# The Encoding Paradigm in LaTeX2e and the Projected X2 Encoding for Cyrillic Texts

A. Berdnikov, O. Lapko, M. Kolodin, A. Janishevsky and A. Burykin

*email:* berd@ianin.spb.su, olga@mir.msk.su, myke@iias.spb.su

**Abstract**. *This paper describes the X2 encoding which is designed to support Cyrillic writing systems for the multilanguage mode of LaTeX 2ε. The restrictions of the LaTeX 2ε kernel, the specific features of Cyrillic writing systems and the basic principles used to create X2 are considered. This projected X2 encoding supports all the Cyrillic writing systems known to us, although the majority of the accented letters need to be constructed from pieces. The general scheme of the X2 encoding was approved at CyrTUG-97 (the annual conference of Russian-speaking TEX users) and its final form was agreed on the* `cyrtex-t2` *mailing list.*

## 1. Introduction

The version of the X2 encoding that we describe aims to offer a tool which enables Latin-writing people to occasionally use Cyrillics in their documents. It is not designed for use as a Cyrillic encoding by native users of Cyrillic alphabets, since it is the task for the national TEX User Groups to organize the local encoding so that it is comfortable for their language. Moreover, since this projected X2 encoding does not contain Latin glyphs and ASCII codes in 32–127, use of this table could sometimes result in unpredictable results inside LaTeX 2ε (see Sections 2 and 3 for more details). Nevertheless this table is sufficient to include Cyrillic names, bibliography references and short quotations in your document in *nearly all* Cyrillic writing systems and languages without too large an increase of the number of fonts used for the purpose.

In parallel, we have tried to organize X2 so that it is useful not only for Latin-writing people. The Cyrillic characters in X2 are placed in such a way that the combination of the glyphs in positions 128–255 with those in positions 0–127 in the original CM fonts (the OT1 encoding used by default in LaTeX 2ε) is suitable to support the majority of the Cyrillic languages. In this sense the X2 encoding may be the basis for the standard for Cyrillic characters in TeX, in the same way that the CM fonts are the standard for Latin characters, independently of the computer platform used.

## 2. Encodings supported by the LaTeX Team

The following types of encodings are recognised by the LaTeX Team[1]:

OT$\langle n\rangle$ — essentially 7 bit "old" encodings. Typically these will be small modifications of the original TeX encoding, OT1 (for example, OT4, a variant for Polish).

T$\langle n\rangle$ — 8 bit Text encodings. T$\langle n\rangle$ encodings are the main text encodings that LaTeX uses. They have some essential technical restrictions to enable multilingual documents with standard TeX: (a) they should have the basic Latin alphabet, the digits and punctuation symbols in the ASCII positions, (b) they should be constructed so that they are compatible with the lowercase code used by T1. Further discussion of the technical requirements for T$\langle n\rangle$ encodings is given in Section 3.

X$\langle n\rangle$ — other 8 bit text encodings (eXtended, or eXtra, or X=Non Latin). Sometimes it may be necessary, or convenient, to produce an encoding that does not meet the restrictions placed on the T$\langle n\rangle$ encodings. Essentially arbitrary text encodings may be registered as X$\langle n\rangle$, but it is the responsibility of the maintainers of the encoding to clearly document any restrictions on the use of the encoding.

TS$\langle n\rangle$ — Text Symbol encodings. Encodings of symbols that are designed to match a corresponding text encoding (for example, paragraph signs, alternative forms for digits). The font style of fonts in a TS$\langle n\rangle$ encoding will ordinarily be changed in parallel with that of fonts in a T$\langle n\rangle$ encoding using NFSS mechanisms. As a result, at any moment the TS$\langle n\rangle$ font style is compatible with the T$\langle n\rangle$ font, and glyphs from a TS$\langle n\rangle$ font (accents, punctuation symbols, etc.) can be mixed with glyphs from the corresponding T$\langle n\rangle$ font.

S$\langle n\rangle$ — Symbol encodings. The style of fonts in S$\langle n\rangle$ encoding need not be synchronized with that of T$\langle n\rangle$ fonts. These encodings are used for arbitrary symbols, "dingbats", ornaments, frame elements, etc.

A$\langle n\rangle$ — encodings for special Applications (not currently used).

E$*$ — Experimental encodings, but those intended for wide distribution (currently used for the ET5 proposal for Vietnamese).

---

[1] The following text is slightly adapted from a post by David Carlisle to the `cyrtex-t2` mailing list.

L∗ — Local, unregistered encodings (for example, the LR1 encoding discussed below).
OM∗ — 7 bit Mathematics encodings.
M∗ — 8 bit Mathematics encodings.
U — Unknown (or unclassified) encoding.

Although the LaTeX Team's technical specifications for X⟨$n$⟩ encodings are less restrictive than those for "ordinary" text encodings, there are restrictions on their use, and some *desirable* properties for them to have. In particular:

- If the encoding does not have Latin letters in ASCII slots then the users must take care not to enter such text, otherwise "random" incorrect output will be produced, with no warning from the LaTeX system. Also, care must be taken with "moving" text that is generated internally within LaTeX (such as cross-references), which may fail if a different encoding is selected.
- To reduce the problems with cross-reference information, the LaTeX maintainers strongly recommend that at least the digits and "common" punctuation characters are placed in their ASCII slots.
- If the encoding uses a lowercase table that is incompatible with the lowercase table of T1, then it is not possible to mix this encoding and a T⟨$n$⟩ encoding within a single paragraph, and obtain correct hyphenation with standard TeX.

If the X⟨$n$⟩ encoding does not use a lowercase table that is compatible with that of T1, the package supporting this encoding should ensure that encoding switches only happen between paragraphs (or that hyphenation is suppressed when switching to the new encoding). It should be noted that this restriction on the lowercase table *only* applies to systems using standard TeX (version 3 and later). Using $\varepsilon$-TeX version 2 will remove the need for this restriction as the hyphenation system has been improved — it will use a suitable lowercase table for each language (the table will be stored along with each language hyphenation table).

## 3. Technical specifications for T⟨$n$⟩ encodings

There are two main restrictions to be fulfilled before an encoding may be considered as an encoding with the prefix "T" satisfying the requirements of the LaTeX 2$_\varepsilon$ kernel:

- the `\lccode`–`\uccode` pairs should be the same as they are in the LaTeX 2$_\varepsilon$ kernel (i.e. as they are in the T1 encoding);

— the Latin characters and symbols: `!`, `'`, `(`, `)`, `*`, `+`, `,`, `-`, `.`, `/`, `:`, `;`, `=`, `?`, `[`, `]`, `` ` ``, `|`, `@` (questionable), `0`–`9`, `A`–`Z`, `a`–`z` should be at the positions corresponding to ASCII, and the symbols produced by the ligatures `--`, `---`, ` `` `, `''` (at arbitrary positions).

If the encoding requires the redefinition of the values `\lccode`–`\uccode`, or if it does not contain the necessary Latin characters in the ASCII positions, it produces undesirable effects in some situations inside LaTeX $2_\varepsilon$ and makes the encoding incompatible with the general multilanguage mode.

One of the reasons for the unchangeable `\lccode`–`\uccode` values is a specific feature of the commands `\uppercase` and `\lowercase` provided by TeX. Consider the following example. If the command `\FR` selects French, `\GE` selects German, `\RU` selects Russian, etc., and if these languages require different `\lccode`–`\uccode` values, the following operation produces the wrong result:

```
\uppercase{ ... english ... \FR ... french
            \GE ... german ... \RU ... russian}
```

Since `\uppercase` processes its argument as a single block using the values `\lccode`–`\uccode` specified before it starts its work, the changes of `\lccode`–`\uccode` hidden *inside* the language-switching commands placed inside `\uppercase` play no role even if the argument of `\uppercase` is expanded in advance up to primitive commands:

```
\edef\temp{ ... english ... \FR ... french
            \GE ... german ... \RU ... russian}
\uppercase\expandafter{\temp}
```

As a result the headline and the table of contents entries produced using the command `\uppercase` may contain rubbish text (especially if the corresponding lines are composed from more than one language).

The other reason why the languages constituting multilanguage text should have the same values for `\lccode`–`\uccode` relates to automatic hyphenations. Hyphenations are inserted when the whole text of the paragraph has reached TeX's stomach. Following the language-switching commands TeX will use different hyphenation patterns for different languages, but to compare the text with corresponding hyphenation patterns the `\lowercase` command is applied to the original text. The `\lccode`–`\uccode` values active when the end of the paragraph is processed are used by this procedure, so that local changes in `\lccode`–`\uccode` inserted into language-switching macro are just ignored by TeX. In order to get correct automatic hyphenations, the languages used in the paragraph should have the same `\lccode`–`\uccode` values (or, more precisely, only `\lccode` values).

The reason to keep full ASCII in 32–127 (more precisely, to conserve the characters which have `\catcode=11` or `\catcode=12` in LaTeX's kernel) is less evident[2]. There is a large number of packages, class files and kernel commands which automatically generate text, e.g. commands like `\thesection` (which displays a representation of the section counter), `\ref`, etc. This generated text consists of text in internal LaTeX $2_\varepsilon$ representation and is composed from the following elements:

— encoding-specific commands such as `\"a` or `\textunderscore`,
— straight ASCII symbols like `a`...`z`, `0`...`9`, `.`, `,`, `:`, `;`, `(`, `)`, `+`, `-`, `!`, etc.,
— abbreviation ligatures: `--`, `---`, ` `` `, `''`.

When some output encoding *XXX* is active, such generated text is processed in the following way:

— Encoding-specific commands are handled by the LaTeX $2_\varepsilon$ encoding mechanism: (a) if the symbolic command is known in this encoding, the proper output with the glyph represented by this command is generated, (b) if the symbolic command is unknown, it results in a warning or error message informing the user that in encoding *XXX* the glyph *yyy* cannot be represented.
— The characters with `\catcode=11` or `\catcode=12` are passed straight on from input to output so that there is no way for LaTeX $2_\varepsilon$ to recognize that, for example, the digits `0`–`9` are not present in this encoding or that there are some other glyphs at their positions. It means that if some macro uses `\theequation` to display an equation counter value, arbitrary rubbish is produced as a result.
— The abbreviation ligatures like `--`, ` `` `, etc., are passed straight on also and in the same manner may result in something completely incorrect.

Since there is no way to update *all* packages so that they contain symbolic names like `\DigitO`, `\DigitI`, ..., `\DigitIX` instead of explicit ASCII characters `0`, `1`, ..., `9`, for compatibility reasons the T$\langle n\rangle$ encodings should contain the full set of ASCII characters and other symbols at their proper positions.

2 These arguments are due to Frank Mittelbach, who posted them to the mailing list *CyrTeX-T2* some time ago.

## 4. The `\lccode`–`\uccode` pairs reserved in the $\LaTeX\,2_{\varepsilon}$ kernel

The X2 encoding should follow the $\LaTeX\,2_{\varepsilon}$ agreements about `\lccode`–`\uccode` not to produce rubbish for the headings, table of contents, hyphenations inside paragraphs. As a result it contains predefined positions where the uppercase and lowercase letter forms should be placed, and some other positions where symbols, accents, punctuation marks, etc., should be placed.

It is easiest to explain the `\lccode`–`\uccode` pairs in the $\LaTeX\,2_{\varepsilon}$ kernel by considering the T1 encoding and the EC fonts (Table 1). Most *uppercase–lowercase* pairs are just shifted by 32 (`"20`) — `"61`↔`"41`, `"62`↔`"42`, etc.:

| | | | |
|---|---|---|---|
| lowercase letters: | "61 – "7a, | "a0 – "bc, | "e0 – "ff |
| uppercase letters: | "41 – "5a, | "80 – "9c, | "c0 – "df |

All other positions except `"19`, `"1a`, `"9d` and `"9e` are reserved for non-letter glyphs which are not affected by `\uppercase` and `\lowercase` and are not used in hyphenation patterns. But the characters `"19`, `"1a`, `"9d`, `"9e` have the following `\uccode`–`\lccode` assignments:

| Code | | | `\lccode` | | | `\uccode` | | |
|---|---|---|---|---|---|---|---|---|
| `"19` | ı | (dotless i) | `"19` | ı | (dotless i), | `"49` | *I* | (letter I) |
| `"1a` | ȷ | (dotless j) | `"1a` | ȷ | (dotless j), | `"4a` | *J* | (letter J) |
| `"9d` | | (dotted I) | `"69` | *i* | (letter i), | `"9d` | | (dotted I) |
| `"9e` | *đ* | (stroked d) | `"9e` | *đ* | (stroked d), | `"d0` | *Đ* | (stroked D). |

In parallel with these specific `\uccode`–`\lccode` values the characters $i$–$I$ (`"69`–`"49`), $j$–$J$ (`"6a`–`"4a`), *Đ*–*ð* (`"d0`–`"f0`) form their own ("standard") pairs `\uccode`–`\lccode`. It helps to economize encoding cells but as a result of this dirty trick commands like

`\uppercase{\lowercase{...}}`, `\lowercase{\uppercase{...}}`

work incorrectly[3]. From the point of view of the encoding designer such "specific" assignments for `\uccode`–`\lccode` mean that the positions 25 (`"19`), 26 (`"1a`), 157 (`"9d`), 158 (`"9e`) can hardly be used for some letters and should be used for character glyphs which are referred to by using the command `\char`⟨*nn*⟩, which will not be affected by `\uppercase` and `\lowercase`.

3 Frank Mittelbach has explained that the $\LaTeX$ Team didn't design T1 but rather inherited it. There are more rational ways to fix these peculiarities, but for compatibility and reusability reasons they are kept in $\LaTeX\,2_{\varepsilon}$.

# 5. The basic principles used in X2

The "glyph container" X2 should include all the glyphs necessary to represent in LATEX 2ε documents containing texts from stable Cyrillic languages. The basis of X2 is the Russian alphabet (since it is the main language used for publication in Cyrillic). Because of the many varieties in old Cyrillic texts, only modern alphabets which are still in use are included in X2. As an exception, we have nevertheless included the characters Ѣ/ѣ, Ѫ/ѫ, Ѵ/ѵ which were used in Russian and Bulgarian texts at the beginning of the 20$^{\text{th}}$ century.

X2 is designed so that by combining `"00–"7f` from OT1 and `"80–"ff` from X2 one can construct an encoding which is adequate to support the commonest Cyrillic languages. This permits use of X2 as a component of the base Cyrillic encoding for a variety of TEX formats (*Plain*, AMS-TEX, *BLUE*TEX, LATEX 2.09, etc.) as well as LATEX 2ε. (The local encoding is the one called LR1 below. The design aim for LR1 was to select glyphs required by the most widely-used languages, and to place them into positions 128–255 of X2.)

Unfortunately the full set of glyphs including accented letters is too big to fit in 256 characters, especially taking into account the `\lccode`–`\uccode` restrictions. So it is necessary to accept some principles of selection which enable us to decrease the number of Cyrillic glyphs included in X2:

1. All glyphs used in publishing for some language are included in X2 if they cannot be constructed as accented letters or letters with additional modifiers using TEX commands.
2. The X2 encoding includes all punctuation symbols, digits, mathematical symbols, accents, hyphens, dashes, etc., to form the full set of symbols necessary for Cyrillic typography.
3. The additional Cyrillic letters which are used in PC 866 and MS Windows 1251 code pages are included in X2 even if they are accented forms.
4. Variant glyphs for Cyrillic alphabets are also included in X2 if there is some free space and if different languages use different variants.
5. Glyphs which are not used now but which were used at some stage in the 20$^{\text{th}}$ century may be included if there are serious reasons to do so (as, for example, with the old Russian and Bulgarian letters).
6. Glyphs which were used in old Cyrillic texts before 1900 (Old Slavonic, Church Slavonic, Glagolitic, old phonetic symbols, etc.) should be moved to a separate glyph container. There could also be an additional glyph container to collect the exotic glyphs used in some contemporary Cyrillic texts.
7. When jettisoning accented letters it is necessary to take into account that they may be necessary for hyphenation patterns for some languages (if

such patterns have been created or if there is a chance that they will be created sometime). For example, accented letters for Russian, Ukrainian and Belorussian, Kazakh, Tatar, and Bashkir are included in X2.

8. When deciding whether to jettison an accented letter that is used in a language supported by LR1, one must keep in mind that only the CM accents are available in that encoding.
9. The following priorities are used when the accented letters or letters with simple modifiers are thrown away: (0) letters which are easily constructed by the internal command `\accent` (so that the letters using accents available in CM fonts have lower significance); (1) letters which contain a centered diacritic below the letter (cedilla, ogonek, dot, macron) and are easily constructed using a command similar to `\c` in *Plain* TEX; (2) letters which contain a horizontal stroke positioned symmetrically; (3) letters which require special alignment of accents and modifiers.
10. Accents and modifiers used in Cyrillic are included in X2 even if all accented forms are included in X2 for some other reasons (an example is *cyrillic breve* used for Й and Ў).
11. Latin letters or glyphs which are similar to some Latin letter (used in Macedonian, Kurdish, etc.) are placed at the same positions as the Latin letters are in ASCII. Among other things, this increases the number of languages supported by the LR1 encoding.

## 6. Glyphs used in X2

The X2 encoding is shown in Table 2. The Russian letters А–Я, а–я (except Ё and ё) are placed in the only region in the encoding table where 32 consecutive letter positions are available — i.e. positions `"c0–"df` and `"e0–"ff`. The Russian letters Ё and ё are placed at the end of the block `"80–"9c` and `"a0–"bc` which simplifies the ordering of non-Russian letters. Latin letters and letters similar to Russian letters are placed as in ASCII. Letters used in other Cyrillic alphabets are grouped into the parts `"80–"ff` and `"00–"7f` of the encoding table according to the "popularity" of corresponding languages (to satisfy the requirements of the LR1 encoding). They are placed in free positions reserved by LATEX $2_\varepsilon$ for letters in some quasi-alphabetic order. The old Russian and Bulgarian letters are placed at the end of the block of letters in `"00–"7f`.

Accents and modifiers are placed in X2 at `"00–"1f`; those also used by T1 are placed at the same positions as in T1. The same is true for additional symbols produced by the ligatures `--`, `''`, etc. The punctuation symbols, digits, mathematical symbols, etc., are placed as they are positioned in ASCII. A special case is made of the symbols № ¤ § „ « » which are essential for Russian typography. These symbols are placed in `"80`–`"ff` at the positions reserved for symbols, to guarantee the correctness of the LR1 encoding.

Some accents (macron, dot) can be used as lower accents as well for transliteration systems. In some specific cases the upper comma (`"1b`) and lower comma are also used as accents. The lower accents will be constructed using TeX commands from the upper accents available in X2.

The accents ̑ (`"12`) and ̏ (`"13`) are used as stresses in Serbian; there is no letter in any Cyrillic language where these symbols are used as "normal" accents.

The quasi-letters ' (apostrophe, `"27`), " (double apostrophe, `"22`) and Ӏ (palochka, `"0d`) are used like letters in some languages but do not have uppercase and lowercase forms (i.e. for these letters the uppercase form is just the same as the lowercase form).

Single quotes are not used in Cyrillic writing, and for this reason there is no need to keep single French quotes. In their place, the angular brackets ⟨ (`"0e`) and ⟩ (`"0f`) are provided. Angular brackets *are* used in Cyrillic typography, and it is good if their style is changed in parallel with the style of other symbols.

The Cyrillic breve "˘" (`"14`) is a very famous glyph (it is even included in Adobe and WordPerfect Cyrillic fonts). Although all letters with this accent (Й/й, Ў/ў) are included in X2, it is included as a special glyph as well.

Cedilla "¸" (`"0b`) and ogonek "˛" (`"0c`) are used by some letters already included in X2 (Ҙ, Ҫ, Ҿ). These letters have variant forms where *cedilla* could be oriented to the left or to the right depending on the user's taste. Also, some applications use *ogonek* instead of *descender* for Қ, Ҳ, Ґ, Ҭ, etc. The availability of *cedilla* and *ogonek* in X2 makes it possible to satisfy these needs.

Percentage zero "₀" (`"18`) is included as a useful idea borrowed from the T1 encoding and EC fonts: this symbol is used to convert "%" into "‰" and "‱".

The compound word mark (`"17`) — as in the T1 encoding and EC fonts the "empty" character with zero thickness, height 1*ex* and no visual image can be useful for special applications (such as hyphenation of compound words and accents placed over the invisible space between two letters).

The dotless letters "ı" and "ȷ" (`"19` and `"1a`) are included since the Latin letters I/i and J/j are used in some Cyrillic alphabets. In any case their positions correspond to specific `\lccode`–`\uccode` values, and these cells cannot be used for anything else.

Punctuation ligatures, i.e. the symbols produced by the abbreviations `--` (endash, `"15`), `---` (emdash, `"16`), ` `` ` (opening English quotes, `"10`), `''` (closing

English quotes, `"11`) are used in the same manner and are placed at the same position as in T1, as is `-` (the hyphen used for hanging hyphenation, `"7f`).

Another special case is made of the positions `"1c`–`"1f`. These positions are reserved for exotic characters which may be discovered (or proposed) in some texts in future. Although in principle it is not possible to put *true letters* in these positions due to the restrictions on `\lccode`–`\uccode` values, we could place glyphs here which *simulate* letters, i.e. glyphs that are converted like letters by the LaTeX $2_\varepsilon$ `\MakeUppercase` and `\MakeLowercase` transformations[4], but which cannot be used in hyphenation patterns and which break the automatic hyphenation whenever they appear in a word. We currently expect that these positions will be used for the Nivh letters Ӻ/ӻ (`"1c`/`"1d`) and Ӿ/ӿ (`"1e`/`"1f`), but we would emphasize that a final decision has not yet been made.

## 7. The Cyrillic glyph container X2 versus *T2*∗ Cyrillic encodings

To construct a T$\langle n\rangle$ encoding, we must keep the ASCII glyphs in positions 32–127 for reliable work in case of multiple languages. However, this requirement is very restrictive (it leaves only 61 positions for non-ASCII letters), and it is even more restrictive for Cyrillic encodings where it is also necessary to keep 32 base Russian letters in each encoding (since they are encountered in almost all Cyrillic alphabets). As a result, most characters in the T$\langle n\rangle$ tables are the same, and to fit the Cyrillic letters of X2 into T$\langle n\rangle$ encodings would require at least *three* tables.

To support each such T$\langle n\rangle$ encoding it is necessary to have a separate font class like the EC fonts. To keep such an enormous numbers of fonts is too high a price for people who only use Cyrillic occasionally. On the other hand, if all Cyrillic glyphs are put in one table without the Latin letters in 32–127, but these glyphs satisfy the `\lccode`–`\uccode` requirements, one table and one font class is enough if the user obeys some elementary rules of safety.

There is a similar situation for Old Slavonic characters and some other encodings which are only occasionally used by normal users. To resolve this problem, "glyph containers" like X2 could again be helpful. The "glyph container" encodings X$\langle n\rangle$ should be an intermediate case between T$\langle n\rangle$ and "free style" X$\langle n\rangle$: such encodings do not have ASCII in 32–127, but they have correct `\lccode`–`\uccode` values.

[4] These commands first check the internal list `\@uclclist` composed from the encoding commands joined by an uppercase–lowercase relation, and only then the primitive operations `\uppercase` or `\lowercase` are applied.

Currently the LaTeX Team only supports the T$\langle n\rangle$ encodings and TS$\langle n\rangle$ encodings, while the support an X$\langle n\rangle$ encoding is entirely the responsibility of the designer of the encoding. It seems to us that the LaTeX Team should support of "glyph container" encodings: such support should include the registration procedure for glyph containers and formalization of the list of exceptions where the glyph container encodings produce undesirable results.

## 8. Preliminary remarks about the TS2 encoding

For typographical reasons, "wide" versions of some accents — macron, tilde, breve, etc. — are desirable. These versions would be used for extra wide letters: as compared with the Latin alphabet, Cyrillic has a far higher proportion of wide letters. Such wide versions of the accents are good candidates for a TS2 encoding. Similarly, the lowercase/uppercase variants of cedilla and ogonek may be a good contribution to TS2.

The letters Љ/љ and Њ/њ used in some Cyrillic languages are actually ligatures "Л+Ь" and "Н+Ь". As well as the uppercase and lowercase forms there is also a *title* form for these letters: the combination of the uppercase form for "Л" or "Н" and the bowl for the lowercase "ь". This form is used for titles where the first letter is capital while the other letters are ordinary (a similar effect occurs for "IJ" used in Dutch). Such title letters should be placed in TS2 and shared by X2 and *T2*∗ encodings.

To construct some exotic letters from pieces, special modifiers are necessary: horizontal stroke "-", vertical stroke "ˌ", diagonal strokes "ˎ" and "ˏ". The diagonal strokes are used only for letters Ꞓ/ꞓ (Enetz) and Ᵽ/ᵽ (Saam, or Lappish). Vertical strokes are used only for letters Ҝ/ҝ and Ҹ/ҹ which are already included in X2. Horizontal strokes are used in several Cyrillic letters (Ғ/ғ, Ҟ/ҟ, Ұ/ұ, etc.) and at least two such letters (Ӻ/ӻ and Ӿ/ӿ in Nivh language) are currently outside X2. There are serious reasons to keep these modifiers in TS2: there are still minor languages for which alphabets based on Cyrillic are in development. The availability of these modifiers in TS2 would support such developments without the necessity to include more glyphs in X2 and *T2*∗ encodings.

## 9. The weak points of X2

X2 does not contain the ASCII characters required for T$\langle n\rangle$ encodings. It does not contain accented letters, and thus (for some languages) throws the user back on the `\accent` primitive which prevents construction of correct hyphenation tables and destroys kerning pairs. It is also overloaded (to some extent) with rare glyphs, which arise from the attempt to collect *all* Cyrillic glyphs in one table. So it appears to be a good *Cyrillic glyph container* suitable for Latin-text users who use Cyrillic from time to time, but it is not an *encoding* that satisfies the needs of native Cyrillic writers. (Such additional encodings which satisfy the specifications of the LaTeX Team and which are comfortable for native Cyrillic users are to be created separately.)

An important disadvantage of the current X2 project is that two letters — **Ӻ**/**ӻ** and **Ӿ**/**ӿ** — necessary for Nivh language are not included here. Although these letters can be constructed from pieces available in X2 and TS2, the effort required is considerable; fortunately, Nivh is not widely used. A similar situation arises with the Enetz letters **Ҫ**/**ҫ** and the Saam (Lappish) letters **Ҏ**/**ҏ** which are also constructed from pieces, some of which will have to be taken from TS2. This particular problem is relieved by the fact that nobody but a few linguists uses these letters: no books are published in Enetz, and all but one publication in Saam use the glyph **Р̦**/**р̦** for this purpose.

Another disadvantage of minor importance is that there are two glyphs (**Ѣ**/**ѣ** and **Ѳ**/**ѳ**) which correspond to logically different letters: **Ѣ**/**ѣ** stands for Saam *semisoft sign* and for old Russian *yat*, and **Ѳ**/**ѳ** stands for *o-barred* and old Russian *fita*. Although graphically these symbols are similar, they are different logically. This situation can be accepted taking into account the status of X2 as a glyph table rather than a table for direct text coding. In structured markup, the ambiguity would be addressed by assigning *two* symbolic names for each glyph (say, `\yat`/`\semisft` and `\fita`/`\obarred`) and only using the semantically correct one to code texts.

Some preliminary information about exotic glyphs and pure phonetic symbols has been provided by linguists studying some minor writing systems. These letters and symbols are not currently included in X2 although the reserved positions in `"1c`–`"1f` could be used for this purpose. The reason for not including the glyphs at this stage is that the writing systems are very unstable and are subject to change from publication to publication. There is no justification for including such symbols in the version of X2 proposed as a *standard* until the situation becomes stable.

It seems that all specific Cyrillic glyphs are included in X2, but there is also a chance that some minor writing system is omitted. There is also a chance that linguists will suggest a new alphabet for some minor language using their own glyphs not available in X2. Until this happens we can consider X2 a comprehensive glyph container for modern Cyrillic texts (although not very comfortable and not specifically adjusted for intensive Cyrillic writing).

The question of the specific typographic requirements of italic writing in Bulgarian (mentioned in [1]) remains, but the probability is that these "requirements" are merely æsthetic features of a particular set of fonts.

## Acknowledgements

We are thankful to Vladimir Volovich and Werner Lemberg for their work on macro support for X2 encoding and to the participants of the mailing list *CyrTeX-T2* who discussed enthusiastically the X2 and *T2∗* problems. (To subscribe to this mailing list you should send to `Majordomo@vvv.vsu.ru` an email with the command: `subscribe cyrtex-t2` *`your-email-address`*.)

We are grateful to Robin Fairbairns for his time spent polishing the text of this paper.

This research was partially supported by a grant from the Dutch Organization for Scientific Research (NWO grant No 07–30–007).

## Bibliography

[1] A. Berdnikov, O. Lapko, M. Kolodin, A. Janishevsky, and A. Burykin, *Alphabets Necessary for Various Cyrillic Writing Systems (Towards X2 and T2 Encodings).* In Proceedings of *EuroTeX* '98, St. Malo, March 1998, *Cahiers GUTenberg* 28–29 (1998), 32–43.

Table 1 – The T1 encoding.

| | x0/x8 | x1/x9 | x2/xA | x3/xB | x4/xC | x5/xD | x6/xE | x7/xF |
|---|---|---|---|---|---|---|---|---|
| 0x | ` | ´ | ˆ | ˜ | ¨ | ˝ | ˚ | ˇ |
| 0x | ˘ | ¯ | ˙ | ¸ | ˛ | ‚ | ‹ | › |
| 1x | “ | ” | „ | « | » | – | — | |
| 1x | ‰ | ı | ȷ | ﬀ | ﬁ | ﬂ | ﬃ | ﬄ |
| 2x | | ! | " | # | $ | % | & | ’ |
| 2x | ( | ) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 3x | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | A | B | C | D | E | F | G |
| 4x | H | I | J | K | L | M | N | O |
| 5x | P | Q | R | S | T | U | V | W |
| 5x | X | Y | Z | [ | \ | ] | ^ | _ |
| 6x | ‘ | a | b | c | d | e | f | g |
| 6x | h | i | j | k | l | m | n | o |
| 7x | p | q | r | s | t | u | v | w |
| 7x | x | y | z | { | \| | } | ~ | - |
| 8x | Ă | Ą | Ć | Č | Ď | Ě | Ę | Ğ |
| 8x | Ĺ | Ľ | Ł | Ń | Ň | Ŋ | Ő | Ŕ |
| 9x | Ř | Ś | Š | Ş | Ť | Ţ | Ű | Ů |
| 9x | Ÿ | Ź | Ž | Ż | IJ | İ | đ | § |
| Ax | ă | ą | ć | č | ď | ě | ę | ğ |
| Ax | ĺ | ľ | ł | ń | ň | ŋ | ő | ŕ |
| Bx | ř | ś | š | ş | ť | ţ | ű | ů |
| Bx | ÿ | ź | ž | ż | ij | ¡ | ¿ | £ |
| Cx | À | Á | Â | Ã | Ä | Å | Æ | Ç |
| Cx | È | É | Ê | Ë | Ì | Í | Î | Ï |
| Dx | Ð | Ñ | Ò | Ó | Ô | Õ | Ö | Œ |
| Dx | Ø | Ù | Ú | Û | Ü | Ý | Þ | SS |
| Ex | à | á | â | ã | ä | å | æ | ç |
| Ex | è | é | ê | ë | ì | í | î | ï |
| Fx | ð | ñ | ò | ó | ô | õ | ö | œ |
| Fx | ø | ù | ú | û | ü | ý | þ | ß |
| | x0/x8 | x1/x9 | x2/xA | x3/xB | x4/xC | x5/xD | x6/xE | x7/xF |

Table 2 – The projected X2 encoding.

| | x0/x8 | x1/x9 | x2/xA | x3/xB | x4/xC | x5/xD | x6/xE | x7/xF |
|---|---|---|---|---|---|---|---|---|
| 0x | ` | ´ | ˆ | ˜ | ¨ | ˝ | ˚ | ˇ |
| 0x | ˘ | ¯ | ˙ | ¸ | ˛ | I | ⟨ | ⟩ |
| 1x | “ | ” | ̑ | ̏ | ̈̆ | – | — | |
| 1x | 0 | ı | ȷ | ’ | • | • | • | • |
| 2x | ␣ | ! | " | # | $ | % | & | ’ |
| 2x | ( | ) | * | + | , | - | . | / |
| 3x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 3x | 8 | 9 | : | ; | < | = | > | ? |
| 4x | @ | Æ | Ꚋ | Ћ | Ҽ | Ҿ | Ҟ | Ҡ |
| 4x | Ӆ | I | J | Љ | Ӎ | Њ | Ҩ | Ҧ |
| 5x | Ҏ | Ԛ | Ҭ | S | Ҵ | Џ | Ҹ | Ԝ |
| 5x | Ѣ | Ѫ | Ѵ | [ | \ | ] | ˆ | _ |
| 6x | ‘ | æ | ђ | ћ | ҽ | ҿ | ҟ | ҡ |
| 6x | ӆ | i | j | љ | ӎ | њ | ҩ | ҧ |
| 7x | ҏ | ԛ | ҭ | s | ҵ | џ | ҹ | ԝ |
| 7x | ѣ | ѫ | ѵ | { | \| | } | ˜ | - |
| 8x | Ґ | Ғ | Ӷ | Ҕ | Һ | Җ | Ҙ | Ӡ |
| 8x | Ї | Қ | Ҝ | Ӄ | Ԓ | Ң | Ҥ | Ӈ |
| 9x | Ө | Ҫ | Ў | Ү | Ұ | Ҳ | Ӽ | Ҷ |
| 9x | Ӌ | Є | Ә | Ԑ | Ё | № | ¤ | § |
| Ax | ґ | ғ | ӷ | ҕ | һ | җ | ҙ | ӡ |
| Ax | ї | қ | ҝ | ӄ | ԓ | ң | ҥ | ӈ |
| Bx | ө | ҫ | ў | ү | ұ | ҳ | ӽ | ҷ |
| Bx | ӌ | є | ә | ԑ | ё | „ | « | » |
| Cx | А | Б | В | Г | Д | Е | Ж | З |
| Cx | И | Й | К | Л | М | Н | О | П |
| Dx | Р | С | Т | У | Ф | Х | Ц | Ч |
| Dx | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
| Ex | а | б | в | г | д | е | ж | з |
| Ex | и | й | к | л | м | н | о | п |
| Fx | р | с | т | у | ф | х | ц | ч |
| Fx | ш | щ | ъ | ы | ь | э | ю | я |
| | x0/x8 | x1/x9 | x2/xA | x3/xB | x4/xC | x5/xD | x6/xE | x7/xF |