DOI: 10.14746/linpo.2022.64.1.3

A Multi-Layer Transcription Model – concept outline

Daniel Śledziński

Adam Mickiewicz University, Poznań
fon@amu.edu.pl | ORCID: 0000-0003-0762-6869

Abstract: Daniel Śledziński, *A Multi-Layer Transcription Model – concept outline.* The Poznań Society for the Advancement of Arts and Sciences, PL ISSN 0079-4740, pp. 49-71

This paper discusses the assumptions of a Multi-Layer Transcription Model (hereinafter: MLTM). The solution presented is an advanced grapheme-to-phoneme (G2P) conversion method that can be implemented in technical applications, such as automatic speech recognition and synthesis systems. The features of MLTM also facilitate the application of text-to-transcription conversion in linguistic research. The model presented here is the basis for multi-step processing of the orthographic representation of words with those being transcribed gradually. The consecutive stages of the procedure include, among other things, identification of multi--character phonemes, voicing status change, and consonant clusters simplification. The multi-layer model described in this paper makes it possible to assign individual phonetic processes (for example assimilation), as well as other types of transformation, to particular layers. As a result, the set of rules becomes more transparent. Moreover, the rules related to any process can be modified independently of the rules connected with other forms of transformation, provided that the latter have been assigned to a different layer. These properties of the multi-layer transcription model in question provide crucial advantages for the solutions based on it, such as their flexibility and transparency. There are no assumptions in the model about the applicable number of layers, their functions, or the number of rules defined in each layer. A special mechanism used for the implementation of the MLTM concept enables projection of individual characters onto either a phonemic or a phonetic transcript (obtained after processing in the final layer of the MLTM-based system has been completed). The solution presented in this text has been implemented for the Polish language, however, it is not impossible to use the same model for other languages.

Keywords: G2P, grapheme-to-phoneme conversion, Polish language, text processing.

1. Introduction

The author's main inspiration for the creation and development of the concept of Multi-Layer Transcription Model was the book by Maria Steffen-Batogowa: *The Automatization of the Phonemic Transcription of Polish Orthographic Texts* (hereinafter: *Automatization...*) (Steffen-Batogowa 1975). The model discussed here can be considered a far-reaching modification of the concepts presented in that monograph. The solutions presented in *Automatization...* can be considered as a single-layer system. This means

that a given orthographic word is processed only once with the application of a specific set of rules. After this step, the final phonological transcription is obtained.

Work on the automatic conversion of an orthographic Polish text into a transcription record has been in progress since the early 1970s. In 1973 M.S.-B. published a paper: *The problem of automatic phonemic transcription of written Polish* (Steffen-Batogowa 1973). However, *Automatization...* was the first comprehensive study of the problem in question. It referred to all results of phonetic research available at that time. The methods presented, the way the rules were worded, as well as the rules themselves were used repeatedly in later works. In the article: *Implementation of the Phonematic Transcription Algorithm* (Wypych 1999), the author described his own implementation of the M.S.-B. transcription rules and included a review of works by other authors which date from the 1970s, 1980s and 1990s. The list begins with a publication entitled: *Program for the Odra-1204 machine for automatic phonematic transcription of Polish texts* (Warmus 1972). It concerned the first working computer program based on the then still incomplete rules proposed by M.S.-B. The system itself was so slow that it was impossible to use it in real time. This was mainly due to hardware limitations. Subsequent publications in this compilation introduce mainly technical improvements (Maksymienko & Bolc 1992; Chomyszyn 1986; Nowak 1995; Jassem 1996). In the solution described by Krzysztof Jassem in the paper: *A phonemic transcription – syllable division rule engine*, a rule module stored in a text file and a rule interpreter module were extracted. The accomplishments of *Automatization...* were also developed later by the author herself. In the paper: *Rules for the mutual conversion of the phonemic and phonetic transcriptions of the Polish texts* (Steffen-Batóg 1989-1990) the principles enabling the conversion of a phonological transcription into a phonetic transcription were discussed. Meanwhile, the paper: *An algorithm for phonetic transcription of orthographic texts in Polish* (Steffen-Batóg & Nowakowski 1992) discussed the principles for converting a text into a phonetic transcription. These papers (including the 1975 monograph) formed the basis for subsequent practical implementations. In 2003 an article was published entitled: *Implementation of grapheme-to-phoneme rules and extended SAMPA alphabet in Polish text-to--speech synthesis* (Demenko et al. 2003). It contains a description of selected linguistic problems connected with automatic transcription for the purposes of speech synthesis. The important issue of the transcription alphabet was addressed (a modification of the SAMPA standard was used). Another example of work based on M.S.-B. is presented in the article: *Ortfon2 - tool for orthographic to phonetic transcription* (Skurzok et al. 2015). The authors presented their own solutions, including a method for binary representation of segment strings. In the paper: *Algorithm and implementation of automatic phonemic transcription for Polish* (Kłosowski 2016) the author discussed the operation of *TransFon* software. This program was written in Python and its operation consists in sequential processing of single letters of a word using a set of rules in which context is taken into account. Another solution was implemented within the CLARIN platform (Common Language Resources & Technology Infrastructure) (Koržinek et al. 2016a; 2016b). The discussed items concern systems for converting orthographic notation into transcription, which are based on designed rules. However, algorithms, which aim at automatic creation of transcription rules, have also been developed for Polish. In the monograph:

Reconstruction of Phonematic Transcription Rules Based on a Learning Sample (Plucińs-ki 2002), the author described an original concept of a learning system that, based on empirical data, can build certain statistics. They facilitate adaptive selection and reconstruction of transcription rules. Another publication which discusses the process of automatic generation of transcription rules for Polish is: *The generation of letter-to-sound rules for grapheme-to-phoneme conversion* (Przybysz & Kasprzak 2013).

2. Linguistic nomenclature and transcription

2.1. Nomenclature of linguistic units

An important issue related to the multi-layer transcription model concerns the naming of linguistic units. It results from the unconventional essence of this solution. The first layer the MLTM-based project receives orthographic words understood as sequences of alphabet letters delimited on both sides by spaces or punctuation marks. A full-fledged transcription is achieved only after processing in the last layer. There is no assumption about how words should be transcribed during layer-by-layer processing. This means that any (conventional) way of transcription can be adopted. In this process, words gradually lose their orthographic features and take on transcriptional features. Thus, during processing, the notation of each word is indirect, and the units of which this notation consists are neither orthographic signs nor phonemes consistent with linguistic definitions. The solution to the problem at hand is to adopt a conventional nomenclature, but no specific nomenclature can be included in the model definition because the model does not make assumptions about the functions of each layer or the permissible units of word construction within those layers. These functions are determined at the time of designing a specific transcription system based on the multi-layer model. In the example design discussed in Section six, the units passed to the n-th layer (as word components) are conventionally called n-th degree segments.

2.2. Phonological inventory

"Polish phonological inventories are varied in terms of the phonemes distinguished due to differences concerning the criteria adopted, the methodology and the way of interpreting the phonemic status of certain sounds" (Lorenc & Ptaszkowska 2015: 230; Szpyra-Kozłowska 2007). Numerous proposals of such inventories can be found in the literature. Their comparative analysis was performed by E. Wolańska (Wolańska 2019). The phonological inventory proposed by Maria Steffen-Batogowa in *Automatizaton*.... is used in this publication. It was previously used in a study enabling the design of an example MLTM-based transcription system (see Section 6).

An important issue is the transcription alphabet used. This is a modified version of the SAMPA standard (Wells 1997). Table 1 shows the adopted phoneme inventory, which

was transcribed using the modified SAMPA alphabet (Demenko et al. 2003). The adopted inventory also includes the separate notation /rz/ (next to the notation /Z/), which actually denotes the same phoneme. This unconventional approach is due to the fact that the phone corresponding to the digraph *rz* is subject to lag assimilation, which is important for transformations related to modification of the voicing status. The notation /rz/ was distinguished for technical reasons and has to do with the functioning of an example transcription system based on the multi-layer model (see Section 6).

Table 1: Phoneme inventory adopted for this publication

No.	Phoneme SAMPA	Example SAMPA	Example word	No.	Phoneme SAMPA	Example SAMPA	Example word
1	/a/	/m.**a**.k/	*mak*	20	/rz/	/**rz**.e.k.a/	*rzeka*
2	/e/	/z.**e**.r.o/	*zero*	21	/Z/	/**Z**.a.b.a/	*żaba*
3	/i/	/n'.**i**.g.dz'.e/	*nigdzie*	22	/s'/	/**s'**.m.j.e.x/	*śmiech*
4	/o/	/s.**o**.v.a/	*sowa*	23	/z'/	/**z'**.a.r.n.o/	*ziarno*
5	/y/	/b.**y**.k/	*byk*	24	/x/	/k.u.**x**.n'.a/	*kuchnia*
6	/u/	/j.**u**.t.r.o/	*jutro*	25	/p/	/**p**.a.l.e.ts/	*palec*
7	/w/	/p.u.**w**.k.a/	*półka*	26	/b/	/**b**.u.d.a/	*buda*
8	/j/	/**j.**e.d.e.n/	*jeden*	27	/t/	/**t**.a.m.a/	*tama*
9	/w~/	/j.e.**w~**.z.y.k/	*język*	28	/d/	/**d.**o.m/	*dom*
10	/l/	/v.j.e.**l**.e/	*wiele*	29	/k/	/p.o.**k**.u.j/	*pokój*
11	/r/	/**r**.y.b.a/	*ryba*	30	/g/	/**g**.o.s'.ts'/	*gość*
12	/m/	/**m**.o.Z.e/	*morze*	31	/c/	/**c**.i.n.o/	*kino*
13	/n/	/m.o.**n**.e.t.a/	*moneta*	32	/J/	/**J**.i.t.a.r.a/	*gitara*
14	/n'/	/k.o.**n'**/	*koński*	33	/ts/	/**ts**.y.r.k/	*cyrk*
15	/f/	/**f**.u.t.r.o/	*futro*	34	/dz/	/**dz**.v.o.n.e.k/	*dzwonek*
16	/v/	/**v**.j.a.t.r/	*wiatr*	35	/tS/	/**tS**.a.s/	*czas*
17	/s/	/v.y.**s**.o.k.i/	*wysoki*	36	/dZ/	/**dZ**.u.m.a/	*dżuma*
18	/z/	/k.o.**z**.a/	*koza*	37	/ts'/	/k.o.**ts'**.o.w/	*kocioł*
19	/S/	/m.a.**S**.t/	*maszt*	38	/dz'/	/**dz'**.a.w.k.a/	*działka*

2.3. Primary transcription

Primary transcription (hereafter in this Section: PT) is a concept that was introduced for the purpose of the solutions discussed in this paper. It covers the elementary relationships between letters and phonemes. It can be said to be the original set of transcription rules. PT facilitates the use of the multi-layer model discussed, but is not a part of it (there is no obligation to use the primary transcription when creating MLTM-based designs).

The PT is conventional in nature and can be modified due to, among other things, the differences that exist between particular phonological inventories. An example is the biphonemic or monophonemic approach to so-called nasal vowels. Thus, the PT for the

letter ą and the letter ę may be diverse. Differences may also be related to the distributional properties of individual elements. Some orthographic dyads are digraphs only in specific contexts, so their inclusion or exclusion from PT may be due to conventional arrangements (see below).

The basic assumption associated with PT is the assignment of specific phonemes to individual characters or strings of orthographic characters, which may be specific graphemes. It is possible that a particular orthographic element in a particular context should be transcribed differently than assumed in PT. However, often the correct transcription coincides with it.

The second important assumption concerns the relationship between single letters and orthographic dyads. According to it, PT rules set for longer segments (in terms of the number of characters) have higher priority. For example, a rule for the dyad *sz* has a higher priority than a rule set for the letter *s* or for the letter *z*.

Table 2 and Table 3 present the conventional PT adopted for the purposes of this publication (information on single letters and orthographic dyads is presented separately). No PT was established for the letters *ę* and *ą* (due to contemporary numerous and varied phonological interpretations of these letters, which take into account various contexts).

Table 2: PT adopted for single letters

No.	Letter	PT	Voicing status	No.	Letter	PT	Voicing status
1	*a*	/a/	voiced	17	*m*	/m/	voiced
2	*ą*	–	voiced	18	*n*	/n/	voiced
3	*b*	/b/	voiced	19	*ń*	/n'/	voiced
4	*c*	/ts/	voiceless	20	*o*	/o/	voiced
5	*ć*	/ts'/	voiceless	21	*ó*	/u/	voiced
6	*d*	/d/	voiced	22	*p*	/p/	voiceless
7	*e*	/e/	voiced	23	*r*	/r/	voiced
8	*ę*	–	voiced	24	*s*	/s/	voiceless
9	*f*	/f/	voiceless	25	*ś*	/s'/	voiceless
10	*g*	/g/	voiced	26	*t*	/t/	voiceless
11	*h*	/x/	voiceless	27	*u*	/u/	voiced
12	*i*	/i/	voiced	28	*w*	/v/	voiced
13	*j*	/j/	voiced	29	*y*	/y/	voiced
14	*k*	/k/	voiceless	30	*z*	/z/	voiced
15	*l*	/l/	voiced	31	*ż*	/Z/	voiced
16	*ł*	/w/	voiced	32	*ź*	/z'/	voiced

Table 3: PT adopted for orthographic dyads

No.	Orthographic dyad	PT	Voicing status
1	*ch*	/x/	voiceless
2	*cz*	/tS/	voiceless
3	*dz*	/dz/	voiced
4	*dź*	/dz'/	voiced
5	*dż*	/dZ/	voiced
6	*rz*	/rz/	voiced
7	*sz*	/S/	voiceless

In addition to the dyads listed in Table 3, which are included in PT, the orthographic system in Polish includes several dyads and one triad, which mark a single phoneme only before a letter denoting a vowel (other than the letter *i*). These structures include the letter *i*, which is only a palatalization marker – in the context given, it is treated as a grapheme component. Because of this strong contextual consideration, they were not included in the PT. They are listed in Table 4.

Table 4: Strucures excluded from the adopted PT

No.	Orthographic notation	Transcription in vowel context	Transcription in consonant context	Voicing status
1	*ci*	/ts'/	/ts'.i/	voiceless
2	*dzi*	/dz'/	/dz'.i/	voiced
3	*ni*	/n'/	/n'.i/	voiced
4	*si*	/s'/	/s'.i/	voiceless
5	*zi*	/z'/	/z'.i/	voiced

The voicing is a phonetic (physical) feature of speech sounds. It has to do with the vibration of the vocal folds during the articulation process. Thus voicing is one of the basic characteristics of linguistic units associated with speech sounds, i.e. with phonemes and phones. In this study, the concept of voicing of characters is used in a conventional way. It denotes the voicing status of the phonemes that correspond to these units in the adopted PT. This information is included in the *Voicing status* column in Tables: 2, 3 and 4.

3. Basic terms connected with MLTM

There are several basic terms that are central to describing the construction and operation of MLTM. These include the term layer already mentioned. A layer in MLTM is a self-contained mechanism that has its own set of rules and is used to process strings of segments. A single layer is a transducer to which text data are passed and whose

output contains the processed text data. The processing is specified by the rules defined in the layer. It is important to note that different layers function independently of each other in the MLTM-based transcription system. It should be noted, however, that the rules belonging to a particular layer must be constructed in such a way that it is possible to process the transcription of words received from the previous layer. In other words: the rules associated with the particular layer must "understand" the way of writing words processed in the previous layer.

The next important elementary terms are segment and index. A segment is a section of the word being processed. The division of a given word into segments functions at any stage of its processing (within any layer). There are two types of segments in MLTM: single segments and multiple segments. Single segments can include basic units associated with a given notation system. In orthographic notation, these are letters, while in phonological transcription, the term refers to phonemes. Here are examples of single segments used in the project discussed in Section six:

– *a* (a segment including one letter);
– /a/ (a segment including one phoneme);
– /ts/ (a segment including one phoneme).

The notation of several units which, for specific reasons, constitute a distinct whole can be realized in multiple segments. The individual components of a multiple segment (e.g. phonemes) are combined using the character: & (in this way two or more segments can be written if necessary). Here are two examples of multiple segments:

– /e/&/n/ (a multiple segment including a sequence of phonemes);
– /o/&/w~/ (a multiple segment including a sequence of phonemes).

The index represents the segment number in the word. It should be noted that when a word is passed to the first layer, the number of indices is the same as the number of characters in that word increased by 2 due to the start and end markers. Table 5 illustrates this correlation on an exemplary word *wszędzie*.

Table 5**:** Initial state of assigning indices to segments in the word: *wszędzie*

Segment	#	w	s	z	ę	d	z	i	e	#
Index	0	1	2	3	4	5	6	7	8	9

The assignment of indices to each segment may change as the word is processed in subsequent layers. The important point is that the initial set of indices is identical to the set of indices in the word after it has been processed in the final layer of a given MLTM-based solution (in the example given, it is a set of ten elements).

4. The structure of MLTM rules

Each rule consists of two parts – the first module of the rule indicates the fragment of the word for which it will be applied (the processed word must contain this fragment

for the rule to be applied). The second module of the rule defines all modifications that apply to the segment of the word specified in the first part. In addition, the rule can be accompanied by an exclusions list or an inclusions list (see Section 5.5). It includes word initial, medial, or final orthographic strings to identify specific inflectional forms.

Both parts of the rule are divided into the same number of smaller analogous elements, which correspond to particular segments that constitute a fragment of the processed word. The MLTM rule construction scheme is presented in Table 6.

Table 6: WMT rule construction scheme

First module of the rule			Second module of the rule		
element 1	element 2	element n	element 1*	element 2*	element n*

In the first module of a rule belonging to the first layer of a particular solution based on MLTM, each element corresponds to exactly one letter. The meaning of the rule elements in subsequent layers results from the previously applied rules and the adopted way of transcribing the segments – it is not predefined. Individual elements are separated by dots. One element of the first module of the rule may refer to a particular word segment or it may be a notation defining a set of segments, for example:

– [A]−ą−ę (set A without letters: *ą* and *ę*);
– [A]+ł+j (set A increased by letters: *ł* and *j*);
– p+d+t+d+k+g (six-element letter set).

All set designations used must first be defined, for example the definition of set A could look like this: [A]={a,ą,e,ę,o,ó,u,i,y}. The following list contains examples of constructions of the first module of the rule:

– p.i.[A]−i (notation specific to the words: *pianino, opieszały, pieniądze*);
– r.ł.b (notation specific to the words: *dotarłby, nażarłby, wsparłbyś*);
– a.i (notation specific to the words: *zaistnieć, zaiskrzyć*).

These three examples are suitable for the first layer of the MLTM-based solution (orthographic words are passed to it). The third element in the first example includes a set of acceptable segments (here: a set of letters). The set symbols are used to specify the acceptable context for an element located in the first module of the rule, which is accompanied by a modification definition written at an analogous position in the second module of the same rule. For an element in the first module of a rule that includes a certain set of segments, no modification should be defined in the second module of that rule. An exception to this rule concerns the functioning of mapping lists. This mechanism makes it possible to define a set of transformations for different segment variants, while it is not known in advance which segment will be part of a particular word being processed.

The information presented below concerns the construction of the second module in MTML rules. The number of elements in both modules is identical, so that the individual elements in the second module of a rule relate to the analogous elements in the first module of that rule, which in turn relate to segments in the words being processed.

However, the nature and functions of the elements in the second module are different. Each such element may contain a notation associated with one of three possibilities:

- transcribing in unmodified form the word segment indicated by the analogous element located in the first module of the rule;
- transcribing in a modified form the word segment indicated by the analogous element located in the first module of the rule;
- using a special command concerning the operation to be performed on the word segment indicated by the analogous element located in the first module of the rule.

Detailed explanations of the listed options are provided in the next Section of this paper. The last option listed above requires additional explanation. The following commands can be used in the second rule module:

- !NC (no change) – this command can be used in any case, and especially for elements that were used in the first rule module only to specify context;
- !ML (move left) – as a result of this command a given segment in the processed word is removed, and the index that was assigned to this segment is appended to the index of the segment preceding the given segment (thus a single segment may be associated with several indices);
- !RM (remove) – this command removes the segment, but leaves the index in place – after applying this command a given index is assigned to a special segment: EMP (empty).

Operations on words processed in a given layer can be performed using, among other things, the mentioned commands. It should be noted that the rule is applied to the processed word provided that it contains a sequence of segments compliant with the structure defined in the first module of the rule. The discussed structure of the first module of the rule: p.i.[A]−i occurs, among others, in the words: *piasek* and *dopiekać*. In the word *piasek* the three elements of the first module of the rule coincide with the segments (letters) to which the indices: 1, 2, 3 are assigned (assuming that index 0 is assigned to the segment containing the character #). In the word *dopiekać* the mentioned structure of the first module of the rule coincides with the segments (letters), to which the indices: 3, 4, 5 are assigned. Thus, in the first case a sequence of segments (letters): *pia* (together with the assigned indices: 1, 2, 3) would be processed, while in the second case it would be a sequence: *pie* (together with the indices: 3, 4, 5).

The next Section of this paper discusses the operations (actions) that can be defined in the second module of the MLTM rule.

5. Operations carried out with the application of MLTM rules

5.1 Preserving the segment and index (no change)

The simplest operation is to preserve the segment unchanged and to leave the index that is assigned to that segment in the same position. If the element within the first module of the rule is a particular segment, then !NC command can be used to achieve the intended purpose. The other way is to rewrite the identical segment content in the

analogous element of the second rule module. Here are rule examples illustrating these two possibilities:

/b/./s/./t/./v/>/p/.!NC.!NC./f/;
/b/./s/./t/./v/>/p/./s/./t/./f/;

Both rules are suitable for processing the word *hrabstwo* written using the primary transcription (/x/./r/./a/./b/./s/./t/./v/./o/). Such a transcript would also be obtained after processing the orthographic word in the first layer of the example MLTM-based transcription system (see Section 6). The following rules would have to be used:

h>/x/;
r>/r/;
a>/a/;
b>/b/;
s>/s/;
t>/t/;
w>/v/;
o>/o/;

Thus, the two rules discussed are not appropriate for the first ('input') layer, to which only orthographic words are transferred. In the primary transcription, the voicing status assigned to orthographic characters is preserved (see Section 2.3). Table 7 and Table 8 illustrate the operation of the rules in question using word /x/./r/./a/./b/./s/./t/./v/./o/ processing as an example.

Table 7: Rule operating scheme: /b/./s/./t/./v/>/p/.!NC.!NC./f/;

Initial indexation	0	1	2	3	4	5	6	7	8	9
Word before the rule is applied	#	/x/	/r/	/a/	/b/	/s/	/t/	/v/	/o/	#
I module of the rule					/b/	/s/	/t/	/v/		
II module of the rule					/p/	!NC	!NC	/f/		
Word after the rule is applied	#	/x/	/r/	/a/	/p/	/s/	/t/	/f/	/o/	#
Final indexation	0	1	2	3	4	5	6	7	8	9

Table 8: Rule operating scheme: /b/./s/./t/./v/>/p/./s/./t/./f/;

Initial indexation	0	1	2	3	4	5	6	7	8	9
Word before the rule is applied	#	/x/	/r/	/a/	/b/	/s/	/t/	/v/	/o/	#
I module of the rule					/b/	/s/	/t/	/v/		
II module of the rule					/p/	/s/	/t/	/f/		
Word after the rule is applied	#	/x/	/r/	/a/	/p/	/s/	/t/	/f/	/o/	#
Final indexation	0	1	2	3	4	5	6	7	8	9

The difference between the two possibilities is significant. If a given segment is rewritten in the corresponding element of the second module of the rule in an unchanged form (as in the case of segments: /s/ and /t/ in the example given in Table 8), the possibility of its further modification will be blocked in the processed word. This means that when using other rules defined in this layer, this segment cannot be modified (this applies to rules used later). Using the !NC command also causes the segment to remain unchanged, but it will still be possible to replace that segment using other rules defined in the same layer.

The mechanism works in the following way: first, all rules defined in a given layer are checked for their possible use in the currently processed word (the checking starts from the longest rules). Rules appropriate for the word are continuously registered (during the checking) and at the same time, based on the structure of these rules, appropriate segments are blocked. Blocking particular segments means that they cannot be modified based on rules checked later. After checking all rules the registered changes are executed, then the segments are unlocked and the word is passed to the next layer for processing.

If the element of the first module of the rule is a set of segments, then in order to avoid changes, in the corresponding element of the second module only the !NC command can be used (because it is not known in advance which segment from this set will be part of the processed word). Here is an example of a rule containing this structure:

```
[WB2]./v/>!NC./f/;
```

In the example MLTM-based project discussed in Section 6, the WB2 set includes all second layer segments that are voiceless consonants proper. Table 9 illustrates how the rule works for the word /g/./w/./u/./p/./s/./t/./v/./o/. Such a notation would be obtained after processing the word *głupstwo* in the first layer. The following rules would be used for this purpose:

```
g>/g/;
ł>/w/;
u>/u/;
p>/p/;
s>/s/;
t>/t/;
w>/v/;
o>/o/;
```

By using the !NC command on any segment from the WB2 set (here it is: /t/), the possibility of modifying this element using other rules defined in the same layer is preserved.

Table 9: Rule operating scheme: [WB2]./v/>!NC./f/;

Initial indexation	0	1	2	3	4	5	6	7	8	9
Word before the rule is applied	#	/g/	/w/	/u/	/p/	/s/	/t/	/v/	/o/	#
I module of the rule							[WB2]	/v/		
II module of the rule							!NC	/f/		
Word after the rule is applied	#	/g/	/w/	/u/	/p/	/s/	/t/	/f/	/o/	#
Final indexation	0	1	2	3	4	5	6	7	8	9

5.2. Modification of a segment while maintaining its current index

The following operations can be distinguished to modify a segment in the word being processed that do not change the index assigned to that segment:

- replacing a single segment with another single segment;
- replacing a single segment with a multiple segment;
- replacing a multiple segment with another multiple segment;
- replacing a multiple segment with a single segment.

The operation of changing a single segment to another single segment involves replacing the string assigned to the index with another string. This type of operation was already covered in the description in Section 5.1. The /v/ (second level) segment was replaced with its voiceless counterpart, the /f/ segment.

An example rule for replacing a single segment with a multiple segment is as follows:

ę.t>/e/&/n/.!NC;

It causes the index originally assigned to segment *ę* be assigned to multiple segments including /e/ and /n/. The operation of this rule is illustrated by the example in Table 10.

Table 10**:** Rule operating scheme: ę.t>/e/&/n/.!NC;

Initial indexation	0	1	2	3	4	5	6
Word before the rule is applied	#	p	ę	t	l	a	#
I module of the rule			ę	t			
II module of the rule			/e/&/n/	!NC			
Word after the rule is applied	#	p	/e/&/n/	t	l	a	#
Final indexation	0	1	2	3	4	5	6

Another type of operation listed is the replacement of a multiple segment with another multiple segment. The operation could be performed when the phonological transcription of a multiple segment needs to be converted into a phonetic transcription. Such a situation

could occur in the case of designing an additional layer in a particular MLTM-based project, the purpose of which would be to convert the resulting phonological transcription into a phonetic transcription. It is difficult to find an analogy in Polish for the last listed operation mode, i.e. for replacing a multiple segment with a single segment.

5.3. Removing a segment (!RM command)

Using the !RM (remove) command removes the segment and leaves the index associated with that segment at the same position in the word being processed. After applying this command the index is assigned to a special EMP (empty) segment. If there is more than one index associated with the deleted segment, then all of those indices are assigned to the EMP segment. Using the !RM command modifies word indexation in the sense that it irrevocably removes the specified index (or indices) from further processing. The !RM command can be used for processing chunks of the word that are not realized.

Table 11 shows a processing scheme for the word /p/./o/./r/./ts/./j/./i/. Such a transcript would be obtained after processing the word *porcji* in the first layer of the example transcription system discussed in Section 6. The following rules would be used:

p>/p/;
o>/o/;
r>/r/;
c>/ts/;
j>/j/;
i>/i/;

No changes would be made to the second layer in this solution. In the third layer, a rule containing the command !RM would be used:

/ts/./j/./i/.#>!NC.!RM.!NC.!NC;

According to this rule, index number 5, which was assigned to the /j/ segment, is associated with the EMP segment.

Table 11: Rule operating scheme: /ts/./j/./i/.#>!NC.!RM.!NC.!NC;

Initial indexation	0	1	2	3	4	5	6	7
Word before the rule is applied	#	/p/	/o/	/r/	/ts/	/j/	/i/	#
I module of the rule					/ts/	/j/	/i/	#
II module of the rule					!NC	!RM	!NC	!NC
Word after the rule is applied	#	/p/	/o/	/r/	/ts/	EMP	/i/	#
Final indexation	0	1	2	3	4	5	6	7

5.4. Removing a segment while moving the index to the left (!ML command)

Using the !ML (move left) command results in a segment being deleted and the index associated with that segment being appended to the index of the segment preceding it. This command is primarily used to separate digraphs and trigraphs in orthographic words. The use of the !ML command can be demonstrated on the example of processing the word *szafa*. In this word, index 1 is initially assigned to the letter *s*, and index 2 is assigned to the letter *z* (taking into account the indexation of the # tag). The orthographic dyad of *sz* must eventually be transcribed as a single segment /S/. The two indices that were originally assigned to the segments *s* and *z* are assigned to it. This can be achieved using the following rule:

s.z>/S/.!ML;

In addition to using the !ML command, it is also necessary to ensure that the segment to which the two indices will be assigned is modified accordingly. If phonological notation was to be obtained as a result of processing orthographic words, the orthographic segment *s* should be replaced with the phoneme /S/. Table 12 presents the scheme of the above rule in relation to the word *szafa*. After such an operation indices 1 and 2 would be assigned to the phoneme /S/.

Table 12: Rule operating scheme: s.z>/S/.!ML;

Initial indexation	0	1	2	3	4	5	6
Word before the rule is applied	#	s	z	a	f	a	#
I module of the rule		s	z				
II module of the rule		/S/	!ML				
Word after the rule is applied	#	/S/	–	a	f	a	#
Final indexation	0	1,2	–	3	4	5	6

It is important to mention a few important limitations associated with the use of the !ML command. It cannot be used immediately after an element placed in the second module of a rule that contains the !RM command. It also cannot be used with respect to a segment to which index 1 is assigned, or with respect to a segment containing an EMP label or # character.

However, the !ML command can be used on several consecutive elements of the second rule module (located in its immediate vicinity). This can be used to separate trigraphs. An example rule containing a sequence of two !ML commands looks like this:

d.z.i.a>/dz'/.!ML.!ML.!NC;

Table 13 shows a diagram of how this rule works for the orthographic word *wydział*.

Table 13: Rule operating scheme: d.z.i.a>/dz'/.!ML.!ML.!NC;

Initial indexation	0	1	2	3	4	5	6	7	8
Word before the rule is applied	#	w	y	d	z	i	a	ł	#
I module of the rule				d	z	i	a		
II module of the rule				/dz'/	!ML	!ML	!NC		
Word after the rule is applied	#	w	y	/dz'/	–	–	a	ł	#
Final indexation	0	1	2	3,4,5	–	–	6	7	8

5.5. Exclusions lists and inclusions lists

The final topic discussed in this chapter involves exclusions lists and inclusion lists. These lists can be attached to rules. One list may be attached to a rule that contains strings of orthographic characters to identify specific inflectional forms. No full stops are used in the notation of these structures, but # character may be used to identify specific orthographic forms based on the fragments contained in the onset or rhyme of the word. The rule containing the exclusions list is structured as follows:

first_module_of a rule>second_module_of a rule>!EXC:exclusion_1,exclusion_2,...,exclusion_n;

Here is an example of the rule following the given scheme:

d.ż>/dZ/.!ML>!EXC:#nadż,#ponadż,#śródż,#odż,#współodż,#przedż,#podż,#ponadż,#nienadż,#nieponadż,#nieśródż,#nieodż,#niewspółodż,#nieprzedż,#niepodż,#nieponadż;

Individual items on the list are separated only by a comma (no space). The above rule contains a list of exclusions (or in other words, a list of exceptions). This means that any inflectional form that can be identified on the basis of any orthographic sequence included in the given list (in this example, these are onset sequences) is not covered by the rule – that is, the given rule cannot be used for it.

Inclusions lists make it possible to identify the set of inflectional forms for which a rule is to be used. It cannot be used for processing words outside the set that matches the specified inclusion list. The difference between writing an inclusions list and writing an exclusions list only concerns using the command: !INC instead of !EXC.

6. Example project based on MLTM

In this Section, an example project of an MLTM-based transcription system is briefly discussed. In this solution, the following conventional terminology is used: the words to be processed in the *n*-th layer (before processing) are composed of *n*-th degree segments, at the same time these segments belong to the *n*-th degree inventory. Thus, the words to

be processed in the second layer of this example solution are composed of second level segments that belong to the second level inventory. The first level segments (processed in the first layer of any MLTM-based project) are orthographic characters. The solution discussed here has three layers and their names are also conventional – they were determined by the project discussed here and are not defined in the MLTM.

6.1 Primary transcription layer

First layer in the presented model is supposed to separate characters, dyads, and triads in the orthographic notation that relate to phonological segments, i.e. that constitute graphemes. This initial transformation is done mainly on the basis of the primary transcription. The separation of character strings corresponding to individual phonemes requires taking into account the fact of the multi-functionality of the letter *i*. If it is only a sign of palatalization, it is treated as a component of a digraph or trigraph (see Section 2.3).

The transcript obtained after the word processing in the first layer does not contain modifications related to the assimilation of segmental voicing within consonant groups heterogeneous in terms of the status of so-called orthographic voicing status (see Section 2.3). It also does not contain simplifications within consonant groups and several other important modifications – rules related to the mentioned issues have been placed in higher layers.

Below are definitions of sets that are used only in rules belonging to the primary transcription layer (hence the number 1 in the names of these sets). The definition of the X1 set, which includes all letters, takes the following form:

[X1]={a, e, i, o, y, u, ó, ę, ą, ł, j, l, r, m, n, ń, f, w, s, z, ż, ś, ź, h, p, b, t, d, k, g, c, ć};

Set SA1 contains letters that represent vowels:

[SA1]={a, e, i, o, y, u, ó, ę, ą};

The set SP1 constitutes the difference between the sets: X1 and SA1, it contains letters that represent consonants and semivowels:

[SP1]={ł, j, l, r, m, n, ń, f, w, s, z, ż, ś, ź, h, p, b, t, d, k, g, c, ć};

The examples of rules incorporated into the primary transcription layer are discussed further. Here are single-element rules consistent with the adopted primary transcription:

a>/a/;
b>/b/;
c>/ts/;
h>/x/;
z>/z/;

d>/d/;
ż>/Z/;

The following rules apply to selected dyads included in the primary transcription. These are two-element rules, so they have higher priority than the one-elemnt rules given. The last rule contains a list of exclusions. The listed orthographic sequences make it possible to identify inflectional forms in which the orthographic dyad *dż* denotes two phonemes /d/ and /Z/ due to the morphological structure:

c.h>/x/.!ML;
c.z>/tS/.!ML;
d.ż>/dZ/.!ML>!EXC:#nadż,#ponadż,#śródż,#odż,#współodż,#przedż,#podż,#ponadż,#nienadż,#nieponadż,#nieśródż,#nieodż,#niewspółodż,#nieprzedż,#niepodż,#nieponadż;

The orthographic dyad *si* before a consonant indicates a sequence of phonemes /s'.i/ (e.g.: *sidła*, *siwy*). The following primary two-element rule can illustrate it:

s.i.[SP1]+#>/s'/./i/.!NC;

In the case of some words, the dyad *si* before a consonant indicates a phoneme sequence /s.i/ (e.g.: *silikat, sinus, pleksiglas*). These can be treated as exceptions. The rule created for the word *silikat* takes the following shape:

#.s.i.l.i.k>!NC./s/./i/.!NC.!NC.!NC;

It is a six-element rule, so it has a higher priority than the primary rule. This is the second method to accommodate exceptions and to exclude certain word forms from the operation of certain rules.

Table 14 covers the processing of the following example words in the primary transcription layer: *jeden, roztwór, marzłem, rzęsy, szczwacz, działo, jabłko*. Column two contains the transcript of the words before processing, while column three contains the processed words. Column four lists all the rules that are applied to each word. The longer rules have higher priority, so they are listed first. All rules used for the words: *jeden, roztwór* and *jabłko* are one-element ones. The dyad *rz* in the word *marzłem* indicates two phonemes. Six-element rule: #.m.a.r.z.ł>!NC.!NC.!NC./r/./z/.!NC; supersedes the two-element rule: r.z>/rz/.!ML;. In turn, this two-element rule is appropriate for the word *rzęsy* and it has a higher priority than the two one-element rules: r>/r/; and z>/z/;. Similarly, the two-element rules: s.z>/S/.!ML; and c.z>/tS/.!ML; have higher priority than the one-element rules for processing the letters *s, c, z.* They are appropriate for the word *szczwacz*. The word *działo* contains the triad *dzi*, which stands for the phoneme /dz'/. The corresponding rule: d.z.i.[SA1]−i>/dz'/.!ML.!ML.!NC; has higher priority than the rules: d>/d/; z>/z/; and i>/i/;.

Table 14: Processing of sample words in the primary transcription layer

No.	Word notation before applying rules	Word notation after applying rules	The applied rules
1	#.j.e.d.e.n.#	#./j/./e/./d/./e/./n/.#	j>/j/; e>/e/; d>/d/; e>/e/; n>/n/;
2	#.r.o.z.t.w.ó.r.#	#./r/./o/./z/./t/./v/./u/./r/.#	r>/r/; o>/o/; z>/z/; t>/t/; w>/v/; ó>/u/; r>/r/;
3	#.m.a.r.z.ł.e.m.#	#./m/./a/./r/./z/./w/./e/./m/.#	#.m.a.r.z.ł>!NC.!NC.!NC./r/./z/.!NC; m>/m/; a>/a/; ł>/w/; e>/e/; m>/m/;
4	#.r.z.ę.s.y.#	#./rz/./ę/./s/./y/.#	r.z>/rz/.!ML; ę>/ę/; s>/s/; y>/y/;
5	#.s.z.c.z.w.a.c.z.#	#./S/./tS/./v/./a/./tS/.#	s.z>/S/.!ML; c.z>/tS/.!ML; w>/v/; a>/a/;
6	#.d.z.i.a.ł.o.#	#./dz'/./a/./w/./o/.#	d.z.i.[SA1]–i>/dz'/.!ML.!ML.!NC; a>/a/; ł>/w/; o>/o/;
7	#.j.a.b.ł.k.o.#	#./j/./a/./b/./w/./k/./o/.#	j>/j/; a>/a/; b>/b/; ł>/w/; k>/k/; o>/o/;

6.2. The layer of voicing status

Second layer in the present project is supposed to modify the voicing status of segments in words processed in the primary transcription layer (Śledziński 2019; Ostaszewska & Tambor 2000; Rocławski 2001). The change in the voicing status involves only obstruents – each segment of the second degree that is a voiceless obstruent (except for /x/) has its voiced counterpart, which is characterized by the same place and manner of articulation.

After being processed in the first layer of the project, the voicing status of the segments reflects the implicit voicing of the orthographic characters. In the example words: *podkowa, władca, nadfiolet, fałdka, odpiorą*, /d/ segment would be used. However, after being processed in the second layer, the voicing status of the segments should reflect the actual articulatory features of speech. In the notation of the above words, the segment /t/ would be used.

The modification of the voicing status applies to the word initial, medial and final consonant groups, which in the original orthographic notation are heterogeneous in terms of the so-called orthographic voicing status (see Section 2.3). This modification consists in unifying the status of voicing within the groups. So, some voiced segments can be replaced with their voiceless counterparts or voiceless segments can be replaced by their voiced counterparts.

The following definitions of sets are used in the rules included in the voicing status layer:

[WD2]={/z/,/Z/,/z'/,/b/,/d/,/g/,/dz/,/dZ/,/dz'/};
[WB2]={/f/,/s/,/S/,/s'/,/x/,/p/,/t/,/k/,/ts/,/tS/,/ts'/};

The following two rules deal with desonorisation. The first rule deals with the left-side devoicing context of /v/ segment of the second degree. The second rule deals with the right-side devoicing context:

[WB2]./v/>!NC./f/;
/v/.[WB2]>/f/.!NC;

The next rule presented concerns sonorisation:

/k/.[WD2]>/g/.!NC;

Segments /v/ and /rz/ of the second degree were excluded from WD2. In the adopted primary transcription they correspond to the letter *w* and the dyad *rz*, thus they are subject to lag assimilation. The possible inclusion of these segments in the WD2 set would render the example rule: /ts'/.[WD2]>/dz'/.!NC; and similar rules inapplicable. It would cause an incorrect voiced onset transformation in the example words: *ćwierć*, *ćwiczyć*. In the discussed solution, this is also the reason for the separation of the notation /rz/ next to the notation /Z/ (see Section 2.2).

Table 15 covers the processing of example words in the voicing status layer. The second column includes the same words that were used in Table 14 (after being processed in the primary transcription layer). Column three in Table 15 includes the words after applying the rules belonging to the voicing status layer. The fourth column shows that modification of the voicing status is necessary for three words: *roztwór***,** *szczwacz* and *jabłko*. In the word *roztwór* two second degree segments (/z/ and /v/) are replaced with their voiceless counterparts thanks to the presence of the rules: /z/.[WB2]>/s/.!NC; [WB2]./v/>!NC./f/;. In the word *szczwacz*, only the /v/ segment of the second degree is present in the devoicing context. In the word *jabłko*, the modification includes the segment /b/ of the second degree. The semivowel /w/ is placed between it and the devoicing context.

Table 15: Processing sample words in the voicing status layer

No.	Word notation before applying rules	Word notation after applying rules	The applied rules
1	#./j/./e/./d/./e/./n/.#	#./j/./e/./d/./e/./n/.#	–
2	#./r/./o/./z/./t/./v/./u/./r/.#	#./r/./o/./s/./t/./f/./u/./r/.#	/z/.[WB2]>/s/.!NC; [WB2]./v/>!NC./f/;
3	#./m/./a/./r/./z/./w/./e/./m/.#	#./m/./a/./r/./z/./w/./e/./m/.#	–
4	#./rz/./ę/./s/./y/.#	#./rz/./ę/./s/./y/.#	–
5	#./S/./tS/./v/./a/./tS/.#	#./S/./tS/./f/./a/./tS/.#	[WB2]./v/>!NC./f/;
6	#./dz'/./a/./w/./o/.#	#./dz'/./a/./w/./o/.#	–
7	#./j/./a/./b/./w/./k/./o/.#	#./j/./a/./p/./w/./k/./o/.#	/b/.[SO2].[WB2]>/p/.!NC.!NC;

6.3. Modification layer

The function of the final (third) layer has to do with various modifications in transcription. Some of the rules included in this layer are related to obligatory transformations, for example, related to various interpretations of Polish nasal consonants (Lorenc 2016). In the discussed project, in the primary transcription layer, temporary notation is assigned to the letter *ę* and to the letter *ą*: /ę/ and /ą/. Rules in which all contexts are included are placed in the modification layer. The following list contains rules related to the transcription of the temporary notation /ę/ (based on Table 11 in *Automatizaton...*):

/ę/./l/+/w/>/e/.!NC;
/ę/.#>/e/.!NC;
/ę/./p/+/b/>/e/&/m/.!NC;
/ę/./t/+/d/+/k/+/g/>/e/&/n/.!NC;
/ę/./ts/+/dz/>/e/&/n/.!NC;
/ę/./ts'/+/dz'/>/e/&/n'/.!NC;
/ę/./s'/+/z'/+/f/+/v/+/s/+/z/+/S/+/Z/+/rz/+/x/>/e/&/w~/.!NC;

The last rule in this list is appropriate for the word *rzęsy* after it has been processed in the voicing status layer (Table 16). One more rule is included in this Table: /b/./w/./k/>!NC.!RM.!NC;. It concerns the reduction of the semivowel in the word *jabłko*. This problem is particularly evident in Polish, which contains numerous consonant clusters with structures not found in other languages (Dobrogowska 1984, 1990, 1992; Dukiewicz 1985; Dunaj 1985, 1986).

The WB3 and WD3 sets (not used in the examples) are similar to the WB2 and WD2 sets. The only difference is the inclusion of /v/ and /rz/ segments in the WD3 set (the modification layer does not contain rules related to modifying the voicing status).

Table 16: Processing sample words in the modification layer

No.	Word notation before applying rules	Word notation after applying rules	The applied rules
1	#./j/./e/./d/./e/./n/.#	#./j/./e/./d/./e/./n/.#	–
2	#./r/./o/./z/./t/./v/./u/./r/.#	#./r/./o/./s/./t/./f/./u/./r/.#	–
3	#./m/./a/./r/./z/./w/./e/./m/.#	#./m/./a/./r/./z/./w/./e/./m/.#	–
4	#./rz/./ę/./s/./y/.#	#./rz/./e/&/w~/./s/./y/.#	/ę/./s'/+/z'/+/f/+/v/+/s/+/z/+/S/+/Z/+/rz/+/x/>/e/&/w~/.!NC;
5	#./S/./tS/./v/./a/./tS/.#	#./S/./tS/./f/./a/./tS/.#	–
6	#./dz'/./a/./w/./o/.#	#./dz'/./a/./w/./o/.#	–
7	#./j/./a/./b/./w/./k/./o/.#	#./j/./a/./p/.EMP./k/./o/.#	/b/./w/./k/>!NC.!RM.!NC;

7. Conclusion

The publication presents the concept of the multi-layer model of text transcription. The author was inspired by Maria Steffen-Batogowa's book *The Automatization of the Phonemic Transcription of Polish Orthographic Texts*. This model can be useful in applications where there is a need to take into account various factors affecting the final transcription. These may be, for example, linguistic research or analysis performed for technical purposes.

The concept discussed here consists in the gradual processing of orthographic words. In successive stages of the algorithmic operation, the transcription of these words becomes more and more similar to the target transcription. In the terminology associated with the discussed model, the stages of the algorithmic operation are identified with successive layers. Each layer comprises an independent set of rules. The model does not dictate the functions assigned to each layer; instead, the model specifies the syntax, operation, and the principles of rule generation. On the example of the briefly discussed project, it has been presented that individual layers can be associated with specific text transcription issues.

The multi-layer model makes it possible to account for different phonological inventories and any phenomena or issues that affect transcription variability. This could be the synchronous or asynchronous realization of palatalization when pronouncing the orthographic sequences *gi*, *ki*, *ni* before a vowel (Retz 1989; Rocławski 1984; Sawicka & Grzybowski 1999). Related to this problem is the possibility of separating the phonemes: /c/ and /J/. Another problem concerns the pronunciation of the so-called Polish nasal vowels. In more recent studies, this pronunciation is biphonemic, but phonological interpretations concerning particular contexts vary. Another issue that can be taken into account in projects based on the MLTM is the influence of morphological structure (presence of juncture) on the biphonemic realization of some orthographic dyads (e.g.: *dż*, *dz*). This publication also refers to the phenomena of reduction or assimilation of voicing within consonant groups. All these phenomena may be the subject of thorough research. It should be emphasized that the results of such research can be easily incorporated into any transcription system project based on the MLTM. This is due to the fact that the rules have a simple structure and can be applied to individual issues. This is the main advantage and edge over the solution presented in *Automatization...*, which assumes single processing of each word. Therefore, some rules are complex and may address several transcription problems simultaneously. This makes the whole set inflexible (its modifications are prone to errors). On the other hand, the possibility of linking rules with individual issues (in solutions based on the MLTM) opens a wide range of possibilities to create alternative subsets of simple rules. Another significant advantage of the discussed solution is a special segment indexation mechanism, which enables projection of the initial orthographic transcription onto the resulting transcription (regardless of the number of layers used). Such linkage of the orthographic plane with the phonological or phonetic plane increases the possibilities of linguistic analyses, as it enables the precise transfer of information between these planes.

Due to its volume, this paper does not discuss all the issues and concepts associated with the MLTM. These include: mapping lists, which allow a significant reduction in the

number of rules; the concept of generative systems, which facilitate generation of different sets of transcription rules; probabilistic systems, which make it possible to include the probability calculus in the rules. The paper also does not address interword anticipatory assimilation, i.e., the possible modification of a word-final based on the structure of the initial of the next word. Also, the section on the exemplary design of the transcription system is limited to discussing a few examples and does not contain information on many transformations specific to Polish language. It is not impossible to use the discussed model for transcription of texts in other languages, in particular for Slavic languages (Sawicka 1988, 2007). These issues will be further developed in future publications.

References

Chomszyn, J. 1986. A phonemic transcription program for Polish. *International Journal of Man-Machine Studies* 25. 271-293.

Demenko, G. & Wypych, M. & Baranowska, E. 2003. Implementation of grapheme-to-phoneme rules and extended SAMPA alphabet in Polish text-to-speech synthesis. In Demenko, G. & Karpiński, M. (eds.), *Speech and language technology*, vol. 7, 79-96. Poznań: Polskie Towarzystwo Fonetyczne.

Dobrogowska, K. 1984. Śródgłosowe grupy spółgłosek w polskich tekstach popularnonaukowych. *Polonica* 10. 15-34.

Dobrogowska, K. 1990. Word internal consonant clusters in Polish artistic prose. *Studia Phonetica Posnaniensia* 2. 43-67.

Dobrogowska, K. 1992. Word initial and word final consonant clusters in Polish popular science text and in artistic prose. *Studia Phonetica Posnaniensia* 3. 47-121.

Dukiewicz, L. 1985. Nagłosowe grupy spółgłosek w polskich tekstach popularnonaukowych i prasowych. *Studia Gramatyczne* 6. 17-34.

Dunaj, B. 1985. *Grupy spółgłoskowe współczesnej polszczyzny mówionej (w języku mieszkańców Krakowa)*. Warszawa – Kraków: Państwowe Wydawnictwo Naukowe PWN.

Dunaj, B. 1986. Wygłosowe grupy spółgłoskowe współczesnej polszczyzny mówionej. *Zeszyty Naukowe Uniwersytetu Jagielońskiego: Prace Językoznawcze* 82. 103-117.

Jassem, K. 1996. A phonemic transcription – syllable division rule engine. In *Proceedings of ONOMASTICA–COPERNICUS Research Colloquium Edinburgh* (pages unknown). Edinburgh: Centre for Communication Interface Research, University of Edinburgh.

Kłosowski, P. 2016. Algorithm and implementation of automatic phonemic transcription for Polish. In *Proceedings of 20th IEEE International Conference Signal Processing Algorithms, Architectures, Arrangements and Applications*, 298-303. Poznań: University of Technology.

Koržinek, D. & Brocki, Ł. & Marasek, K. 2016a. Polish grapheme-to-phoneme tool and service. (http://hdl.handle.net/11321/295) (Accessed 2022-10-02)

Koržinek, D. & Marasek, K. & Brocki, Ł. 2016b. Polish read speech corpus for speech tools and services. In *Proceedings, CLARIN Annual Conference 2016, Aix-en-Provence: Common Language Resources and Technology Infrastructure*. (https://www.clarin.eu/sites/default/files/korzinek-etal-CLARIN2016_paper_20.pdf) (Accessed 2022-10-02)

Lorenc, A. 2016. *Wymowa normatywna polskich samogłosek nosowych i spółgłoski bocznej.* Warszawa: Dom Wydawniczy ELIPSA.

Lorenc, A. & Ptaszkowska, A. 2015. Programowanie języka dziecka z uszkodzeniem słuchu z zastosowaniem metody audytywno-werbalnej: Studium przypadku. *Logopedia Silesiana* 4. 229-247.

Maksymienko, M. & Bolc, L.1981. *Komputerowy system przetwarzania tekstów fonematycznych.* Warszawa: Wydawnictwa Uniwersytetu Warszawskiego.

Nowak, I. 1995. Transkrypcja fonematyczna tekstów polskich. In Pogonowski, J. (ed.), *Eufonia i logos*. Poznań. Wydawnictwo Naukowe UAM.

Ostaszewska, D. & Tambor, J. 2000. *Fonetyka i fonologia współczesnego języka polskiego.* Warszawa: Wydawnictwo Naukowe PWN.

Pluciński, A. 1992. *Rekonstrukcja reguł transkrypcji fonematycznej na podstawie próby uczącej.* Poznań: Wydawnictwo Naukowe UAM.

Przybysz, P. & Kasprzak, W. 2013. The generation of letter-to-sound rules for grapheme-to-phoneme conversion. In *6th International Conference on Human System Interactions (HSI) (Sopot, Poland)*, 292-297. IEEE (Institute of Electrical and Electronics Engineers). (https://ieeexplore.ieee.org/document/6577838) (Accessed 2022-10-02)

Retz, R.I. 1989. Zanik korelacji palatalności we współczesnej polszczyźnie ogólnej. *Poradnik Językowy.* 1. 20-32; 2. 90-99; 3. 160-168.

Rocławski, B. 1984. *Palatalność: Teoria i praktyka.* Gdańsk: Uniwersytet Gdański. Wydawnictwo.

Rocławski, B. 2001. *Podstawy wiedzy o języku polskim dla glottodydaktyków, pedagogów, psychologów i logopedów*. Gdańsk: Glottispol.

Sawicka, I. 1988. *Fonologia konfrontatywna polsko-serbsko-chorwacka.* Wrocław: Zakład Narodowy im. Ossolińskich.

Sawicka, I. (ed.). 2007. *Komparacja współczesnych języków słowiańskich*. Vol. 2. *Fonetyka i fonologia*. Opole: Wydawnictwo Uniwersytetu Opolskiego.

Sawicka, I. & Grzybowski, S. 1999. *Studia z palatalności w językach słowiańskich*. Vol. 1. Toruń: Wydawnictwo Naukowe UMK.

Skurzok, D. & Ziółko, B. & Ziółko, M. 2015. Ortfon2 – tool for orthographic to phonetic transcription. In *7th Language & Technology Conference*, 115-119. Poznań: Adam Mickiewicz University. (http://ltc.amu.edu.pl/a2015/book/papers/SPEECH1-2.pdf) (Accessed 2022-10-02)

Śledziński, D. 2019. *Asymilacja dźwięczności w polskich grupach spółgłoskowych.* Poznań: Bogucki Wydawnictwo Naukowe.

Steffen-Batóg, M. 1989-1990. Rules for the mutual conversion of the phonemic and phonetic transcriptions of the Polish texts. *Lingua Posnaniensis* 32-33. 211-223.

Steffen-Batóg, M. & Nowakowski, P. 1992. An algorithm for phonetic transcription of orthographic texts in Polish. *Studia Phonetica Posnaniensia* 3. 135-183.

Steffen-Batogowa, M. 1973. The problem of automatic phonemic transcription of written Polish. *Biuletyn Fonograficzny* XIV. 75-86.

SteffenBatogowa, M. 1975. *Automatyzacja transkrypcji fonematycznej tekstów polskich.* Warszawa: Państwowe Wydawnictwo Naukowe.

Szpyra-Kozłowska, J. 2007. Inwentarze fonemów języka polskiego i ich konsekwencje. *Logopedia* 31. 7-26.

Warmus, M. 1972. Program na maszynę Odra-1204 dla automatycznej transkrypcji fonematycznej tekstów języka polskiego. *Prace CO PAN* 66. 1-22.

Wells, J. 1997. SAMPA computer readable phonetic alphabet. In Gibbon, D. & Moore, R., & Winski, R. (eds.), *Handbook of standards and resources for spoken language systems*, part IV, section B. Berlin – New York: Mouton de Gruyter.

Wolańska, E. 2019. *System grafematyczny współczesnej polszczyzny na tle innych systemów pisma.* Warszawa: Dom Wydawniczy ELIPSA.

Wypych. M. 1999. Implementacja algorytmu transkrypcji fonematycznej. In Jassem, W. & Basztura, Cz., & Demenko, G., & Jassem, K. (eds.), *Speech and language technology*, vol. 3, 89-105. Poznań: Polskie Towarzystwo Fonetyczne.