1. INTRODUCTION

This article presents a new type of synchronic morphological description of Old Church Slavonic (OCS). It is based on a computational two-level model created by Kimmo Koskenniemi (1983). The main characteristics of this model are as follows:

1. A language is described morphologically by means of two-level rules which define the permissible correspondences between the lexical and surface representation of a word form. No intermediate representations like those used in generative phonology are needed.
2. The model does not distinguish between production and recognition: the rules are not directional.
3. The model is implemented as a computer program which, given a morphological description of a language, can both produce (generate) and recognize (analyse) word forms of that language.

The model is described in detail by Koskenniemi (1983); Karttunen (1983) can also be consulted. Koskenniemi has written a computationally effective implementation of his model in the PASCAL programming language; there is also an INTERLISP implementation (Gajek et al. 1983). Koskenniemi’s book contains a full morphological analysis of Finnish in terms of two-level rules. Similar descriptions of English, French, Rumanian and Japanese can be found in Vol. 22 (1983) of the Texas Linguistic Forum. Analyses of Swedish, Modern Greek and Classical Arabic are being prepared at the University of Helsinki.

Below I present my two-level description of OCS and demonstrate how the automatic analysis of word forms works (their automatic generation is a simpler job and will not be discussed). Notational conventions of two-level morphology and its other general features will be introduced in the course of the presentation.

What use can be made of such a computer model of OCS? First, it is a linguistic description, just like any other morphological (e.g. generative)
description. But it is easier to test because its correctness can be checked by executing the program. Secondly, the model can be used for philological purposes like tagging and lemmatizing\(^1\) computerized OCS corpora (once such corpora have been established). Thirdly, OCS is a suitable testing ground for the morphological rules needed in describing all modern Slavonic languages, whose automatic analysis would have numerous scientific and commercial applications.

2. THE PRESENT STATE OF THE OCS MODEL

The test dictionary and rule component of the OCS description have been chosen to permit experimentation with (i) all the main declensional and conjugational types and (ii) all the productive morphophonological rules. The full paradigms of different types of nouns and verbs and the definite and indefinite declension of the adjective have been implemented. The comparison of adjectives has not been included here, but this should not be more difficult to implement than the participle formation of verbs, which already works correctly. Numerals have not yet been incorporated into the system. As regards pronouns, the system now recognizes the main declensional type *t\(\text{ь}\)*, *togo* and its soft counterpart *moi*, *mojego*.

All the lexemes of the test dictionary are listed below to show the inflectional types the present system can handle:

- biti, bog\(\text{ь}\), brati, c\(\text{ё}\)sar\(\text{ь}\), c\(\text{ё}\)ky, \(\text{ч}\)jati, \(\text{ч}\)isti, \(\text{ч}\)uti, chvaliti, d\(\text{ё}\)lati, d\(\text{ё}\)v\(\text{ь}\), d\(\text{ё}\)olv\(\text{ь}\), du\(\text{s}\)a, d\(\text{ё}\)\(\text{ш}\)ti, dvign\(\text{ь}\)ti, glagolati, glagol\(\text{ь}\), gra\(\text{ž}\)danin\(\text{ь}\), greti, g\(\text{n}\)nati, ig\(\text{о}\), im\(\text{ё}\), jed\(\text{ь}\), j\(\text{ё}\)ti, kamy, k\('\text{ё}\)sar\(\text{ь}\), kon\(\text{ь}\), k\(\text{о}\)st\(\text{ь}\), kopati, krai, k\(\text{в}\)\(\text{ь}\), kryti, k\(\text{н}\)\(\text{ё}\)dz\(\text{ь}\), k\('\text{n}\)\(\text{i}\)ga, l\(\text{ё}\)to, l\(\text{г}\)gati, mati, min\(\text{ь}\)ti, moi, mo\(\text{ш}\)ti, mo\(\text{з}\)\(\text{ь}\), mysliti, mysl\(\text{ь}\), nare\(\text{ш}\)ti, naricati, nastaviti, nesti, noga, nov\(\text{ь}\), o\(\text{ть}\)\(\text{ь}\), otro\(\text{ч}\)\(\text{ь}\), o\(\text{же}\), padati, pasti, p\(\text{ё}\)ti, pluti, pol\('\text{е}\), p\(\text{от}\)\(\text{ь}\), pr\(\text{ё}\)biti, rab\(\text{ь}\), re\(\text{ш}\)ti, r\(\text{k}\)a, s\(\text{ё}\)ti, s\(\text{ё}t\)i (*and* s\(\text{ё}\)jati), slovo, sly\(\text{ш}\)ati, sm\(\text{ь}\)jati, strada, stradati, syn\(\text{ь}\), t\(\text{ь}\), te\(\text{ш}\)ti, teti, t\(\text{ё}\)lo, tok\(\text{ь}\), t\(\text{ь}\)\(\text{ш}\)\(\text{ь}\), vel\(\text{ь}\)ti, velii, vesti (*ved\(\text{о}\)*), v\(\text{ё}\)rovati, vid\(\text{ь}\)ti, v\(\text{ы}v\(\text{в}\)\(\text{ь}\)\(\text{ь}\)ti, vy\(\text{ы}\)\(\text{шн}\)\(\text{ь}\), zeml\('\text{а}\), zm\(\text{i}\)ja, znamenije, znati, z\(\text{в}\)vati, \(\text{ж}\)\(\text{ена}\), \(\text{ж}\)\(\text{ё}\)ti, \(\text{ж}\)\(\text{р}\)\(\text{ё}\)ti.

---

\(^1\) Morphological *tagging* means providing the words of a corpus with "tags" indicating their morphological and morphosyntactic analysis, which enable the user to search automatically, say, for all feminine accusative forms. *Lemmatizing* assigns each word form a lemma, i.e. its dictionary headword (or other conventional name of the lexeme), and is useful for compiling concordances, etc.
The system is a kind of "computerized Leskien"—i.e. it knows the standard paradigms and morphophonological alternations found in classical Handbücher like Leskien's (1969) and Vailant's (1964). The surface representations of the words are transliterations of their Cyrillic orthography, with certain graphemic variations taken into account. As not all computers can yet handle diacritics easily, I have substituted capital letters for the missing characters: A=č, E=ę, O=о, I=Ъ, U=Ъ; S=§, Z=ž, C=ć. For ch I have used x (h would have done equally well since it has no other uses), and dz is represented by D. The palatalization sign ' (' after n, l, r, ch, k, and g is marked by '. In the present article this transliteration is used only when parts of the computer files are quoted directly. Otherwise I have employed the traditional system with diacritics. For ch and dz, however, I have had to use x and D because digraphs would have spoiled the vertical alignment of the two-level rules.

The system knows that j can appear optionally after all palatal consonants (including §, ž, c, etc.), and can even replace the palatalization sign. It also knows that j is only permissible before certain vowels; combinations such as *jo are impossible.

An interactive session with the computer (Burroughs B7800) is presented in the Appendix as an example. It shows that the system can cope with the homography of forms of different lexemes, and with syncretism within paradigms.

The morphological system that makes the automatic analysis of OCS word forms possible consists of two files, a dictionary file and an automata file, the latter simulating two-level rules. Technically, these are data files: the description is not part of the program code, which is why the two-level program can be used to describe any language and differs essentially from language-specific systems like that presented in Sågvall (1973). The two data files are discussed below consecutively.

3. THE DICTIONARY

In the two-level descriptions, the dictionary consists of a number of lexicons and definitions of continuation classes. The longest lexicon is the Root lexicon containing the roots of all the lexemes. The other lexicons comprise (a) stem alternation patterns, and (b) suffix sets. In every lexicon, the format of the lexical entries is

(lexical representation) (continuation class) "(meaning)"
For example, the Root lexicon includes the following entry:

\texttt{slov o/es "word N \langle NE\rangle";}

which means that \textit{slov} is a neuter noun meaning ‘word’ and belonging to the continuation class o/es. This continuation class is defined as a pointer to a lexicon called o/es (it happens to have the same name as the class but this is not indispensable). The lexicon o/es contains the following entries:

\begin{verbatim}
o § "SG NOM";
o § "SG ACC";
es /CS:ne ""
\end{verbatim}

All s stems refer to this same lexicon, so their typical alternation pattern need be presented only once. The sign § denotes the end of the word, i.e. the continuation class with no pointers to other lexicons. The string es has no meaning of its own but its continuation class is /CS:ne (‘neuter consonant stem’), which is defined as a pointer to the lexicons CS:sg (singular consonant-stem endings common to all three genders), CS:pl (the corresponding plural and dual endings), and CS:ne (endings proper to the neuter consonant stems only). For example, the lexicon CS:sg appears as follows:

\begin{verbatim}
+e § "SG GEN";
+i § "SG DAT"
\end{verbatim}

If a stem is not subject to alternations other than those specified by the two-level rules (see below), the continuation class of the root points directly to the suffix lexicons, e.g. the o-stem \textit{rab}:

\begin{verbatim}
rab /o:ma "slave N \langle MA\rangle";
\end{verbatim}

Thus, the dictionary consists of a complicated network of lexicons, held together by the continuation class definitions. It is easy to see how this system can be used to describe, say, the different declensions of nouns, so I will not go into greater detail. Some interesting points can be noted, however.

To start with, the dictionary does not need to know the distinction between “hard” and “soft” endings, so fewer paradigms are required than in OCS Handbücher. The surface string \textit{kon’b} ‘horse’ (nom. and acc. sg., and gen. pl.) is paired with the lexical string \textit{kon’+b}. The
relation \( \leftrightarrow \) is handled by the "vowel-fronting" two-level rule (rule (5a) below).

Adjectives make use of the existing lexicons of nominal and pronominal endings. The continuation class /Adj points to the lexicon Adj, which looks like this:

\[
\begin{align*}
0 & \quad /o:ma \quad "\text{INDEF } \langle \text{MA} \rangle"; \\
0 & \quad /a \quad "\text{INDEF } \langle \text{FE} \rangle"; \\
0 & \quad /o:ne \quad "\text{INDEF } \langle \text{NE} \rangle"; \\
+\%Uj & \quad /Pron \quad "\text{DEF}"; \\
+\%yj & \quad /Pron \quad "\text{DEF}"; \\
+\%Yj & \quad /Pron \quad "\text{DEF}"; \\
+\%aj & \quad /Pron \quad "\text{DEF}"; \\
+\%uj & \quad /Pron \quad "\text{DEF}"; \\
+\%Åj & \quad /Pron \quad "\text{DEF}"; \\
+\%oj & \quad /Pron \quad "\text{DEF}"; \\
+\%ij & \quad /Pron \quad "\text{DEF}"; \\
+\% & \quad /Pron \quad "\text{DEF}"
\end{align*}
\]

The first three entries add no characters to the lexical representation; they merely refer to the suffix lexicons of o- and a-stems. The other ten entries represent case ending remnants which (together with the old pronominal stem \( \text{j-} \)) are to be found in the definite adjective forms in front of the pronominal endings (to which /Pron points). (The morphophonemes \( \text{Å} \) and \( \text{Y} \) are explained in section 4.) The sign "\( \% \)" is a morphological feature triggering a two-level rule according to which every case-ending remnant must be followed by the correct pronominal endings. The string \(+\%Uj\) is paired only with pronominal endings which have the feature "1", \(+\%yj\) only with those marked "2" and so on. The alternative would have been to assign a separate continuation class to each of the ten case-ending remnants and divide the now unitary Pron lexicon into corresponding smaller lexicons. The present treatment is preferred because the pronominal declension by itself forms a neat paradigm; it is the definite adjective declension that is messy in OCS (corresponding synchronically to the diachronic flux).

In the case of verbs, it is interesting to note that there are no really separate conjugations in the same way as there are separate declensions of nouns. All the verbs share the same set of suffix lexicons. The only exceptions—not counting the few athematic verbs—are the endings of
the first person singular\(^2\) and third person plural of the present tense, and the formatives of the active and passive present participle, which distinguish Leskien's fourth class (xvaliti, vidětī) from the other three classes. The difference between Leskien's classes can be accounted for in terms of stem alternation patterns; the thematic vowel is treated as part of the stem.

This approach to verb conjugation has reduced the number of suffix lexicons but it has also destroyed the traditional division into classes. The uj/uje/ova alternation of věrovati 'to believe' is now described in a different lexicon than the simpler alternation /e/a of glagolati 'to speak', though the věrovati pattern is of course a special case of the latter pattern. This generalization could easily be captured by writing a two-level rule governing the ov/u alternation (the same rule could then be used also in the case of pluti/plovq 'to swim'). By adding rules of this kind we could further reduce the number of verb stem alternation patterns, but natural and productive morphophonological rules should be clearly distinguished from historical processes (see the discussion in section 5).

4. TWO-LEVEL RULES

The purpose of the two-level rules is to define the permissible correspondences between the lexical and surface representation of each word-form. They differ from rules of generative phonology in several respects. The nearest two-level equivalent of the generative rule

\[(i) \quad A \rightarrow B / C \_ D\]

would be

\[(ii) \quad \frac{A}{B} \iff C \_ D\]

to be read as "character A at the lexical level is realized as character B at the surface level if, and only if, it occurs in the context C\_D". Note that B is necessarily a surface character; there can be no rules after this rule that would further change B into some other character. Generative

\(^2\) The ending is, of course, the same -φ in all the thematic classes. But in the fourth class it is represented in the dictionary as ıJ+φ, where ıJ is the old j changing the previous consonant (cf. the viždO = viždq example in the Appendix). In the imperfect tense, the ıJ can be taken into account in forming the stem because it appears before all the endings of the imperfect lexicon.

Scando-Slavica, Tomus 30 1984
rules are ordered in series, while two-level rules operate in parallel: each word-form must satisfy all of them simultaneously (provided the context specifications are met). Two-level rules also give wider possibilities for context specification. Actually, rule (ii) is merely an abbreviation of

\[
\begin{align*}
(iii)\quad & A \leftrightarrow C \quad D \\
& B \leftrightarrow C \quad \_ \quad D
\end{align*}
\]

and nothing prevents us from writing rules like

\[
(iv)\quad A \leftrightarrow C \\
B \leftrightarrow G \quad \_ 
\]

i.e. "A corresponds to B if, and only if, it follows a C that corresponds to G (according to some other rule)". Optionality of a surface realization is marked by making the context condition a necessary but not sufficient condition, i.e. by using the \(\Rightarrow\) ("only if") operator instead of \(\leftrightarrow\).

For example, to describe the correspondence \(ved+ti \leftrightarrow vesti\) 'to take (someone somewhere)', we need the following rules:

\[
(v)\quad d \quad s \quad \_ \quad +t \\
(vi)\quad + \quad \_ \quad \_ 
\]

The latter rule deletes the morpheme boundary in all contexts. (For the actual formulation of these rules, see (10a) and (4b) below.) Taken together, (v) and (vi) allow the following pairing of lexical and surface strings:

\[
\begin{align*}
ved + ti \\
ves \quad \emptyset \quad ti
\end{align*}
\]

which is just what we want.

The two-level program does not yet understand two-level rules, as such, but only special matrices representing finite-state automata that simulate the rules. It is psycholinguistically interesting to note that the rules are equivalent to such computationally simple and effective devices (cf. Koskenniemi 1983: 134ff. on processability). The exact form of the automata is not of great linguistic interest because all the rule automata could in principle be merged into one big automaton representing all the rules, whereas the rules themselves cannot coalesce in the same way. Below, I will not present the contents of the automata file, but I will describe the rules that the automata encode.
To start with, the alphabet to be used in the rules must be specified. It contains all the surface characters from a to ż, including the ’ sign, and morphophonemes, junctures, and feature signs. There are six morphophonemes:

- Y: suffixal y whose soft counterpart is ę
- Â: “Jat-2”
- H: ̀b which is in an Ablaut relation to i
- &: the ending of the aorist 2nd and 3rd p. sg.
- ų: the consonant in the sigmatic aorist
- J: underlying “j” which is never realized as such but alters the previous consonant

There are two morphological junctures: the ordinary morpheme boundary “+”, and the juncture “/” used in the past participle passive as its distinctive mark (i.e. to encode both the juncture and a morphological feature). A dozen more signs, such as “!”, “(” and “””, are used as morphological features in the lexical representations.

The context specifications of the rules have been simplified by defining subsets of the alphabet. I have used the following:

- =: all the characters of the alphabet
- K: consonants (b c č ... z ž ’ ų)
- V: vowels (a e ... y Y Â H &)
- P: palatal consonants (’ j J ų ż č c D)
- W: vowels before which a j can be written (a e ę q u)
- M: vowels triggering the first palatalization (b e ę ě &)
- N: vowels triggering the second palatalization (i Â)³
- B: labials (p b v m)
- Q: velars (k g x)
- L: dental sonorants (l n r)

These sets can only refer to concrete correspondence pairs established by other rules, which is why a context condition like

```
t  t
= — =
```

does not mean “between two t’s that can each be realized as any character” but “between two t’s that are realized as defined by other rules”. In most cases this actually means

³ In the endings, i is always an i₂, so it can be placed in set N.

*Scando-Slavica, Tomus 30 1984*
\[ t \rightarrow t \left( \text{i.e. } \begin{array}{c} t \\ t \end{array} \right) \]

but context specifications are usually not made so redundant (though this is also a matter of style).

The rules are as follows:

(1) **Surface characters**

(1a) \[ a \; b \; \ldots ; \; \ddot{z} \; \Rightarrow \; - \]

This rather trivial rule defines the characters that can surface as such in any context, but they must be realized as something else if another rule so specifies; that is why the context condition is only necessary ("\( \Rightarrow \)") but not sufficient ("\( \Leftarrow \)"). I have used semicolons to merge rules in an obvious fashion.

(1b) \[ ' \; \Rightarrow \; [L \; | \; Q] \; - \]

' can only appear after l, n, r, k, g, or x. The notation \([\ldots] \) is used to mark alternative parts of the context.

(2) **Graphemic options for soft consonants**

(2a) \[ ' \; j \; \Rightarrow \; - \]

According to the rule, ' can always be realized as j (but note that rule (3a) may prevent this in specific cases).

(2b) \[ ' \; j \; \Rightarrow \; \begin{array}{c} == \\ \ddot{s} \; t \\ \ddot{z} \; d \\ P \; z \end{array} \; - \]

The morpheme boundary can correspond to j if it follows anything that is realized on the surface as \( \ddot{s}t \), \( \ddot{z}d \) or a palatal consonant, including z representing D. This rule accounts for the graphemic variation m\( \ddot{z}a \)/m\( \ddot{z}ja \) 'man (gen. sg., nom. dl.)' and similar cases. As the two-level model does not allow for absolute epenthesis (\( \emptyset \) is only permitted on the surface level), the epenthetic j must technically be taken from the boundary mark.

(3) **The graphemics of j**

(3a) \[ j \; \Rightarrow \; - \; W \]
An example: in the locative *kraj+Ā* ‘edge’, Ā becomes i according to rule (5b); but in this case the context condition in (3b) is met, j disappears, and we obtain the surface form *krai*.

Logically, rules (3b) and (3d) only make sense if their context conditions are understood to be in a disjunctive relation. This is a general convention of the two-level model (Koskenniemi 1983: 37–38).

**(4) Unmarked realizations**

\[(4a) \begin{array}{cccccccc}
D & D & Y & \text{Ā} & H & \& & \breve{S} & J \\
D & z & y & \breve{e} & b & e & s & \emptyset
\end{array} \Rightarrow -
\]

This rule defines the “context-free” realizations of the morphophonemes and allows D to surface either as D or as z.

\[(4b) + / \emptyset; \emptyset; \emptyset; \ldots; \emptyset; \emptyset \Rightarrow -
\]

Junctures and feature signs are always realized as a zero string. The only exceptions are cases in which they are used as placeholders for ephentheses, as in rule (2b) above.

**(5) Vowel fronting**

\[(5a) \begin{array}{cccccccc}
y & Y & \breve{b} & o & \breve{e} \\
i & \breve{e} & b & e & a
\end{array} \Leftrightarrow \begin{array}{c|c|c|}
\text{---} & \text{---} & \text{---} \\
\breve{S} & T & \breve{Z} & D & j
\end{array} -
\]

This rule produces the endings of “soft” declensions, among other things. Note how (5a) and (7a) interact in the imperfect *tečaaxь* ‘I was running’:

\[
t e \ oxed{k + \breve{c}} \ a \ x \ \breve{b} \\
t e \ \emptyset \ \breve{c} \ \emptyset \ a \ x \ \breve{b}
\]

By a general convention of the two-level model (Koskenniemi 1983: 40),

\[\text{---}
\]

\[4\text{ For the } \breve{c} \Leftrightarrow a \text{ correspondence the term “fronting” is not justifiable phonetically, but it is included in the fronting rule owing to its identical context conditions.}\
\]

*Scando-Slavica, Tomus 30 1984*
intervening junctures and feature signs—"+" in this example—need not be mentioned in the context conditions of rules if they are optional.

\[(5\, b) \quad \hat{A} \leftrightarrow \hat{i} \iff \begin{pmatrix} \text{t} & \text{d} \\ \hat{s} & \hat{z} \end{pmatrix} \]

\[(5\, c) \quad \hat{A} \leftrightarrow \hat{i} \iff \begin{pmatrix} c \mid D' \end{pmatrix} + \begin{pmatrix} c \mid = \end{pmatrix} = = = \]

The context condition for the \( \hat{A} \leftrightarrow \hat{i} \) correspondence resembles that of the general fronting rule (5a) but there is an important exception: after c, D and palatalized z, this correspondence only holds good (i) if a c or D appears at the lexical level too, and (ii) if there is a morpheme boundary between this character and \( \hat{A} \). The need for the first condition can be seen by comparing the locative singulars outci ‘father’ and tocē ‘flow’, or ktonezi ‘prince’ and boze ‘god’. The second condition is necessary to ‘protect’ such words as celb ‘healthy’. (So this root must be represented lexically as cAL, because the form cel would surface as *cal. If a representation like kAL were used, the morpheme boundary condition in (5c) would not be necessary—but this solution would be too abstract from a synchronic point of view.)

\[(6) \quad \text{Choices conditioned by lexical softness}\]

\[\begin{align*}
(6\, a) & \quad \{ \Rightarrow \begin{pmatrix} \text{t} & \text{d} \\ \hat{s} & \hat{z} \end{pmatrix} \} = = = = = = = = = \[ \]
(6\, b) & \quad \{ \} \leftarrow \begin{pmatrix} \text{t} & \text{d} \\ \hat{s} & \hat{z} \end{pmatrix} = = = = = = = = = \[ \]
\end{align*}\]

where \( P' \) is identical to \( P \) except that c and D are excluded. According to this rule, the feature ‘(’ can only occur after palatal consonants of the lexical level; ‘)’ cannot occur in the same context (\{ \} is the empty set: ‘)’ cannot correspond to anything, not even to the zero string \( \emptyset \)). The two feature signs are used in the o-stem vocative endings to account for the difference between rabe ‘slave’ and možu, možju ‘man’, which is not due to the vowel fronting rules.

\[(7) \quad \text{Palatalization of velars}\]

\[\begin{align*}
(7\, a) & \quad \begin{pmatrix} k \mid g \mid x \end{pmatrix} \leftrightarrow \begin{pmatrix} M \mid J \end{pmatrix} \[ \]
(7\, b) & \quad \begin{pmatrix} k \mid g \mid g \mid x \end{pmatrix} \leftrightarrow \begin{pmatrix} N \end{pmatrix} \[ \]
\end{align*}\]
\[ (7 \text{c}) \quad c \quad D \quad \equiv \quad [+M | J] \]

(For (7 c), cf. vocatives oťče ‘father’, kňže ‘prince’.)

(8) Consonant + J (excluding velars)\(^5\)

\[ (8 \text{a}) \quad t! \quad d! \quad J \]
\[ \quad \tilde{\text{s}}t \quad \tilde{\text{d}} \quad \equiv \quad = = = \]

‘‘!’’ is a sign that always precedes J at the lexical level and is used as a place-holder in epenthesis.

\[ (8 \text{b}) \quad s \quad z \quad \equiv \quad [J \quad nJ \quad lJ \quad l=] \]
\[ \quad \tilde{s} \quad \tilde{z} \quad \equiv \quad = = = = = \]

\[ (8 \text{c}) \quad j \quad \Rightarrow \quad [\quad \tilde{s} \quad \tilde{z} \quad \tilde{s} \quad t \quad \tilde{z} \quad d \quad] \quad = = \]

\[ (8 \text{d}) \quad \text{!J or } \text{!J } \text{or } \text{!J } \equiv \quad L \quad = \]

\[ (8 \text{e}) \quad \text{!J or } \text{!J } \text{or } \text{!J } \equiv \quad B \quad = \]

Note that, since !J is most often followed by ‘‘+’’, the third graphemic possibility after a labial consonant, viz. l’j, follows from (2b) (e.g. \text{na+stavl'}\text{J}+\varphi \equiv \text{nastavl'\varphi, nastavlj} \varphi, \text{nastavl'}\text{J} ‘I shall put’).

(9) Consonant alternation in the verb stem, I: velars and labials\(^6\)

\[ (9 \text{a}) \quad p \quad b \quad v \quad \equiv \quad = = = \quad [\quad \tilde{s} \quad t \quad] \]
\[ \quad \text{!} \quad \text{!} \quad \text{!} \quad \equiv \quad = = = = = \]

\[ (9 \text{b}) \quad k \quad g \quad x \quad \equiv \quad = = = = = \]

\[ (9 \text{c}) \quad k \quad g \quad x \quad \equiv \quad = = = = = \]

\[ (9 \text{d}) \quad \text{!J or } \text{!J } \text{or } \text{!J } \equiv \quad B \quad = \]

\(^5\) The use of a sequence of correspondence pairs in the correspondence part (left-hand side) of a rule must be regarded as a condensation of rules in which each of these pairs appears in the correspondence part in turn, the others being in the context-condition part.

\(^6\) Actually the rules (9 a) . . . (9 c) are implemented as if they were as follows:

\[ (i) \quad k \quad g \quad x \quad p \quad b \quad v \quad \equiv \quad = = = \quad [\quad \tilde{s} \quad t \quad] \]
\[ (ii) \quad \text{!} \quad \text{!} \quad \text{!} \quad \equiv \quad = = = = = \]

Scando-Slavica, Tomus 30 1984
The sign ‘.’ is put in the lexicon before the stem-final consonant of every verb having a consonantal stem in the infinitive and aorist. This is merely an implementational feature: since the automata simulating the two-level rules proceed from left to right, the automaton implementing (9a) and (9b) would try to delete every single velar and labial of every word and only after checking the right context would it realize that it was the wrong alternative. The automaton would work even without the ‘.’ trigger; its sole purpose is to make the computation more efficient. (This sign is used in rule (14a) to identify a certain class of verbs, because it already exists.)

(10) Consonant alternation in the verb stem, II: dentals

(10a) \[
\frac{\text{t} \cdot \text{d}}{\text{s}} \iff \text{t} + \text{t}
\]

(10b) \[
\frac{\text{t} \cdot \text{d}}{\emptyset} \iff \text{t} + \left[\begin{array}{c}
\emptyset \\
\emptyset
\end{array}\right]
\]

(10c) \[
\frac{\text{s} \cdot \text{z}}{\emptyset} \iff \text{s} + \emptyset
\]

(10d) \[
\text{z} \iff \text{t}
\]

Rule (10d) is, of course, a special case of voice assimilation but no such general rule of assimilation is needed in the synchronic system.

(11) Consonant alternation in aorist endings

(11a) \[
\frac{\emptyset}{\text{x}} \iff \left[\begin{array}{c}
\emptyset \\
\text{V'}
\end{array}\right] + = \text{V'}
\]

where V' = a vowel other than ħ.

(11b) \[
\frac{\emptyset}{\text{x}} \iff \left[\begin{array}{c}
\emptyset \\
\text{V'}
\end{array}\right] + = 0
\]

Note that $\emptyset \iff \text{x}$ is the default case according to rule (4a).

(11c) \[
\frac{\emptyset}{\text{x}} \Rightarrow \emptyset + = \text{V'}
\]

(11d) \[
\frac{\emptyset}{\text{x}} \Rightarrow \emptyset + = 0
\]

(11c) and (11d) account for the variation *pešē*/*pešē* ‘I stretched’ and *pešē* / *pešē* ‘they stretched’.

Scando-Slavica, Tomus 30 1984

12—849127

51
(12) Aorist Ablaut
   \[
   \begin{align*}
   (12\ a) \quad e & \quad o & H \quad K+\hat{\$} \\
   \epsilon & \quad \dot{a} & i & \quad \emptyset
   \end{align*}
   \]
   Example: *ne.s+\hat{\$} \leftrightarrow n\dot{e}s\dot{\nu} \quad \text{‘I carried’}.

(13) Avoiding the hiatus in aorist and imperfect endings
   \[
   \begin{align*}
   (13\ a) \quad & \epsilon \leftrightarrow \emptyset \\
   & \emptyset \leftrightarrow V = \emptyset
   \end{align*}
   \]
   In the dictionary the imperfect endings appear only in their full forms
   ėax̌, etc., and \& is realized as e by default. If the stem ends in a vowel,
   this rule deletes the first vowel of the imperfect ending and forces \& to
   be realized as a zero string.

(14) r̄ci ‘say!’; t̄ci ‘run!’; p̄ci ‘bake!’
   \[
   \begin{align*}
   (14\ a) \quad e & \leftrightarrow \emptyset \\
   b & \leftrightarrow = = = c
   \end{align*}
   \]

(15) Blocking verbal endings
   There are two rules whose formalization would be of no use to the reader
   because they refer to specific feature signs in the dictionary file. The first
   of these blocks the older type of sigmatic aorist in Leskien’s second verb
   class, permitting dviǧ and dvigox̌ ‘I moved’ but not *dvix̌. This rule
   is necessary because it is assumed in the dictionary that all verbs can
   have this form; only the occurrence of the other two types is regulated
   by continuation classes.
   The second rule blocks certain active past participles 1, such as *p̄x̌v̌
   for p̄x̌v̌ ‘having stretched’.

(16) Selecting the case ending remnant in definite adjectives
   The purpose of this rule has already been explained in section 3, in
   connection with the Adj lexicon.

(17) Coalescing endings in definite adjectives
   \[
   \begin{align*}
   (17\ a) \quad o & \leftrightarrow \% = j \\
   \emptyset & \leftrightarrow \emptyset \emptyset \emptyset \emptyset \emptyset
   \end{align*}
   \]
   This rule takes care of cases like nov+\%yj+2oǰ \leftrightarrow novyǰ (the feminine
   definite genitive singular) and nov+\%Āj+6oi \leftrightarrow novēi (the feminine
   definite dative singular). In the case of novyǰ, the second j is deleted by
   \((3\ d)\).
(18) **Peculiarities of the active participles**

(18a) \[ \begin{array}{c}
* \quad V+ \\
v \Leftrightarrow \\
== \\
i
\end{array} \]

"*" is a sign used in the present and first past active participles. In the latter, it is used as a placeholder for an epenthetic v if the verb stem ends in a vowel.

Rule (18b)—whose formalization would force us to introduce additional notational conventions—states that no "*" can be followed by a "",", even if there are intervening characters. The feature sign ""," is inserted in the nominal and pronominal case endings that cannot be used in the active participles. Thus the participle entries can safely refer to the Adj lexicon after the endings peculiar to participles (e.g. the fem. nom. pl. -i) have been listed.

(19) **Allomorphs of the past passive participle**

In the dictionary, there are two entries for the formative of this participle: /en and /t. The rules of this block define their correct distribution and account for the e \(\Leftrightarrow\) 0 correspondence after a vowel. In principle, it would have been possible to use only one lexical entry and write an additional n \(\Leftrightarrow\) t rule, but that would hardly have been a psycholinguistically natural solution.

(19a) \[ \begin{array}{c}
/ t \\
== \\
\Rightarrow \\
\begin{bmatrix}
\xi \\
= \\
= \\
\check{e}
\end{bmatrix}
\]

where \(\xi\), \(\check{v}\) or \(\check{e}\) must be the first vowel of the stem.

(19b) \[ \begin{array}{c}
/ e \\
== \\
\Rightarrow \\
\begin{bmatrix}
V \\
= \\
K
\end{bmatrix}
\]

where V cannot be a first-syllable \(\xi\) or \(\check{v}\). These rules must be supplemented by (19c) . . . (19e) describing the realizations of the participle formative (note that the default realizations t and en need not be mentioned):

(19c) \[ \begin{array}{c}
/ y \\
v \Leftrightarrow \\
\Rightarrow \\
\begin{bmatrix}
y \\
= \\
= 
\end{bmatrix}
\]

(19d) \[ \begin{array}{c}
/ i \\
j \Leftrightarrow \\
\Rightarrow \\
= 
\]

Condition: the vowels on the right-hand sides of (19c) and (19d) must be in the first syllable.
(19e) \( e \leftrightarrow V/ \theta = \theta \)

For the type kr\(\text{\=v}\)en\(\text{\=b}\) ‘covered, hidden’ (from kry/en+\(\text{\=b}\)), (19c) must be further supplemented with

(19f) \( y \leftrightarrow /\text{\=b} \rightarrow v \)

5. DISCUSSION

The present system is satisfactory to the extent it generates and analyses OCS word forms correctly. There are some minor points in which it is overly permissive: it accepts vocatives of all adjectives and some non-existing definite adjective forms like *veliiii ‘big’. These flaws are not serious, because the emphasis of the system is on automatic analysis, and not generation.

With greater justification it could be argued that the Cyrillic-based transliteration adopted is not ideal. The attempts to mimic the variable use of the ligatures \(, \), etc. and of the \( ^\prime \) sign perhaps make the system unnecessarily complicated. A linguistically more interesting model should be based on descriptions like Trubetzkoy’s (1968) and on phonemics that can be reconstructed from the Glagolitic alphabet—with \( \Delta, P, \phi \) rendered, say, as å, ü, ö. It may well be that the orthographic conventions of the various codices, including the Cyrillic ones, could be derived more easily from such a ‘Proto-OCS’ description than from the present Cyrillic-normalized version. This question should be studied before any really large dictionaries are converted into the two-level format.

A system containing more lexemes might also benefit if some of the alternation patterns in the present dictionary were converted into rules. These include certain Ablaut patterns (note that the aorist Ablaut is already treated with a rule), ov/u alternations, liquid metathesis, and the alternation between a nasal vowel and the sequence oral vowel + nasal consonant—all features of the verb morphology. But the mere fact that they can be made rules (which I do not doubt) does not necessarily mean that they should be. It is a question either of linguistic naturalness or computational efficiency, or both, depending on the aims of the system; and both these criteria might favour a lexical solution.

Linguistic evaluation of this two-level description is not a simple task. Take the morphophoneme Y, whose function is to mark the y’s whose
"soft" counterpart is ɛ and not i. Another solution would have been to assume double endings in the relevant cases, )y... and (ɛ..., the selection of which could have been effectively handled by the existing rules (6a) and (6b). Thus Y is, in a way, an unnecessary morphophoneme, but by using it we can acknowledge the fact that the y/ɛ alternation is found in several endings and is not arbitrary in the same way as the e/u alternation in the o-stem vocatives (rabe/moţu).

It is interesting to note that the rules would have allowed much more "abstract" or "archaic" lexical entries than those actually used. Nothing prevents us from relating žena 'woman' to a lexical-level gena, except synchronic plausibility. It would perhaps have been wiser to limit the palatalization rules (7a) and (7b) so that they only operated across the morpheme boundaries, as (7c) already does. In this case we could no longer treat g̣nati/ženq 'to chase' as an ordinary case of the ūna/en alternation pattern but examples of this kind are not numerous.

To sum up, I hope I have shown that Koskenniemi's two-level model is suitable for both the linguistic description and automatic analysis of OCS inflected word-forms and, consequently, for Slavonic morphology in general. Though some of my solutions may prove to be useful in two-level descriptions of Slavonic languages, all of them will have to be submitted to scrutiny when more lexical material is introduced and the various applications of the analysis are considered.

I wish to thank Kimmo Koskenniemi and Olli Blåberg for their advice and constructive criticism. They share no responsibility for my mistakes.

References
Sågvall, Anna-Lena, A System for Automatic Inflectional Analysis, Implemented for Russian, Stockholm 1973 (= Data Linguistica 8).

Scando-Slavica, Tomus 30 1984

55
Appendix

EXAMPLE OF AUTOMATIC MORPHOLOGICAL ANALYSIS

(The bold-faced lines have been typed by the user, all the others by the system. The comments on the right-hand side have been added to explain what is going on.)

**run §OCS/TWOL**

**(start the program)**

§RUNNING 4920

§?

3

**(enter the analysis mode)**

73 Characters in the Alphabet (85)

19 Automata Read In: 1970 States (3000)

125 Rows (200)

287 Columns (300)

120 Aligned Pairs (200)

67 ALT Lists (150) 191 ALT Items (250)

63 Lexicons Read in (90) 779 Nodes (30000)

716 Branches (30000) 462 Entries (10000)

739 Strings (120000) 6783 Characters (120000)

**kamy**

**(word-form to be analysed)**

**kamy**

**(corresponding lexical representation)**

stone N (MA) SG NOM

**(translation, morphosyntactic features)**

**kamene**

**(next form to be analysed)**

kamen+e

stone N (MA) SG GEN

kamen+e

stone N (MA) PL NOM

kamen+e

stone N (MA) SG LOC

*Scando-Slavica, Tomus 30 1984*
synU
syn+U
son N ⟨MA⟩ SG ACC
syn+U
son N ⟨MA⟩ SG NOM

rabU
rab+U
slave N ⟨MA⟩ PL GEN
rab+U
slave N ⟨MA⟩ SG ACC
rab+,U
slave N ⟨MA⟩ SG NOM

zeml’ a
zeml’+,a
earth, land N ⟨FE⟩ SG NOM

zemlj a
zeml’+,a
earth, land N ⟨FE⟩ SG NOM

zeml’ ja
zeml’+,a
earth, land N ⟨FE⟩ SG NOM

zmijE
zmij+Y
snake N ⟨FE⟩ SG GEN
zmij+Y
snake N ⟨FE⟩ PL ACC
zmij+Y
snake N ⟨FE⟩ PL NOM

toi
(two examples of the pronominal declension)
t+6oi
that DemPron ⟨FE⟩ SG LOC
t+6oi
that DemPron ⟨FE⟩ SG DAT
moi
moj+6Å
my PossPron ⟨FE⟩ DL NOM/ACC
moj+6Å
my PossPron ⟨NE⟩ DL NOM/ACC
moj+1U
my PossPron ⟨MA⟩ SG ACC
moj+9,i
my PossPron ⟨MA⟩ PL NOM
moj+1,U
my PossPron ⟨MA⟩ SG NOM

(verb examples)

imO
jIm+O
to take V =prf= PRES SG1

testI
te.k+tU
to run, to flow V =imp= SUP

grasU
gre.b+$U
to row, to dig V =imp= AOR SG1

dvizenU
dvig/en+U
to move V =prf= PCP PSS PAST INDEF ⟨NE⟩ PL GEN
dvig/en+U
to move V =prf= PCP PSS PAST INDEF ⟨FE⟩ PL GEN
dvig/en+U
to move V =prf= PCP PSS PAST INDEF ⟨MA⟩ PL GEN
dvig/en+U
to move V =prf= PCP PSS PAST INDEF ⟨MA⟩ SG ACC
dvig/en+,U
to move V =prf= PCP PSS PAST INDEF ⟨MA⟩ SG NOM

nareCesi
na+re.ke+Si
to call, to name V =prf= PRES SG2

Scando-Slavica, Tomus 30 1984
nariCeSi
na+riceJe+Si
to call, to name .1. \( V = \text{imp} = \text{PRES SG2} \)

nariCajeSi
\( \text{a non-existing form} \)

naricajeSi
na+riceaje+Si
to call, to name .2. \( V = \text{imp} = \text{PRES SG2} \)

viZdO
vid!J+O
to see \( V = \text{imp/prf} = \text{PRES SG1} \)

glagola
\( \text{examples of homography} \)
glagol+a
word N \( \langle \text{MA} \rangle \text{ SG GEN} \)
glagol+a
word N \( \langle \text{MA} \rangle \text{ DL NOM/ACC} \)
glagola+&
to speak \( V = \text{imp} = \text{AOR SG3} \)
glagola+&
to speak \( V = \text{imp} = \text{AOR SG2} \)

mysli
mysl+i
thought N \( \langle \text{FE} \rangle \text{ DL NOM/ACC} \)
mysl+i
thought N \( \langle \text{FE} \rangle \text{ SG VOC} \)
mysl+i
thought N \( \langle \text{FE} \rangle \text{ SG LOC} \)
mysl+i
thought N \( \langle \text{FE} \rangle \text{ SG DAT} \)
mysl+i
thought N \( \langle \text{FE} \rangle \text{ SG GEN} \)
mysl+i
thought N \( \langle \text{FE} \rangle \text{ PL ACC} \)
mysl+i
thought N \( \langle \text{FE} \rangle \text{ PL NOM} \)
myslj+i
to think V =imp= IMPV SG2/3
mysli+&
to think V =imp= AOR SG3
mysli+&
to think V =imp= AOR SG2

\textbf{tAxU}
t+2ÁxU
that DemPron 〈FE〉 PL LOC
t+2ÁxU
that DemPron 〈FE〉 PL GEN
t+2ÁxU
that DemPron 〈NE〉 PL LOC
t+2ÁxU
that DemPron 〈NE〉 PL GEN
t+2ÁxU
that DemPron 〈MA〉 PL LOC
t+2ÁxU
that DemPron 〈MA〉 PL GEN
te.k+§U
to run, to flow V =imp= AOR SG1

duSe
duS+o
soul, mind N 〈FE〉 SG VOC
dux+e
spirit N 〈MA〉 SG VOC

\textbf{tAsta}
te.p+§ta
to strike V =imp= AOR DL2
te.k+§ta
to run, to flow V =imp= AOR DL2

\textbf{ZenO}
Zen+O
woman N 〈FE〉 SG ACC
gen+O
to chase V =imp= PRES SG1

\textit{Scando-Slavica, Tomus 30 1984}
sAli
sA+l+i

to sow .1. V =imp= PCP ACT PAST2 ⟨MA⟩ PL
sA.d+l+i

to sit down V =prf= PCP ACT PAST2 ⟨MA⟩ PL

krai  
(examples of syncretism in paradigms)

kraj+ā
edge N ⟨MA⟩ SG LOC
kraj+y
edge N ⟨MA⟩ PL INS
kraj+u
edge N ⟨MA⟩ PL GEN
kraj+u
edge N ⟨MA⟩ SG ACC
kraj+u
edge N ⟨MA⟩ SG NOM
kraj+u
edge N ⟨MA⟩ PL NOM

velii

velij+ā
big, great A INDEF ⟨NE⟩ SG LOC
velij+y
big, great A INDEF ⟨NE⟩ PL INS
velij+u
big, great A INDEF ⟨NE⟩ PL GEN
velij+ā
big, great A INDEF ⟨NE⟩ DL NOM/ACC
velij+ā
big, great A INDEF ⟨FE⟩ DL NOM/ACC
velij+ā
big, great A INDEF ⟨FE⟩ SG LOC
velij+ā
big, great A INDEF ⟨FE⟩ SG DAT
velij+u
big, great A INDEF ⟨FE⟩ PL GEN
velij+ā
big, great A INDEF ⟨MA⟩ SG LOC
veli\,j + y
big, great A INDEF ⟨MA⟩ PL INS
veli\,j + U
big, great A INDEF ⟨MA⟩ PL GEN
veli\,j + U
big, great A INDEF ⟨MA⟩ SG ACC
veli\,j + ,U
big, great A INDEF ⟨MA⟩ SG NOM
veli\,j + ,i
big, great A INDEF ⟨MA⟩ PL NOM

novojO
nov + ojO
new A INDEF ⟨FE⟩ SG INS
nov + % + ?ojO
new A DEF ⟨FE⟩ SG INS

novOjO
nov + %Oj + 80
new A DEF ⟨FE⟩ SG ACC
nov + %Oj + ?ojO
new A DEF ⟨FE⟩ SG INS

borjemU
borj\,e + mU
to fight, to struggle \textit{V} =\textit{imp}= PRES PL1
borj\,+om+U
to fight, to struggle \textit{V} =\textit{imp}= PCP PSS PRES INDEF ⟨NE⟩ PL GEN
borj\,+om+U
to fight, to struggle \textit{V} =\textit{imp}= PCP PSS PRES INDEF ⟨FE⟩ PL GEN
borj\,+om+U
to fight, to struggle \textit{V} =\textit{imp}= PCP PSS PRES INDEF ⟨MA⟩ PL GEN
borj\,+om+U
to fight, to struggle \textit{V} =\textit{imp}= PCP PSS PRES INDEF ⟨MA⟩ SG ACC
borj\,+om+ ,U
to fight, to struggle \textit{V} =\textit{imp}= PCP PSS PRES INDEF ⟨MA⟩ SG NOM

dvignOtU
dvignO + tU
to move \textit{V} =\textit{prf}= SUP
dvign\OtU

to move V =prf= PRES PL3

1 (exit the analysis mode)
Quitting Xeq
l (exit program)
Quitting (Main)