THE MARKOV SYSTEM OF PRODUCTION RULES

- A UNIVERSAL BRAILLE TRANSLATOR -

by

Wolfgang A. Slaby

Abstract

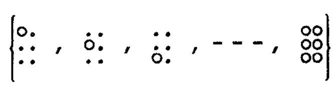
It is the goal of this report to present the concept of MARKOV algorithm as a universal translator into Braille, which is applicable to every language and every definition of the respective (Grade 1 or Grade 2) Braille and to which all the language-dependent components are supplied as parameters.

This report describes in detail the concept of MARKOV algorithm and attempts to make evident that this concept is an adequate general formalization of the translation process of inkprint into Braille.

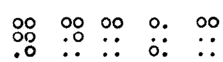
For some years past there have been developed several methods and algorithms for the different languages in order to translate automatically an inkprint text of a special language into Grade 1 or Grade 2 Braille. As in the diverse languages there arise very different problems in automatizing this process of translation, naturally each of these algorithms possesses a lot of language-dependent components.

In order to localize these language-dependent components which mainly come from the definition of the Grade 2 Braille in the respective language we first of all will analyse only the process of translation independent of a special language and a special definition of the respective Braille.

If we disregard all the inherent problems of translation into Grade 2 Braille which arise from the special language we can say that the translation of a word of a special language into the respective Braille is nothing but mapping a string of symbols of some finite alphabet (in this case the alphabet of Latin capital letters, punctuation marks, arabic numerals and other special characters) onto another string of symbols of another finite alphabet (in this case the alphabet of Braille symbols:)



For example, in the process of the translation into Grade 2 Braille according to the German rules of Braille the strings of symbols ERDENKEN (meaning to imagine, to think out) will be mapped into



ER D EN K EN

In order to formalize these ideas we use the following definition:

Definition 1:

Each finite nonempty set Σ is called an alphabet.

[In our problem we are dealing mainly with two alphabets:

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

$$\Sigma_1 = \{A, B, C, \ldots, X, Y, Z, 0, 1, 2, \ldots, 9, ., ?, !, :, ,, \$, \ldots\}$$

2) the alphabet of Braille symbols

$$\Sigma_2 = \{ ⠁ , ⠂ , ⠄ , \ldots , ⠿ \}$$

consisting of 63 symbols.]

Each finite (possibly empty) sequence $w = x_1 x_2 \ldots x_n$ of symbols $x_1, x_2, \ldots, x_n$ 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 ξ.

The set of all words over an alphabet Σ, including ξ, is denoted by Σ^*, that is

$$\Sigma^* = \{ w \mid \exists \; n \in N_0 : \exists \; x_1, \ldots, x_n \in \Sigma : w = x_1 \ldots x_n \}$$

Each subset L of Σ^* ($L \subseteq \Sigma^*$) 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

$$v \cdot w := x_1 x_2 \ldots x_r y_1 y_2 \ldots y_s$$

[In formal language theory Σ^* 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:

x subword of y

$:\Leftrightarrow$ there exist $u, v, \in \Sigma^*$
such that $y = u\,x\,v$

After these preliminary definitions we can return to the formalization of the process of translation into Braille. Under this formal aspect we can regard a translation of inkprint into Braille as a one-to-one transformation

$$T = L_1 \longrightarrow L_2 ,$$

mapping a formal language L_1 over the alphabet Σ_1 into another formal language L_2 over the alphabet Σ_2 of Braille symbols. For example, let L_1 be the formal language over Σ_1 consisting of all orthographically correct written German words and $L_2 = \Sigma^*_2$ and let T_G be the Braille-translation-mapping according to the German Grade 2 Braille definition. Then we can write the above translation of ERDENKEN $\in L_1$ into Braille as

$$T_G (\text{ERDENKEN}) = ⠻ ⠙ ⠉ ⠅ ⠉$$

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 ⠓

L is translated into the Braille symbol ⠇

E 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 rules.

Definition 2: [system of production rules (semi-THUE-system)]

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

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

   $u \longrightarrow v$ with $u, v \in \Gamma^*$ and $\longrightarrow \notin \Gamma$;

   $\mathcal{R}$ is called the set of production rules.

   [More formally one would define:

   $\mathcal{R} \subseteq \Gamma^* \times \Gamma^*$ finite and nonempty,

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

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

We define

$u \longrightarrow v$ is applicable to w

$:\Leftrightarrow$ there exist $x, y \in \Gamma^*$ such that $w = x\,u\,y$

If $u \longrightarrow v$ is applicable to w, then $z := x\,v\,y$ is called the result of the application of $u \longrightarrow v$ 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:

$(\Gamma, \mathcal{R})$ with

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

$\mathcal{R}$ = {ER $\longrightarrow$ ⠻ , EN $\longrightarrow$ ⠉ , EL $\longrightarrow$ ⠽ , LL $\longrightarrow$ ⠟ , N $\longrightarrow$ ⠝ , H $\longrightarrow$ ⠓,

E $\longrightarrow$ ⠑ , L $\longrightarrow$ ⠇}

Looking at the word w = LERNEN (meaning to learn) we easily see that the production rule ER $\longrightarrow$ ⠻ is applicable to w [with x = L and y = NEN]. The result of the application of ER $\longrightarrow$ ⠻ to w is z_1 = L⠻NEN. Next the production rule EN $\longrightarrow$ ⠉ is applicable to z_1 [with x = L⠻N and $y = \xi$] and the result of the application of EN $\longrightarrow$ ⠉ to z_1 is z_2 = L⠻N⠉. Last we apply the production rules L $\longrightarrow$ ⠇ and N $\longrightarrow$ ⠝ 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 $\longrightarrow$ ⠻ , EN $\longrightarrow$ ⠉ , L $\longrightarrow$ ⠇ ,

$E \longrightarrow$ and $N \longrightarrow$ are applicable to w = 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 $H \longrightarrow$. But in the next step we are faced with the problem 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 $EL \longrightarrow$ and then the production rule $L \longrightarrow$ or first the production rule $E \longrightarrow$ and then $LL \longrightarrow$, 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 <u>priority</u>. The adequate analytical tool of formal language theory is the concept of a <u>MARKOV system of production rules.</u>

Definition 3: [MARKOV system of production rules]

A quadrupel $\mathfrak{M} = (\Sigma, \Delta, \Gamma, \mathfrak{R})$ is called a <u>MARKOV system of production rules</u> 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 <u>ordered</u> set.

Let w be any word over Γ. We define:

<u>$\mathfrak{M}$ is applicable to w</u>

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

Let $\mathfrak{M}$ be applicable to w and let $u_0 \longrightarrow v_0$ be the <u>first production rule</u> - first according to the order defined on $\mathfrak{R}$ - , which is applicable to w. Then since u_0 is a subword of w there exist <u>x, y $\in \Gamma^*$ with x of minimal length</u> such that $w = x\, u_0\, y$.

Then define:

$$\underline{\mathfrak{M}(w) := x\, v_0\, y,}$$

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

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

$$\mathfrak{M}^n(w) := \mathfrak{M}\left(\mathfrak{M}^{n-1}(w)\right) \text{ for } n \in N, \text{ provided that } \mathfrak{M} \text{ is applicable to } \mathfrak{M}^{n-1}(w).$$

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

case 1: there exists an integer $r_w \in N_0$ such that $\mathfrak{M}$ is applicable to $\mathfrak{M}^{r_w - 1}(w)$ and $\mathfrak{M}$ is <u>not</u> applicable to $\mathfrak{M}^{r_w}(w)$.

case 2: for each $n \in N_0$ $\mathfrak{M}$ is applicable to $\mathfrak{M}^n(w)$.

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

Since we are engaged in the formalization of a Braille-translation-mapping $T_{spec.lang.}$ transforming any special language into the corresponding Grade 2 Braille, the appropriate MARKOV system of production rules $\mathfrak{M}_{spec.lang.}$ generally has the form

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

Moreover it turns out to be possible to choose $\mathfrak{R}_{spec.lang.}$ in such a manner that for all $w \in \Sigma_1^*$ 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 $\mathfrak{M}$spec. lang. according to the translation rules of a special language into the corresponding Grade 2 Braille, we now can easily describe the Braille-translation-mapping Tspec. lang. of this language as

$$T_{spec.\,lang.}\ (w) \simeq \mathfrak{M}^{\ulcorner w}{}_{spec.\,lang.}\ (w)$$

Thus the algorithm for computing the Braille-translation-mapping Tspec. lang. 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 usually is called a <u>universal MARKOV algorithm.</u>

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 $\mathfrak{M}$spec. lang. 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 [1], where as I suppose, the problem is much easier). The essential difficulties mainly come from the local ambiguities. For instance in the German words "verherrlicht" (meaning glorified) and "ermoglicht" (meaning rendered possible) the subword "lich" corresponding to the English suffix "ly" must be contracted according to the German definition of Grade 2 Braille, whereas it must not be contracted in German words like "talglicht" (meaning tallow-candle) and "Tageslicht" (meaning light of day). The solution of these local ambiguities results in a rapid increase of the number of production rules. A forthcoming research paper of J. Splett, H. Kamp, and me will report our approach to overcome these difficulties by means of some language-independent linguistic tools, and moreover will present, as 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 underlined words denote keywords of PL/I. It is evident that a programmed version of this algorithm should be of high efficiency, on account of which there must be used very efficient list-processing techniques.

Let WORD be the variable which will take the value of the supplied word of $\sum_1^*$, 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 after that 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>0
    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 translator 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 five hundred 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 by 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 complexity $K(\mathfrak{M}_{\text{spec. lang.}})$ 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 [2]. 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. $\{0 \mid 1 \mid \dots \mid 9\} \sqcup \text{yds.} \longrightarrow \text{yds.} \; \{\boxed{0} \mid \boxed{1} \mid \dots \mid \boxed{9}\}$
2. $\{0 \mid 1 \mid \dots \mid 9\} \; \text{yds.} \longrightarrow \text{yds.} \; \{\boxed{0} \mid \boxed{1} \mid \dots \mid \boxed{9}\}$
3. $\sqcup \text{yds.} \longrightarrow \boxed{Y} \; \boxed{D}$,

whereas "reversing the order is not possible with a finite-state machine" [2].

References

1 V. Påske, J. Vinding, The "Perfect" Braille System, Statens Institut for Blinde og Svagsynede, Kobenhavn

2 J. K. Millen, DOTSYS II: Finite-State Syntax-Directed Braille Translation, July 1970, The MITRE Corporation