Typesetting the Cyrillic Alphabet with Ω: Basic Ideas

Yannis Haralambous[*] John Plaice[†]

August 26, 1994

Abstract

The fundamental aspects of typesetting the Cyrillic alphabet with Ω are presented. A brief introduction to Ω is followed by a description of the modern Cyrillic part of the ISO 10646-1/ Unicode standard. The Ω Translation Processes used for Cyrillic typesetting are given. Finally, a 16-bit TEX output encoding for all Cyrillic characters is presented, as is an implementation of this output encoding through one virtual 16-bit font and two real 8-bit fonts.

1 General Introduction to Ω

The Ω program is an extension of TEX that is being developed by the authors. The most important innovations in the current version of Ω are:

1. The internal encoding in Ω is ISO 10646-1/Unicode, thereby allowing 65,536 characters rather than standard TEX3's 256.
2. It includes filtering processes, which can be loaded and unloaded dynamically, whose syntax resembles that of the well-known Lex lexical analyzor generator.

The basic idea of Ω is to separate the three main components of a typesetting system:

User interface — Information exchange — Typography

These components are not clearly separated in TEX, and this causes many problems when languages other than English are dealt with. For example, in the case of Cyrillic, the AMS input encoding used for the *Mathematical Reviews* is quite popular. In this input method, in many cases, a Cyrillic letter is accessed by a single Latin character, so there is only one code for each of the three forms of the letter: input, internal encoding and output encoding. Since Russian Cyrillic letters are more numerous than plain Latin alphabet letters, tricks were used to access them. These tricks are actually internal ligatures, stored in the font metric information. In this way, the combination of ASCII codes `t` and `s` produces the Cyrillic letter ц. The problem with this scheme is that letters `t` and `s` exist in their own right, and a possible hyphenation `t-s` has the effect of replacing ц by т-с (!). This confusion exists because the operation '`ts` → ц', which is an input method that is as good as any other, interferes with internal operations of the typesetting process, such as hyphenation. The letter ц should be recognized as an independent entity before typesetting begins.

[*]Centre d'Études et de Recherche sur le Traitement Automatique des Langues, Institut National des Langues et Civilisations Orientales, Paris. Private address: 187, rue Nationale, 59800 Lille, France, Fax: +33 20.40.28.64, ISDN/Numéris: +33 20.15.81.77, `Yannis.Haralambous@univ-lille1.fr`

[†]Département d'informatique, Université Laval, Ste-Foy, Québec, Canada G1K 7P4, Fax: +1 418.656.2324, `plaice@ift.ulaval.ca`

Those fortunate enough to be able to view Cyrillic characters on their screens normally use 8-bit Cyrillic input schemes (like KOI-8 or ISO 8859-5) to input Cyrillic characters. Two methods are applied to use the same fonts with the new input method: either make these 8-bit characters active, making the expansion be the character position in the font, or re-encode the fonts, directly or by using a virtual font mechanism.

In the first case, one has to use macros ('active characters') to convert an input scheme into the internal encoding; these macros may interfere with other, higher level, macros. In the second case, the number of fonts is multiplied by the number of possible encodings. Neither solution is natural, since fonts and macros should only serve for typography, and not for re-encoding purposes.

Finally, whatever the case may be, it is impossible to place all of the Cyrillic characters (including accented vowels, which are very useful for dictionaries and textbooks) in a single 8-bit font since they are just too numerous.

1.1 How Ω deals with input/output encodings

The filtering processes of Ω, called Ω Translation Processes, work as follows: every set of consecutive tokens of `\catcode` 11 (Letter), 12 (Other character), 10 (Space) or 5 (End of line) is placed in a buffer. Whenever a token of some other `\catcode` is reached, the buffer is filtered, then further processed by Ω.

For most scripts, the following scheme is used: two Ω Translation Processs are applied, the first one sending the user input to ISO 10646-1/Unicode (this is the 'User Interface $\rightarrow$ Information exchange' part) and the second one sending the ISO 10646-1/Unicode encoded text into the output font encoding (this is the 'Information exchange $\rightarrow$ Typography' part). Of course, if the text is already encoded in ISO 10646-1/Unicode, only the second Ω Translation Process is needed.

Through this scheme, one can use any combination of input encoding and output font, without the need of active characters or virtual font constructions. Creating an Ω Translation Process is easy. First one writes a Lex-like file, with *patterns* and *actions* to be taken when the patterns are encountered. For example, to replace `sh` by the ISO 10646-1/Unicode code for ш, one uses the following lines:

```
‘s’‘h’ => @"0448 ;
‘S’(‘H’|‘h’) => @"0428 ;
```

In this way, `sh` becomes the *transliteration* for ш. This is part of the first Ω Translation Process. The second Ω Translation Process translates ISO 10646-1/Unicode characters into font positions. Assuming that glyphs ш, Ш are at positions `"0068`, `"0048` of the (16-bit) virtual font, the corresponding lines of the second Ω Translation Process file are:

```
@"0448 => @"0068 ;
@"0428 => @"0048 ;
```

Ω hyphenates after having applied the filters, i.e., the characters ш, Ш are now represented by `"0068` and `"0048`. This is the encoding in which the hyphenation patterns are written, so hyphenation is not affected by the transliteration, and is completely oblivious to the input method. The input method and the output font are now completely independent.

1.2 'Big' versus 'small' fonts

Ω accepts (virtual) fonts with 2^{16} (65,536) characters and $2^{16} \times 2^{16}$ (4,294,967,296) kerning pairs or ligatures, so that all Cyrillic characters (European and Asian) can be included in the same font (together with characters for Latin alphabet languages, African, IPA, and Greek). For this, extended versions of the `tfm`, `vpl` and `vf` formats have been defined: they are, respectively, `xfm`, `xvp` and `xvf`. An extended version of `vptovf`, called `xvptoxvf`, converts *extended virtual property files* (`xvp`s) into *extended virtual fonts* (`xvf`s).

These 'big' fonts are always virtual: an extension of Peter Breitenlohner's `dvicopy`, called `xdvicopy`, converts a `dvi` file using 'big' virtual fonts into a `dvi` file using only 'small' (8-bit) *real* fonts. To avoid the time-consuming process of de-virtualizing a `dvi` after every compilation by Ω, it would be convenient if `dvi` drivers were updated to handle 16-bit fonts, be they real or virtual. Tom Rokicki has promised to upgrade `dvips` in this direction.

When PostScript fonts are used, one can produce 'big' virtual fonts using Alan Jeffrey's `fontinst.tex`.

1.3 Solving the `\uppercase` problem

An Ω Translation Process is used to convert letters into their lowercase or uppercase representation. It has the advantage of being exclusively ISO 10646-1/Unicode-dependent: it depends neither on the input nor on the output encoding. See section 4.2 for an example Ω Translation Process for uppercase conversion.

2 Modern Cyrillic and ISO 10646-1/Unicode

Modern Cyrillic uses positions `"0400`–`"04FF` of ISO 10646-1/Unicode.[1] These characters cover the following languages: Russian, Ukrainian, Byelorussian, Bulgarian, Serbian, Macedonian, Moldavian, Azerbaijani, Altaic, Uzbekian, Abkhasian, Bashkir, Yakut, Tatar, Tajik, Mari, Chuvash, Kazakh, Khanty, Chukchi, Khakassian and Cyrillic-script Mongolian. Appendix A contains a table for these positions.

Certain Unicode 1.0 characters are not present anymore in ISO 10646-1/Unicode; these are ӄ Ӄ (`"04C6` `"04C5`) and ӽ Ӽ (`"04CA` `"04C9`), used in Uzbek. Apparently they have been identified with characters қ Қ (`"049B` `"049A`) and ҳ Ҳ (`"04B3` `"04B2`), used in Abkhasian and Tajik.

Other characters have alternative glyphs, as specified in Unicode 1.0. These are:

- ҫ Ҫ, variant forms of ҫ Ҫ.
- ҙ Ҙ, variant forms of ҙ Ҙ.
- ҿ Ҿ, variant forms of ҿ Ҿ.
- ғ Ғ, variant forms of ғ Ғ.

All characters are given in uppercase/lowercase pairs, with the exception of character Ӏ (`"04C0`), which is case-invariant (this symbol denotes an aspiration sign, used in many Caucasian languages).

3 The UCY font family

UCY stands for *Unicode 'Computer-Modern-like' Cyrillic*. Each (virtual) font in this family contains 358 characters, including a complete set of punctuation marks and accents, so that typesetting any language using the Cyrillic alphabet becomes possible. It is the first step towards

[1] Code positions `"0460`–`"0486` are used for Old Cyrillic (Old Church Slavonic). A Slavonic/Paleocyrillic package is under preparation by the first author, and will be included in ScholarTEX, version 2.

Characters `"0462`, `"0463` (yat), `"0472`, `"0473` (fita), `"0474`, `"0475` (izhitsa), are used in pre-1917 Russian; `"046A`, `"046B` (yus) is used in pre-1945 Bulgarian. Hence, they are considered as 'modern' Cyrillic letters.

the *Unicode Computer Modern* font, which will contain all Latin, Greek, Cyrillic, IPA and similar characters.

The `ucyr10` font is a Roman style, 10 point font. As a virtual font, it is based on two real fonts: `wcyr10` and `ecyr10`.

The `wcyr10` font contains the punctuation marks, digits and other symbols, identical to those in `dcr10` (the DC family of fonts, based on Computer Modern and assembled/designed by Norbert Schwarz). The remaining characters are sufficient for European languages using the Cyrillic alphabet.

The `ecyr10` font contains those characters that are specific to Asian languages using the Cyrillic alphabet.

There are three sets of additional characters in UCY that are referred to in ISO 10646-1/ Unicode:

1. Accented vowels: these are necessary for dictionaries and textbooks. An accenting mechanism cannot be used —at least not in the present version of Ω— because, as in TEX, it would prohibit hyphenation. These vowels are encoded using two ISO 10646-1/Unicode positions: first the vowel, then `"0301` (non-spacing acute diacritical mark). For example: `"044F"0301` corresponds to я́.

2. Variant forms, given in Unicode, version 1.0 (cf. section 2). By changing the second Ω Translation Process, the user can specify which glyphs are to be used, and under what conditions.

3. Characters which the first author has detected in other sources and which are not mentioned in any version of the Unicode standard. These are і̡, І̡, ԯ, Ԯ, н̧, Н̧, ж̧ and Ж̧. The authors would be grateful for any comments on the use or origin of these characters, as well as for any proposals for additional characters or variant forms.

Appendix B shows the current version of the UCY font family table.

4 Description of some Ω Translation Processes used in Cyrillic Ω typesetting

Three Ω Translation Processes are presented as examples of the use of Ω for typesetting with the Cyrillic script. The reader may modify these according to his/her needs. Up to 2^{16} translation processes may run simultaneously.

4.1 Example of input Ω Translation Process

As mentioned in the introduction, the first Ω Translation Process is used to map arbitrary user input into the ISO 10646-1/Unicode character set. Here is an Ω Translation Process for the AMS Cyrillic transliteration, as defined by Barbara Beeton in *Mathematical symbols and Cyrillic fonts ready for distribution (revised)*, TUGboat **6** (1985) No. 3, pp. 124–128.

```
%%%
%%% AMS Cyrillic -> UNICODE
%%% (c) Yannis Haralambous & John Plaice, 1994
%%%
input: 1 ;
output: 2 ;
%%%
%%% the above two lines mean that the input uses 1-octet characters
```

```
%%% while the output uses 2-octet characters.
%%%
expressions:
`A'             =>   @"0410 ;
`a'             =>   @"0430 ;
`B'             =>   @"0411 ;
`b'             =>   @"0431 ;
`V'             =>   @"0412 ;
`v'             =>   @"0432 ;
`G'             =>   @"0413 ;
`g'             =>   @"0433 ;
`D'(`J'|`j'|`1')=>   @"0402 ;
`d'(`j'|`1')    =>   @"0452 ;
`D'`2'          =>   @"040F ;
`d'`2'          =>   @"045F ;
`D'`3'          =>   @"0405 ;
`d'`3'          =>   @"0455 ;
`D'             =>   @"0414 ;
`d'             =>   @"0434 ;
`E'`1'          =>   @"042D ;
`e'`1'          =>   @"044D ;
`E'`2'          =>   @"0404 ;
`e'`2'          =>   @"0454 ;
`E'             =>   @"0415 ;
`e'             =>   @"0435 ;
`Z'(`H'|`h'|`1')=>   @"0416 ;
`z'(`h'|`1')    =>   @"0436 ;
`Z'             =>   @"0417 ;
`z'             =>   @"0437 ;
`I'`1'          =>   @"0406 ;
`i'`1'          =>   @"0456 ;
`I'             =>   @"0418 ;
`i'             =>   @"0438 ;
`J'`1'          =>   @"042F ;
`j'`1'          =>   @"044F ;
`J'`2'          =>   @"042E ;
`j'`2'          =>   @"044E ;
`J'             =>   @"0408 ;
`j'             =>   @"0458 ;
`H'             =>   @"0425 ;
`K'(`H'|`h')    =>   @"0425 ;
`h'             =>   @"0445 ;
`k'`h'          =>   @"0445 ;
`K'             =>   @"041A ;
`k'             =>   @"043A ;
`L'(`J'|`j'|`1')=>   @"0409 ;
`l'(`j'|`1')    =>   @"0459 ;
`L'             =>   @"041B ;
`l'             =>   @"043B ;
`M'             =>   @"041C ;
`m'             =>   @"043C ;
`N'`0'          =>   @"2116 ; %%% the NUMERO symbol
`N'(`J'|`j'|`1')=>   @"040A ;
```

```
'n'('j'|'1')    =>   @"045A ;
'N'             =>   @"041D ;
'n'             =>   @"043D ;
'O'             =>   @"041E ;
'o'             =>   @"043E ;
'P''1'          =>   @"042C ;
'p''1'          =>   @"044C ;
'P''2'          =>   @"042A ;
'p''2'          =>   @"044A ;
'P'             =>   @"041F ;
'p'             =>   @"043F ;
'R'             =>   @"0420 ;
'r'             =>   @"0440 ;
'W'             =>   @"0429 ;
'S''H''C''H'    =>   @"0429 ;
'S''h''c''h'    =>   @"0429 ;
'w'             =>   @"0449 ;
's''h''c''h'    =>   @"0449 ;
'X'             =>   @"0428 ;
'S'('H'|'h')    =>   @"0428 ;
'x'             =>   @"0448 ;
's''h'          =>   @"0448 ;
'S'             =>   @"0421 ;
's'             =>   @"0441 ;
'T'('S'|'s')    =>   @"0426 ;
't''s'          =>   @"0446 ;
'T'             =>   @"0422 ;
't'             =>   @"0442 ;
'Q'             =>   @"0427 ;
'C'('H'|'h')    =>   @"0427 ;
'q'             =>   @"0447 ;
'c''h'          =>   @"0447 ;
'C''1'          =>   @"040B ;
'c''1'          =>   @"045B ;
'C'             =>   @"0426 ;
'c'             =>   @"0446 ;
'U'             =>   @"0423 ;
'u'             =>   @"0443 ;
'F'             =>   @"0424 ;
'f'             =>   @"0444 ;
'Y'('U'|'u')    =>   @"042E ;
'y''u'          =>   @"044E ;
'Y'('A'|'a')    =>   @"042F ;
'y''a'          =>   @"044F ;
'Y'             =>   @"042B ;
'y'             =>   @"044B ;
@"E0            =>   @"0403 ; %%% \'G
@"E1            =>   @"0453 ; %%% \'g
@"E2            =>   @"0401 ; %%% \"E
@"E3            =>   @"0451 ; %%% \"e
@"E4            =>   @"0404 ; %%% \=E
@"E5            =>   @"0454 ; %%% \=e
@"E6            =>   @"0406 ; %%% \=I
```

```
@"E7           =>   @"0456 ; %%% \=\i
@"E8           =>   @"0407 ; %%% \"I
@"E9           =>   @"0457 ; %%% \"\i
@"EA           =>   @"0419 ; %%% \u{I}
@"EB           =>   @"0439 ; %%% \u\i
@"EC           =>   @"040B ; %%% \'C
@"ED           =>   @"045B ; %%% \'c
@"EE           =>   @"040C ; %%% \'K
@"EF           =>   @"045C ; %%% \'k
@"F0           =>   @"040E ; %%% \u{U}
@"F1           =>   @"045E ; %%% \u{u}
@"F2           =>   @"040F ; %%% \Dzh
@"F3           =>   @"045F ; %%% \dzh
@"F4           =>   @"042A ; %%% \Cdprime
@"F5           =>   @"044A ; %%% \cdprime
@"F6           =>   @"042C ; %%% \Cprime
@"F7           =>   @"044C ; %%% \cprime
@"F8           =>   @"042D ; %%% \'E
@"F9           =>   @"044D ; %%% \'e
@"FA           =>   @"0405 ; %%% \Dz
@"FB           =>   @"0455 ; %%% \dz
%%%
```

The macros used in the AMS input scheme are redefined as follows.

```
\def\'#1{\ifx#1G^^e0\else\ifx#1g^^e1\else\ifx#1C^^ec\else\ifx#1c^^ed%
\else\ifx#1K^^ee\else\ifx#1k^^ef\else\ifx#1E^^f8\else\ifx#1e^^f9%
\else\accent'001#1\fi\fi\fi\fi\fi\fi\fi\fi}
\def\"#1{\ifx#1E^^e2\else\ifx#1e^^e3\else\ifx#1I^^ec\else\ifx#1\i^^ed%
\else\accent'004#1\fi\fi\fi\fi}
\def\=#1{\ifx#1E^^e4\else\ifx#1e^^e5\else\ifx#1I^^e6\else\ifx#1\i^^e7%
\else\accent'011#1\fi\fi\fi\fi}
\def\u#1{\ifx#1I^^ea\else\ifx#1\i^^eb\else\ifx#1U^^f0\else\ifx#1u^^f1%
\else\accent'010#1\fi\fi\fi\fi}
\def\Dzh{^^f2}\def\dzh{^^f3}
\def\Cdprime{^^f4}\def\cdprime{^^f5}
\def\Cprime{^^f6}\def\cprime{^^f7}
\def\Dz{^^fa}\def\dz{^^fb}
```

4.2 Example of an `\uppercase` replacement Ω Translation Process

Once the translation to ISO 10646-1/Unicode has taken place, the 'any letter → uppercase letter' correspondence is well-defined and unique. Here is the Cyrillic part of this Ω Translation Process.

```
%%%
%%% UNICODE (Cyrillic part) -> uppercase UNICODE
%%% (c) Yannis Haralambous & John Plaice, 1994
%%%
input: 2 ;
output: 2 ;
%%%
%%% the above two lines mean that both input and output
```

```
%%% use in 2-octet characters.
%%%
expressions:
@"0401-@"042F   =>   \1 ;
@"0430-@"044F   =>   #(\1 - @"0020) ;
@"0451-@"045F   =>   #(\1 - @"0040) ;
(@"0462 | @"046A | @"0472 | @"0474 |
 @"0490 | @"0492 | @"0494 | @"0496 | @"0498 | @"049A | @"049C | @"049E |
 @"04A0 | @"04A2 | @"04A4 | @"04A6 | @"04A8 | @"04AA | @"04AC | @"04AE |
 @"04B0 | @"04B2 | @"04B4 | @"04B6 | @"04B8 | @"04BA | @"04BC | @"04BE |
 @"04C0 | @"04C1 | @"04C3 | @"04C5 | @"04C7 | @"04C9 | @"04CB |
 @"04D0 | @"04D2 | @"04D4 | @"04D6 | @"04D8 | @"04DA | @"04DC | @"04DE |
 @"04E0 | @"04E2 | @"04E4 | @"04E6 | @"04E8 | @"04EA | @"04EE |
 @"04F0 | @"04F2 | @"04F4 | @"04F8)   =>   \1 ;
(@"0463 | @"046B | @"0473 | @"0475 |
 @"0491 | @"0493 | @"0495 | @"0497 | @"0499 | @"049B | @"049D | @"049F |
 @"04A1 | @"04A3 | @"04A5 | @"04A7 | @"04A9 | @"04AB | @"04AD | @"04AF |
 @"04B1 | @"04B3 | @"04B5 | @"04B7 | @"04B9 | @"04BB | @"04BD | @"04BF |
 @"04C2 | @"04C4 | @"04C6 | @"04C8 | @"04CA | @"04CC |
 @"04D1 | @"04D3 | @"04D5 | @"04D7 | @"04D9 | @"04DB | @"04DD | @"04DF |
 @"04E1 | @"04E3 | @"04E5 | @"04E7 | @"04E9 | @"04EB | @"04EF |
 @"04F1 | @"04F3 | @"04F5 | @"04F9)   =>   #(\1 - @"0001) ;
%%%
```

4.3 Example of an output encoding Ω Translation Process

The last example is the Ω Translation Process to go from ISO 10646-1/Unicode to UCY font family encoding. To simplify the code, tables are used. See the Ω manual for a complete description of the syntax.

```
%%%
%%% UNICODE (Cyrillic part) -> UCY Font Family Encoding
%%% (c) Yannis Haralambous & John Plaice, 1994
%%%
input: 2 ;
output: 2 ;
%%%
%%% the above two lines mean that both input and output
%%% use 2-octet characters.
%%%
tables:
cyrillic_unicode_to_ucy[48] = {
%%% ==================================================================
%%% 0400 - 040F
%%% ==================================================================
@"00FF, @"0085, @"0082, @"0083, @"0087, @"008A, @"008C, @"008E,
@"004A, @"0091, @"0092, @"0094, @"0095, @"00FF, @"0097, @"0098,
%%% ==================================================================
%%% 0410 - 041F
%%% ==================================================================
@"0041, @"0042, @"0056, @"0047, @"0044, @"0045, @"0089, @"005A,
@"0049, @"0090, @"004B, @"004C, @"004D, @"004E, @"004F, @"0050,
%%% ==================================================================
```

```
%%% 0420 - 042F
%%% ==================================================================
@"005A, @"005B, @"005C, @"005D, @"0046, @"0048, @"0043, @"0051,
@"0058, @"0057, @"0099, @"0059, @"009B, @"009E, @"00C0, @"00C2} ;
old_cyrillic_unicode_to_ucy[32] = {
%%% ==================================================================
%%% 0460 - 046F
%%% ==================================================================
@"00FF, @"00FF, @"009C, @"00BC, @"00FF, @"00FF, @"00FF, @"00FF,
@"00FF, @"00FF, @"00C4, @"00E4, @"00FF, @"00FF, @"00FF, @"00FF,
%%% ==================================================================
%%% 0470 - 047F
%%% ==================================================================
@"00FF, @"00FF, @"00C6, @"00E6, @"00C7, @"00E7, @"00FF, @"00FF,
@"00FF, @"00FF, @"00FF, @"00FF, @"00FF, @"00FF, @"00FF, @"00FF} ;
accented_vowels_to_ucy[48] = {
%%% ==================================================================
%%% 0400 - 040F
%%% ==================================================================
@"00FF, @"0086, @"00C9, @"0084, @"0088, @"008A, @"008D, @"008F,
@"004A, @"0091, @"0092, @"0094, @"0095, @"00FF, @"0097, @"0098,
%%% ==================================================================
%%% 0410 - 041F
%%% ==================================================================
@"0080, @"0042, @"0056, @"0083, @"0044, @"0084, @"0089, @"005A,
@"008B, @"0090, @"0095, @"004C, @"004D, @"004E, @"0093, @"0050,
%%% ==================================================================
%%% 0420 - 042F
%%% ==================================================================
@"005A, @"005B, @"005C, @"0096, @"0046, @"0048, @"0043, @"0051,
@"0058, @"0057, @"0099, @"009A, @"009B, @"009F, @"00C9, @"00CB} ;
old_accented_vowels_to_ucy[32] = {
%%% ==================================================================
%%% 0460 - 046F
%%% ==================================================================
@"00FF, @"00FF, @"009D, @"00BD, @"00FF, @"00FF, @"00FF, @"00FF,
@"00FF, @"00FF, @"00C5, @"00E5, @"00FF, @"00FF, @"00FF, @"00FF,
%%% ==================================================================
%%% 0470 - 047F
%%% ==================================================================
@"00FF, @"00FF, @"00C6, @"00E6, @"00C8, @"00E8, @"00FF, @"00FF,
@"00FF, @"00FF, @"00FF, @"00FF, @"00FF, @"00FF, @"00FF, @"00FF} ;
expressions:
@"0020-@"0040   =>   \1 ; %%% <space> - @
@"005B-@"0060   =>   \1 ; %%% [ - ‘
@"007B-@"007E   =>   \1 ; %%% { - ~
@"00A3   =>   @"001C ; %%% \pounds
@"00A7   =>   @"001B ; %%% \S
@"00AB   =>   @"0013 ; %%% <<
@"00BB   =>   @"0014 ; %%% >>
@"0131   =>   @"0019 ; %%% \i
(@"0400-@"042F) @"0301   =>   #(accented_vowels_to_ucy[\1 - @"0400]) ;
(@"0430-@"044F) @"0301   =>   #(accented_vowels_to_ucy[\1 - @"0420] + @"0020) ;
```

```
(@"0450-@"045F) @"0301   =>   #(accented_vowels_to_ucy[\1 - @"0450] + @"0020) ;
(@"0460-@"047F) @"0301   =>   #(old_accented_vowels_to_ucy[\1 - @"0460]) ;
@"0300-@"0303   =>   #(\1 - @"0300) ; %%% \`,\',\^
@"0304   =>   @"0009 ; %%% \=
@"0306   =>   @"0008 ; %%% \u
@"0307   =>   @"000A ; %%% \.
@"0308   =>   @"0004 ; %%% \"
@"030A   =>   @"0006 ; %%% \r
@"030B   =>   @"0005 ; %%% \H
@"030C   =>   @"0007 ; %%% \v
@"0327-@"0328   =>   #(\1 - @"031C) ; %%% \c,\k
@"200B   =>   @"0017 ; %%% cwm
@"2010   =>   @"007F ; %%% hyphen
@"2013-@"2014   =>   #(\1 - @"1FFE) ; %%% --,---
@"201A   =>   @"000D ; %%% , (quote)
@"201C-@"201E   =>   #(\1 - @"200C) ; %%% ``,'', ,,
@"2039-@"203A   =>   #(\1 - @"202B) ; %%% <,> (guillemets)
@"2080   =>   @"0018 ; %%% little 0
@"2116   =>   @"001E ; %%% the NUMERO symbol
@"0400-@"042F   =>   #(cyrillic_unicode_to_ucy[\1 - @"0400]) ;
@"0430-@"044F   =>   #(cyrillic_unicode_to_ucy[\1 - @"0420] + @"0020) ;
@"0450-@"045F   =>   #(cyrillic_unicode_to_ucy[\1 - @"0450] + @"0020) ;
@"0460-@"047F   =>   #(old_cyrillic_unicode_to_ucy[\1 - @"0460]) ;
@"0490   =>   @"0081 ; %%% Ukrainian G
@"0491   =>   @"00A1 ; %%% Ukrainian g
@"04C0   =>   @"008C ; %%% Caucasian I
@"0492-@"04F9   =>   #(\1 - @"0400) ;
@"0041-@"FFFD   =>   @"00FF ; %%% the rest of UNICODE
```

In this Ω Translation Process, combinations of ISO 10646-1/Unicode codes *letter* + `@"0301` (acute accent) are mapped to:

- the accented version of *letter* if it is a vowel, к, К, г or Г;
- the *letter* itself in all other cases.

5 Aknowledgements and Availability

All software mentioned in this paper is in the public domain or under GNU copyleft.

At the time of writing (Aug. 1994), Ω is still being β-tested. Please contact the authors for more information.

The Cyrillic glyphs shown in this paper were generated with METAFONT code by Donald Knuth, Norbert Schwarz, Thomas Ridgeway, Jörg Knappen and the first author. In a forthcoming version of the Ω Cyrillic package they will be replaced by glyphs extracted or derived from the esthetically more satisfying Computer-Modern-like Cyrillic fonts distributed by the Cyrillic TEX Users Group.

It is also planned to develop a mechanism to create the virtual font `ucy***.xvp` from the `pl` files of fonts `wcy***.pl` and `ecy***.pl`. See the `README` file of the package for more information.

The Ω Cyrillic package, containing the METAFONT and `xvp` code for these fonts, as well as the Ω Translation Process files, the text of this paper, and hyphenation patterns for Russian, Bulgarian (β), and Croatian written in the Cyrillic alphabet, can be obtained on the Internet with the following `ftp` operation:

```
ftp ftp.ens.fr
login anonymous
passwd your-name@your-site
cd /pub/tex/yannis/omega/cyrillic
bin
get cyrillic-omega.tar.gz
quit
```

The authors would be grateful for any comments or suggestions. In particular, it would be very helpful for the rest of the Ω users' community if readers having written their own OTP files would make them generously available for inclusion in the package.

6 Future plans

Similar packages for Latin alphabet languages (including Vietnamese), IPA, Greek and symbols/dingbats are being prepared. The next step will be to combine their virtual fonts into one big virtual font. Doing this will (a) eliminate redundant glyphs and (b) enable kerning between characters from different alphabets. At least five popular font families already have sufficient characters and will only need small additions to fit into this scheme:

1. Computer Modern (METAFONT);
2. Times Roman (PostScript);
3. Helvetica (PostScript);
4. Courier (PostScript);
5. Lucida Bright (PostScript).

Typesetting languages using the Cyrillic alphabet requires far more than just fonts: hyphenation patterns, Babel styles, documentation on typographical conventions, alphabetical orders, etc. are all necessary. To the best of the authors' knowledge, this information is only available for Russian (and partly for Bulgarian). If the reader feels he/she can contribute in this direction, please contact the Technical Working Group on Multiple Language Coordination at the first author's address.

A Modern Cyrillic Part of ISO 10646-1/Unicode

	0	1	2	3	4	5	6	7
"040x		Ё	Ђ	Ѓ	Є	Ѕ	І	Ї
	Ј	Љ	Њ	Ћ	Ќ		Ў	Џ
"041x	А	Б	В	Г	Д	Е	Ж	З
	И	Й	К	Л	М	Н	О	П
"042x	Р	С	Т	У	Ф	Х	Ц	Ч
	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
"043x	а	б	в	г	д	е	ж	з
	и	й	к	л	м	н	о	п
"044x	р	с	т	у	ф	х	ц	ч
	ш	щ	ъ	ы	ь	э	ю	я
"045x		ё	ђ	ѓ	є	ѕ	і	ї
	ј	љ	њ	ћ	ќ		ў	џ
"046x			Ѣ	ѣ				
			Ѫ	ѫ				
"047x			Ѳ	ѳ	Ѵ	ѵ		
"049x	Ґ	ґ	Ғ	ғ	Ҕ	ҕ	Җ	җ
	Ҙ	ҙ	Қ	қ	Ҝ	ҝ	Ҟ	ҟ
"04Ax	Ҡ	ҡ	Ң	ң	Ҥ	ҥ	Ҧ	ҧ
	Ҩ	ҩ	Ҫ	ҫ	Ҭ	ҭ	Ү	ү
"04Bx	Ұ	ұ	Ҳ	ҳ	Ҵ	ҵ	Ҷ	ҷ
	Ҹ	ҹ	Һ	һ	Ҽ	ҽ	Ҿ	ҿ
"04Cx	Ӏ	Ӂ	ӂ	Ӄ	ӄ			Ӈ
	ӈ			Ӌ	ӌ			
"04Dx	Ӑ	ӑ	Ӓ	ӓ	Ӕ	ӕ	Ӗ	ӗ
	Ә	ә	Ӛ	ӛ	Ӝ	ӝ	Ӟ	ӟ
"04Ex	Ӡ	ӡ	Ӣ	ӣ	Ӥ	ӥ	Ӧ	ӧ
	Ө	ө	Ӫ	ӫ			Ӯ	ӯ
"04Fx	Ӱ	ӱ	Ӳ	ӳ	Ӵ	ӵ		
	Ӹ	ӹ						
	8	9	A	B	C	D	E	F

B Table of the UCY font family

	0	1	2	3	4	5	6	7
"000x	`	´	ˆ	˜	¨	˝	˚	ˇ
	˘	¯	˙	¸	˛	‚	‹	›
"001x	“	”	„	«	»	–	—	
	0	ı	ȷ	§	£		№	˘
"002x	␣	!	"	#	$	%	&	’
	(	)	*	+	,	-	.	/
"003x	0	1	2	3	4	5	6	7
	8	9	:	;	<	=	>	?
"004x	@	А	Б	Ц	Д	Е	Ф	Г
	Х	И	Ј	К	Л	М	Н	О
"005x	П	Ч	Р	С	Т	У	В	Щ
	Ш	Ы	З	[	\	]	ˆ	_
"006x	‘	а	б	ц	д	е	ф	г
	х	и	ј	к	л	м	н	о
"007x	п	ч	р	с	т	у	в	щ
	ш	ы	з	{	|	}	˜	-
"008x	А́	Ґ	Ђ	Ѓ	Е́	Ё	Ё́	Є
	Є́	Ж	Ѕ	И́	І	І́	Ї	Ї́
"009x	Й̈	Љ	Њ	О́	Ћ	Ќ	У́	Ў̈
	Џ	Ъ	Ы́	Ь	Ѣ	Ѣ́	Э	Э́
"00Ax	а́	ґ	ђ	ѓ	е́	ё	ё́	є
	є́	ж	ѕ	и́	і	і́	ї	ї́
"00Bx	й̈	љ	њ	о́	ћ	ќ	у́	ў̈
	џ	ъ	ы́	ь	ѣ	ѣ́	э	э́
"00Cx	Ю	Ю́	Я	Я́	Ѫ	Ѫ́	Ѳ	Ѵ
	Ѵ́	Ђ́						
"00Dx								
"00Ex	ю	ю́	я	я́	ѫ	ѫ́	ѳ	ѵ
	ѵ́	ђ́						
"00Fx								
"018x	Į	į	Ӆ	ӆ	Ң	ң	Җ	җ
	Ҙ	ҙ	Ҫ	ҫ	Ҿ	ҿ	Ғ	ғ
"019x			Ғ	ғ	Ҕ	ҕ	Җ	җ
	Ҙ	ҙ	Қ	қ	Ҝ	ҝ	Ҟ	ҟ
"01Ax	Ҡ	ҡ	Ң	ң	Ҥ	ҥ	Ҧ	ҧ
	Ҩ	ҩ	Ҫ	ҫ	Ҭ	ҭ	Ү	ү
"01Bx	Ұ	ұ	Ҳ	ҳ	Ҵ	ҵ	Ҷ	ҷ
	Ҹ	ҹ	Һ	һ	Ҽ	ҽ	Ҿ	ҿ
"01Cx		Ӂ	ӂ	Ӄ	ӄ	Қ	қ	Ӈ
	ӈ	Ӽ	ӽ	Ӌ	ӌ			
"01Dx	Ӑ	ӑ	Ӓ	ӓ	Ӕ	ӕ	Ӗ	ӗ
	Ә	ә	Ӛ	ӛ	Ӝ	ӝ	Ӟ	ӟ
"01Ex	Ӡ	ӡ	Ӣ	ӣ	Ӥ	ӥ	Ӧ	ӧ
	Ө	ө	Ӫ	ӫ			Ӯ	ӯ
"01Fx	Ӱ	ӱ	Ӳ	ӳ	Ӵ	ӵ		
	Ӹ	ӹ						
	8	9	A	B	C	D	E	F