 characters will result in an enormous stack of braille-printed pages. Therefore, in each language there has been developed a system of rules which allows people to translate some inkprint words or part-words of inkprint words without translating character by character but using only one or two braille characters as translation. In these cases the translation is called a " full-word contraction " resp. a " part-word contraction " in braille. The result of a translation into braille which uses full-word and part-word contractions is called " grade 2 braille ".

It is evident that the use of contractions brings up new problems of high linguistic complexity because in the grade 2 braille definition of each language there exist a lot of special rules and exceptions in order to avoid incorrect applications of full-word or part-word contractions.

After this introduction we can easily formalize the process of translation into grade 2 braille. For if we disregard all the inherent problems of translation into grade 2 braille that arise from the language-dependent definition of the system of rules which controls the use of full-word or part-word contractions, we can regard a translation of inkprint into grade 2 braille as a one-to-one transformation

$$T\colon L_1 \to \Sigma_2^\star,$$

mapping a suitably chosen formal language L_1 over the alphabet Σ_1 into the set of all braille words $\Sigma_2^\star$.

For example, let L_1 be the formal language over Σ_1 consisting of all orthographically correct written German words and let T_G be the braille-translation-mapping according to the German grade 2 braille definition. Then we can write the translation of some word ERDENKEN $\in L_1$ (meaning: *to imagine, to think out*) into braille as

ER D EN K EN

T_G (ERDENKEN) = .

In order to develop an algorithm computing automatically such a braille-translation-mapping T we must investigate the structure and internal definition of this mapping. In each language the braille-translation-mapping T is defined by a system of rules which control the translation (of parts) of any word. In the definition of the German grade 2 braille, for example, we have, besides others, the rules

ER is translated into the braille symbol 

EN is translated into the braille symbol 

EL is translated into the braille symbol 

LL is translated into the braille symbol 

H is translated into the braille symbol 

E is translated into the braille symbol 

L is translated into the braille symbol 

N is translated into the braille symbol 

These translation rules can easily be formalized by the notion of "production rule".

DEFINITION 2: *System of production rules (semi-THUE-system)*

A pair $(\Gamma, \mathfrak{R})$ is called a "system of production rules" (or semi-THUE-system) if and only if the following holds:

1) Γ is an alphabet
2) $\mathfrak{R}$ is a finite, nonempty set of words of the structure

$$u \to v \text{ with } u,\ v \in \Gamma^\star \text{ and } \to \notin \Gamma;$$

$\mathfrak{R}$ is called the set of "production rules". (More formally one would define:

$$\mathfrak{R} \subseteq \Gamma^\star \times \Gamma^\star \text{ finite and nonempty,}$$

and then each production rule is some pair (u, v) with $u, v \in \Gamma^\star$).

Now we must define what it should mean to apply a production rule to a word over the alphabet Γ. Let $(\Gamma, \mathfrak{R})$ be a system of production rules, $w \in \Gamma^\star$ any word over the alphabet Γ, and let $u \to v \in \mathfrak{R}$ be any production rule.

We define

$$\text{"}u \to v \text{ is applicable to } w\text{"}: \Leftrightarrow u \text{ is a part-word of } w$$
$$(\Leftrightarrow \exists x,\ y \in \Gamma^\star:\ w = xuy)$$

If $u \to v$ is applicable to w, then $z := xvy$ is called "the result of the application of $u \to v$ to w".

Let us try an example. If we formalize the above mentioned braille-translation-rules we obtain the following special system of production rules $(\Gamma, \mathfrak{R})$ with

$\Gamma = \Sigma_1 \cup \Sigma_2$, Σ_1 and Σ_2 as in Def. 1

$\mathfrak{R} = \{$ER $\to$ ⠻, EN $\to$ ⠉, EL $\to$ ⠽, LL $\to$ ⠟,
N $\to$ ⠝, H $\to$ ⠓, E $\to$ ⠑, L $\to$ ⠇$\}$

Looking at the word $w =$ LERNEN (meaning "to learn") we easily see that the production rule ER $\to$ ⠻ is applicable to w [with $x =$ L and $y = \mathrm{NEN}$]. The result of the application of ER → ⠻ to w is $z_1 =$ L⠻NEN. Next, the production rule EN → ⠉ is applicable to z_1 [with $x =$ L⠻N and $y = \varepsilon$] and the result of the application of EN → ⠉ to z_1 is $z_2 =$ L⠻N⠉. Lastly, we apply the production rules L → ⠇ and N → ⠝ and obtain as the final result the correct braille-translation of LERNEN, that is ⠇ ⠻ ⠝ ⠉. At this point one should remark that there is some arbitrariness in the choice of the production rule that should be applied, if more than one production rule is applicable to the word w. In the first step of the derivation above, for example, the five production rules ER → ⠻, EN → ⠉, L → ⠇, E → ⠑, and N → ⠝ are applicable to $w =$ LERNEN. If we would apply the production rule E → ⠑ processing the first letter E, we would not obtain the corret braille-translation of LERNEN.

Let us consider another example, the German word HELL (meaning " bright, luminous "). In translating this word into braille one proceeds similarly as above applying perhaps first the production rule H → ⠓. But in the next step we are faced with the problem as to whether we must translate first EL into ⠽ and then L into ⠇ or first E into ⠑ and then LL into ⠟, that is whether we must apply first the production rule EL → ⠽ and then the production rule L → ⠇ or first the production rule E → ⠑ and then LL → ⠟, leading to different results.

As the braille-translation-mapping must be uniquely defined there exists a meta-rule in the German definition of braille which states that the translation of LL has higher priority than the translation of EL.

Thus, if we attempt to formalize correctly the internal definition of a braille-translation-mapping, we have to develop an analytical tool which enables us to take into consideration the meta-rule of *priority*. The adequate analytical tool of formal language theory is the concept of a " MARKOV system of production rules ".

DEFINITION 3: MARKOV *system of production rules*

A quadrupel $\mathfrak{M} = (\Sigma, \Delta, \Gamma, \mathfrak{R})$ is called a " MARKOV system of production rules " if and only if the following is valid:

1) Σ, Δ, Γ are alphabets with $\Sigma \subseteq \Gamma$ and $\Delta \subseteq \Gamma$ (Σ is called

the " input alphabet ", Δ the " output alphabet ", and Γ the " working alphabet ".)

2) $(\Gamma, \mathfrak{R})$ is a system of production rules and $\mathfrak{R}$ is an *ordered* set. Let w be any word over Γ. We define:

" $\mathfrak{M}$ is applicable to w "
: $\Leftrightarrow$ there exists $u \to v \in \mathfrak{R}$
such that $u \to v$ is applicable to w.

Let $\mathfrak{M}$ be applicable to w and let $u_\emptyset \to v_\emptyset$ be the *first production rule* – first according to the order defined on $\mathfrak{R}$ –, which is applicable to w. Then, since $u_\emptyset$ is a part- word of w, there exist $x, y \in \Gamma^\star$ with x of minimal length such that $w = x\ u_\emptyset\ y$.

Then define:

$$\mathfrak{M}\ (w) := x\ v_\emptyset\ y,$$

and by means of induction we define $\mathfrak{M}^n\ (w)$ as

$$\mathfrak{M}^0\ (w) := w$$

$$\mathfrak{M}^n\ (w) := \mathfrak{M}\ (\mathfrak{M}^{n-1}(w)) \text{ for } n \in \mathbb{N}, \text{ provided that for each } i\ (0 \leq i \leq n\text{-}1)\text{: } \mathfrak{M} \text{ is applicable to } \mathfrak{M}^i\ (w).$$

Now it is easy to see that for each $w \in \Sigma^\star$ exactly *one* of the following two cases is satisfied:

case 1: there exists an integer $r_w \in \mathbb{N}_\emptyset$ such that for each i $(0 \leq i \leq r_w\text{-}1)$ $\mathfrak{M}$ is applicable to $\mathfrak{M}^i\ (w)$ and $\mathfrak{M}$ is *not* applicable to $\mathfrak{M}^{r_w}(w)$.

case 2: for each $n \in \mathbb{N}_\emptyset$ $\mathfrak{M}$ is applicable to $\mathfrak{M}^n\ (w)$.

If in the first case we have $\mathfrak{M}^{r_w}\ (w) \in \Delta^\star$ we call $\mathfrak{M}^{r_w}\ (w)$ the result of the application of $\mathfrak{M}$ to w.

Since we are engaged in the formalization of a braille-translation-mapping T_{sl} transforming any special language (*sl*) into the corresponding grade 2 braille, the appropriate MARKOV system of production rules $\mathfrak{M}_{sl}$ generally has the form

$$\mathfrak{M}_{sl} = (\Sigma_1,\ \Sigma_2,\ \Gamma,\ \mathfrak{R}_{sl}).$$

Moreover, it turns out to be possible to choose $\mathfrak{R}_{sl}$ in such a manner that for all $w \in \Sigma_1^\star$ case 1 of the above-mentioned definition is satisfied.

Let us consider as an example the special MARKOV system of production rules

$\mathfrak{M} = (\Sigma_1, \Sigma_2, \Gamma, \mathfrak{R})$ with
$\Gamma = \Sigma_1 \cup \Sigma_2$ and
$\mathfrak{R} = \{$1. ER → ⠻, 2. LL → ⠟, 3. EN → ⠉,
4. EL → ⠽, 5. E → ⠑, 6. H → ⠓,
7. L → ⠇, 8. N → ⠝$\}$

together with the word w = ERHELLEN (meaning " to illuminate ").
Then $\mathfrak{M}$ is applicable to w and

$\mathfrak{M}$ (w) = ⠻ HELLEN.

Next $\mathfrak{M}$ is applicable to z_1 = ⠻ HELLEN and we have

$\mathfrak{M}$ (z_1) = ⠻ HE ⠟ EN.

Now $\mathfrak{M}$ is applicable to z_2 = ⠻ HE ⠟ EN yielding

$\mathfrak{M}$ (z_2) = ⠻ HE ⠟ ⠉

and in the last two applications we obtain

$\mathfrak{M}$ (⠻ HE ⠟ ⠉) = ⠻ H ⠑ ⠟ ⠉,

and finally $\mathfrak{M}$ (⠻ H ⠑ ⠟ ⠉) = ⠻ ⠓ ⠑ ⠟ ⠉, which is the correct translation of the word ERHELLEN into German grade 2 braille. Moreover $\mathfrak{M}$ now is no longer applicable.

If we suppose we have already developed a complete special MARKOV system of production rules $\mathfrak{M}_{sl}$ according to the translation rules of a special language into the corresponding grade 2 braille, we now can easily describe the braille-translation-mapping T_{sl} of this language as

$$\boxed{T_{sl}\ (w) = \mathfrak{M}_{sl}^{r_w}\ (w)}$$

Thus, the algorithm for computing the braille-translation-mapping T_{st} of a special language is nothing but an algorithm which carries out the application of a special MARKOV system of production rules to any supplied word $w \in \Sigma_1^*$. This algorithm will be called a "universal MARKOV algorithm".

But before I give a description of the universal MARKOV algorithm let me say some words about the developing of a complete MARKOV system of production rules. This step of developing a complete MARKOV system $\mathfrak{M}_{st}$ corresponding to the definition of the grade 2 braille of this special language is a very difficult linguistic problem (with the exception perhaps of the excellently reformed Danish grade 2 braille (V. PÅSKE, J. VINDING), where, I suppose, the problem is much easier). The essential difficulties mainly come from the complicated rules that control the application of any full-word or part-word contraction. For instance, in the words *subscription, addition, substitution,* and *convention,* which have counterparts in the German language with almost the same spelling, the part-word *ion* must be contracted according to the definition of German grade 2 braille, whereas it *must not* be contracted in German words like *Ionisierung* (meaning: "ionisation") or *Radionetz* (meaning: "radio network"), because in the first case the part-word *ion* appears at the beginning of the word and in the second case it goes across the word boundary between the two words *Radio* and *Netz.*

The solution of these difficulties, arising for the most part from the problem of translating correctly compound words into grade 2 braille, results in a rapid increase of the number of production rules. A forthcoming research paper of J. Splett and myself will report our approach to overcome these difficulties by means of some language-independent linguistic tools, and moreover will present, we hope, the complete MARKOV system of production rules corresponding to the definition of the German grade 2 braille.

The remaining task I have yet to do is to construct the universal MARKOV algorithm (which in our context we can call a universal braille translator) that carries out the application of any MARKOV system of production rules to any supplied word. This algorithm is given in a PL/I-like form, where the italicized words denote keywords of PL/I. It is evident that a programmed version of this algorithm should be of high efficiency, therefore, very efficient list-processing techniques must be used.

Let *WORD* be the variable which will take the value of the supplied word of Σ_1^* *LSIDE*(*N*) and *RSIDE*(*N*) two one-dimensional arrays of length *N* which will take the left sides and right sides respectively of the production rules of the considered MARKOV system. (*N* too will be supplied as a parameter and then storage for *LSIDE*(*N*), *RSIDE*(*N*) will be allocated).

Then the essential part of the algorithm is defined by the following instructions:

```
LOOP:
        do I = 1 to N;
        compute M = index (WORD, LSIDE(I));
        if M > Ø
        then do;
        substitute the occurrence of LSIDE(I) in WORD
          starting with position M by RSIDE(I);
        go to LOOP;
        end;
end;
```

Evidently this is a very easy algorithm and it is completely language-independent. Therefore it is in order to call this a "universal braille translator". The language-dependent part consists only of the respective special MARKOV system of production rules which is supplied to the universal braille translation as a parameter.

Conclusion.

Whether this algorithm is very practical or not is not yet clear to us because we expect that the MARKOV system of production rules corresponding to the definition of the German grade 2 braille can have a size of perhaps more than one thousand production rules. But nevertheless, at least from a theoretical point of view, this formalization of grade 2 braille definitions has the following main advantages:

1) formalizing a verbal definition of any grade 2 braille is a great help in localizing ambiguities and perhaps even contradictions which can be solved only reforming the definition of this grade 2 braille.

2) The MARKOV system of production rules provides an excellent device for comparing and measuring the complexity of the different definitions of grade 2 braille in the different languages. (Here the com-

plexity $K(\mathfrak{M}_{sl})$ of a MARKOV system can be defined as the product of the number of production rules with the average length of the left side of a production rule.)

Finally let me remark that the MARKOV system of production rules turns out to be a more adequate formalization of the braille translation process than the concept of finite-state syntax-directed braille translation, as presented by J. K. MILLEN (1970). For if we consider, for instance, the translation of the word *23 yds.* which should produce as output the braille signs for YD23, we easily can write down (in a condensed form) some production rules of a MARKOV system handling this translation:

$$1)\ \{\emptyset|1|\ \ldots\ |9\}\sqcup \mathit{yds.} \rightarrow \mathit{yds.}\ \{\boxed{\emptyset}\ |\ \boxed{1}\ |\ \ldots\ |\ \boxed{9}\}$$

$$2)\ \{\emptyset|1|\ \ldots\ |9\}\quad \mathit{yds.} \rightarrow \mathit{yds.}\ \{\boxed{\emptyset}\ |\ \boxed{1}\ |\ \ldots\ |\ \boxed{9}\}$$

$$3)\qquad \sqcup \mathit{yds.} \rightarrow \boxed{Y}\ \boxed{D},$$

whereas " reversing the order is *not* possible with a finite-state machine " (J. K. MILLEN, 1970).

REFERENCES

V. PÅSKE, J. VINDING, *The "perfect" Braille System*, Statens Institut for Blinde og Svagsynede, København.

J. K. MILLEN, *DOTSYS II: finite-state syntax-directed braille translation*, The MITRE corporation, July 1970.

FREUND, MITTELSTEN SCHEID, BRITZ, *Systematik der deutschen Blindenkurzschrift* (Entwurf), Deutsche Blindenstudienanstalt, Marburg 1972.