Typesetting CJK Languages with Ω

Jin-Hwan Cho
Korean TEX Users Group
chofchof@ktug.or.kr

Haruhiko Okumura
Mie University
Faculty of Education
514-8507
Japan
okumura@acm.org

Abstract

This paper describes how to typeset Chinese, Japanese, and Korean (CJK) languages with Omega, a 16-bit extension of Donald Knuth's TEX. In principle, Omega has no difficulty in typesetting those East Asian languages because of its internal representation using 16-bit Unicode. However, it has not been widely used in practice because of the difficulties in adapting it to CJK typesetting rules and fonts, which we will discuss in the paper.

1 Introduction

Chinese, Japanese, and Korean (CJK) languages are characterized by multibyte characters covering more than 60% of Unicode. The huge number of characters prevented the original 8-bit TEX from working smoothly with CJK languages. There have been three methods for supporting CJK languages in the TEX world up to now.

The first method, called the *subfont scheme*, splits CJK characters into sets of 256 characters or fewer, the number of characters that a TEX font metric file can accommodate. Its main advantage lies in using 8-bit TEX systems directly. However, one document may contain dozens of subfonts for each CJK font, and it is quite hard to insert glue and kerns between characters of different subfonts, even those from the same CJK font. Moreover, without the help of a DVI driver (e.g., DVIPDFM*x* [2]) supporting the subfont scheme, it is not possible to generate PDF documents containing CJK characters that can be extracted or searched. Many packages are based on this method; for instance, CJK-LATEX[1] by Werner Lemberg, HLATEX[2] by Koaunghi Un, and the Chinese module in ConTEXt[3] by Hans Hagen.

On the other hand, in Japan, the most widely used TEX-based system is pTEX [1] (formerly known as ASCII Nihongo TEX), a 16-bit extension of TEX localized to the Japanese language. It is designed for high-quality Japanese book publishing (the "p" of pTEX stands for publishing; the name jTEX was used by another system). pTEX can handle multibyte characters natively (i.e., without resorting to subfonts), and it can typeset both horizontally and vertically within a document. It is upward compatible[4] with TEX, so it can be used to typeset both Japanese and Latin languages, but it cannot handle Chinese and Korean languages straightforwardly. pTEX supports three widely-used Japanese encodings, JIS (ISO-2022-JP), Shift JIS, and EUC-JP, but not Unicode-based encodings such as UTF-8.

The third route, Omega [3], is also a 16-bit extension of TEX, having 16-bit Unicode as its internal representation. In principle, Omega is free from the limitations mentioned above, but thus far there is no thorough treatment of how it can be used for professional CJK typesetting and how to adapt it to popular CJK font formats such as TrueType and OpenType. We set out to fill in this blank.

2 CJK Typesetting Characteristics

Each European language has its own hyphenation rules, but their typesetting characteristics are overall fairly similar. CJK languages differ from European languages in that there are no hyphenation

[1] Avaliable on CTAN as `language/chinese/CJK/`
[2] Available on CTAN as `language/korean/HLaTeX/`
[3] Available on CTAN as `macros/context/`

[4] Although pTEX doesn't actually pass the `trip` test, it is thought to be upward compatible with TEX in virtually all practical situations.

rules. All CJK languages allow line breaking almost anywhere, without a hyphen. This characteristic is usually implemented by inserting appropriate glues between CJK characters.

One fine point is the treatment of blank spaces and end-of-line (EOL) characters. Korean uses blank spaces to separate words, but Chinese and Japanese rarely use blank spaces. An EOL character is converted in TEX to a blank space and then to a skip, which is unnecessary for Chinese and Japanese typesetting. To overcome this problem, pTEX ignores an EOL when it follows a CJK character.

Moreover, whereas Korean uses Latin punctuation marks (periods, commas, etc.), Chinese and Japanese use their own punctuation symbols. These CJK punctuation symbols need to be treated somewhat differently from ordinary characters. The appropriate rules are described in this paper.

3 CJK Omega Translation Process

We introduce here the CJK Omega Translation Process (CJK-ΩTP)[5] developed by the authors to implement the CJK typesetting characteristics mentioned above.

An Omega Translation Process (ΩTP) is a powerful preprocessor, which allows text to be passed through any number of finite state automata, which can achieve many different effects. Usually it is quite hard or impossible to do the same work with other TEX-based systems.

For each CJK language, the CJK-ΩTP is divided into two parts. The first ΩTP (boundCJK.otp) is common to all CJK languages, and controls the boundaries of blocks consisting of CJK characters and blank spaces. The second ΩTP (one of interCHN.otp, interJPN.otp, and interKOR.otp) is specific to each language, and controls typesetting rules for consecutive CJK characters.

4 Common Typesetting Characteristics

The first task of boundCJK.otp is to split the input stream into CJK blocks and non-CJK blocks, and insert glue (`\boundCJKglue`) in between to allow line breaking.

However, combinations involving some Latin and CJK symbols (quotation marks, commas, periods, etc.), do not allow line breaking. In this case, `\boundCJKglue` is not inserted so that the original line breaking rule is applied. This corresponds to pTEX's primitives `\xspcode` and `\inhibitxspcode`.

boundCJK.otp defines seven character sets; the role of each set is as follows.

1. `{CJK}` is the set of all CJK characters; its complement is denoted by `^{CJK}`.
2. `{XSPCODE1}` (e.g., `([{‘`) is the subset of `^{CJK}` such that `\boundCJKglue` is inserted only between `{CJK}` and `{XSPCODE1}` in this order.
3. `{XSPCODE2}` (e.g., `)]}’;,.`) is the subset of `^{CJK}` such that `\boundCJKglue` is inserted only between `{XSPCODE2}` and `{CJK}` in this order.
4. `{XSPCODE3}` (e.g., `0-9 A-Z a-z`) is the subset of `^{CJK}` such that `\boundCJKglue` is inserted between `{CJK}` and `{XSPCODE3}`, irrespective of the order.
5. `{INHIBITXSPCODE0}` (e.g., ―∽…¥) is the subset of `{CJK}` *not* allowing `\boundCJKglue` between `{INHIBITXSPCODE0}` and `^{CJK}`, irrespective of the order.
6. `{INHIBITXSPCODE1}` (e.g., 、。〉》」』】］, CJK right parentheses and periods) is the subset of `{CJK}` *not* allowing `\boundCJKglue` between `^{CJK}` and `{INHIBITXSPCODE1}` in this order.
7. `{INHIBITXSPCODE2}` (e.g., 〈《「『【［〖〔, CJK left parentheses) is the subset of `{CJK}` *not* allowing `\boundCJKglue` in between `{INHIBITXSPCODE2}` and `^{CJK}` in this order.

The second task of boundCJK.otp is to enclose each CJK block in a group ‘`{\selectCJKfont␣...}`’, and convert all blank spaces inside the block to the command `\CJKspace`.

The command `\selectCJKfont` switches to the appropriate CJK font, and `\CJKspace` is defined to be either a `\space` (for Korean) or `\relax` (for Chinese and Japanese) according to the selected language.

Note that if the input stream starts with blank spaces followed by a CJK block or ends with a CJK block followed by blank spaces, then these spaces must be preserved regardless of the language, because of math mode:

```
{{CJK} {SPACE} $...$ {SPACE} CJK}}
```

and restricted horizontal mode:

```
\hbox{{SPACE} {CJK} {SPACE}}
```

5 Language-dependent Characteristics

The line breaking mechanism is common to all of the language-dependent ΩTPs (interCHN.otp, interJPN.otp, and interKOR.otp). The glue `\interCJKglue` is inserted between consecutive CJK characters, and its role is similar to the glue `\boundCJKglue` at the boundary of a CJK block.

[5] Available at `http://project.ktug.or.kr/omega-cjk/`

Some combinations of CJK characters do not allow line breaking. This is implemented by simply inserting a `\penalty` 10000 before the relevant `\interCJKglue`. In the case of boundCJK.otp, however, no `\boundCJKglue` is inserted where line breaking is inhibited.

The CJK characters not allowing line breaking are defined by the following two classes in interKOR.otp for Korean typesetting.

1. `{CJK_FORBIDDEN_AFTER}` does not allow line breaking between `{CJK_FORBIDDEN_AFTER}` and `{CJK}` in this order.
2. `{CJK_FORBIDDEN_BEFORE}` does not allow line breaking in between `{CJK}` and `{CJK_FORBIDDEN_BEFORE}` in this order.

On the other hand, interJPN.otp defines six classes for Japanese typesetting, as discussed in the next section.

6 Japanese Typesetting Characteristics

Most Japanese characters are designed on a square 'canvas'. pTEX introduced a new length unit, `zw` (for *zenkaku* width, or full-width), denoting the width of this canvas. The CJK-ΩTP defines `\zw` to denote the same quantity.

For horizontal (left-to-right) typesetting mode, the baseline of a Japanese character typically divides the square canvas by 0.88 : 0.12. If Japanese and Latin fonts are typeset with the same size, Japanese fonts appear larger. In the sample shown in Figure 1, Japanese characters are typeset 92.469 percent the size of Latin characters, so that 10 pt (1 in = 72.27 pt) Latin characters are mixed with 3.25 mm (= 13 Q; 4 Q = 1 mm) Japanese characters. Also, Japanese and Latin words are traditionally separated by about 0.25 zw, though this space is getting smaller nowadays.

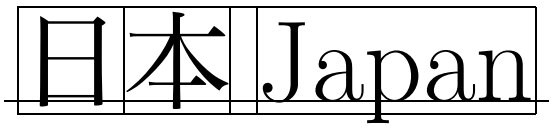

Figure 1: The width of an ordinary Japanese character, 1 zw, is set to 92.469% the design size of the Latin font, and a gap of 0.25 zw is inserted. The baseline is set to 0.12 zw above the bottom of the enclosing squares.

Some characters (such as punctuation marks and parentheses) are designed on a half-width canvas: its width is 0.5 zw. For ease of implementation, actual glyphs may be designed on square canvases. We can use the virtual font mechanism to map the logical shape and the actual implementation.

interJPN.otp divides Japanese characters into six classes:

1. Left parentheses: ‘ “ (〔 [{ 〈 《 「 『 【
   Half width, may be designed on square canvases flush right. In that case we ignore the left half and pretend they are half-width, e.g., `\hbox to 0.5zw{\hss}`. If a class-1 character is followed by a class-3 character, then an `\hskip 0.25zw minus 0.25zw` is inserted in between.
2. Right parentheses: 、 , ’ ”) 〕] } 〉 》 」 』 】
   Half width, may be designed flush left on square canvases. If a class-2 character is followed by a class-0, -1, or -5 character, then an `\hskip 0.5zw minus 0.5zw` is inserted in between. If a class-2 character is followed by a class-3 character, then a `\hskip 0.25zw minus 0.25zw` is inserted in between.
3. Centered points: ・ : ;
   Half width, may be designed centered on square canvases. If a class-3 character is followed by a class-0, -1, -2, -4, or -5 character, then an `\hskip 0.25zw minus 0.25zw` is inserted in between. If a class-3 character is followed by a class-3 character, then an `\hskip 0.5zw minus 0.25zw` is inserted in between.
4. Periods: 。 .
   Half width, may be designed flush left on square canvases. If a class-4 character is followed by a class-0, -1, or -5 character, then an `\hskip 0.5zw` is inserted in between. If a class-4 character is followed by a class-3 character, then an `\hskip 0.75zw minus 0.25zw` is inserted in between.
5. Leaders: —‥…
   Full width. If a class-5 character is followed by a class-1 character, then an `\hskip 0.5zw minus 0.5zw` is inserted in between. If a class-5 character is followed by a class-3 character, then an `\hskip 0.25zw minus 0.25zw` is inserted in between. If a class-5 character is followed by a class-5 character, then a `\kern 0zw` is inserted in between.
0. Class-0: everything else.
   Full width. If a class-0 character is followed by a class-1 character, then an `\hskip 0.5zw minus 0.5zw` is inserted in between. If a class-0 character is followed by a class-3 character, then an `\hskip 0.25zw minus 0.25zw` is inserted in between.

Chinese texts can be typeset mostly with the same rules. An exception is the comma and the

period of Traditional Chinese. These two letters are designed at the center of the square canvas, so they should be treated as Class-3 characters.

7 Example: Japanese and Korean

Let us discuss how to use CJK-ΩTP in a practical situation. Figure 2 shows sample output containing both Japanese and Korean characters, which is typeset by Omega with the CJK-ΩTP and then processed by DVIPDFMx.

TEX はスタンフォード大学のクヌース教授によって開発された組版システムであり、組版の美しさと強力なマクロ機能を特徴としている。

TEX은 스탠포드 大學의 크누스 敎授에 의해 開發된 組版 시스템으로, 組版의 美와 强力한 매크로 機能이 特徵이다.

Figure 2: Sample CJK-ΩTP output.

The source of the sample above was prepared with the text editor Vim as shown in Figure 3. Here, the UTF-8 encoding was used to see Japanese and Korean characters at the same time. Note that the backslash character (\) is replaced with the yen currency symbol in Japanese fonts.

```
¥input omega-cjk-sample

¥hsize=75mm ¥parindent=¥zw

{¥japanese
¥TeXはスタンフォード大学のクヌース教授によって
開発された組版システムであり、組版の美しさと強
力なマクロ機能を特徴としている。
}
¥par¥vskip 10pt
{¥korean
¥TeX은 스탠포드 大學의 크누스 敎授에 의해 開發
된 組版 시스템으로, 組版의 美와 强力한 매크로
機能이 特徵이다.
}

¥bye
```

Figure 3: Sample CJK-ΩTP source.

The first line in Figure 3 calls another TEX file omega-cjk-sample.tex which starts with the following code, which loads[6] the CJK-ΩTP.

```
\ocp\OCPindefault=inutf8
\ocp\OCPboundCJK=boundCJK
\ocp\OCPinterJPN=interJPN
\ocp\OCPinterKOR=interKOR
```

[6] Omega requires the binary form of ΩTP files compiled by the utility otp2ocp included in the Omega distribution.

Note that inutf8.otp has to be loaded first to convert the input stream encoded with UTF-8 to UCS-2, the 16-bit Unicode.

```
\ocplist\CJKOCP=
  \addafterocplist 1 \OCPboundCJK
  \addafterocplist 1 \OCPindefault
  \nullocplist
\ocplist\JapaneseOCP=
  \addbeforeocplist 2 \OCPinterJPN \CJKOCP
\ocplist\KoreanOCP=
  \addbeforeocplist 2 \OCPinterKOR \CJKOCP
```

The glues \boundCJKglue and \interCJKglue for CJK line breaking mechanism are defined by new skip registers to be changed later according to the language selected.

```
\newskip\boundCJKskip % defined later
\def\boundCJKglue{\hskip\boundCJKskip}
\newskip\interCJKskip % defined later
\def\interCJKglue{\hskip\interCJKskip}
```

Japanese typesetting requires more definitions to support the six classes defined in interJPN.otp.

```
\newdimen\zw \zw=0.92469em
\def\halfCJKmidbox#1{\leavevmode%
  \hbox to .5\zw{\hss #1\hss}}
\def\halfCJKleftbox#1{\leavevmode%
  \hbox to .5\zw{#1\hss}}
\def\halfCJKrightbox#1{\leavevmode%
  \hbox to .5\zw{\hss #1}}
```

Finally, we need the commands \japanese and \korean to select the given language. These commands have to include actual manipulation of fonts, glues, and spaces.

```
\font\defaultJPNfont=omrml
\def\japanese{%
  \clearocplists\pushocplist\JapaneseOCP
  \let\selectCJKfont\defaultJPNfont
  \let\CJKspace\relax % remove spaces
  \boundCJKskip=.25em plus .15em minus .06em
  \interCJKskip=0em plus .1em minus .01em
}
\font\defaultKORfont=omhysm
\def\korean{%
  \clearocplists\pushocplist\KoreanOCP
  \let\selectCJKfont\defaultKORfont
  \let\CJKspace\space % preserve spaces
  \boundCJKskip=0em plus .02em minus .01em
  \interCJKskip=0em plus .02em minus .01em
}
```

It is straightforward to extend these macros to create a LATEX (Lambda) class file.

8 CJK Font Manipulation

At first glance, the best font for Omega seems to be the one containing all characters defined in 16-bit Unicode. In fact, such a font cannot be constructed.

There are several varieties of Chinese letters: Traditional letters are used in Taiwan and Korea, while simplified letters are now used in mainland China. Japan has its own somewhat simplified set. The glyphs are significantly different from country to country.

Unicode unifies these four varieties of Chinese letters into one, if they look similar. They are *not* identical, however. For example, the letter 'bone' has the Unicode point `9AA8`, but the top part of the Chinese Simplified letter and the Japanese letter are almost mirror images of each other, as shown in Figure 4. Less significant differences are also distracting to native Asian readers. The only way to overcome this problem is to use different CJK fonts according to the language selected.

(a) Chinese Simplified (b) Japanese

Figure 4: Two letters with the same Unicode point.

OpenType (including TrueType) is the most popular font format for CJK fonts. However, it is neither easy nor simple, even for TEX experts, to generate OFM and OVF files from OpenType fonts.

The situation looks simple for Japanese and Chinese fonts, having fixed width, because one (virtual) OFM is sufficient which can be constructed by hand. However, Korean fonts have proportional width. Since most of the popular Korean fonts are in OpenType format, a utility that extracts font metrics from OpenType fonts is required.

There are two patches of the ttf2tfm and ttf2pk utilities[7] using the freetype library. The first,[8] written by one of the authors, Jin-Hwan Cho, generates OFM and OVF files from TrueType fonts (not OpenType fonts). The other,[9] written by Won-Kyu Park, lets ttf2tfm and ttf2pk run with OpenType (including TrueType) fonts with the help of the freetype2 library. Moreover, two patches can be used together.

Unfortunately, ovp2ovf 2.0 included in recent TEX distributions (e.g., teTEX 2.*x*) does not seem to work correctly, so the previous version 1.*x* must be used.

9 Asian Font Packs and DVIPDFM*x*

A solution avoiding the problems mentioned above is to use the CJK fonts included in the Asian font packs of Adobe (Acrobat) Reader as non-embedded fonts when making PDF output.

It is well known that Adobe Reader can display and print several common fonts even if they are not embedded in the document. These are fourteen base Latin fonts, such as Times, Helvetica, and Courier — and several CJK fonts, if Asian font packs[10] are installed. These packs have been available free of charge since the era of Adobe Acrobat Reader 4. Four are available: Chinese Simplified, Chinese Traditional, Japanese, and Korean. Moreover, Adobe Reader 6 downloads the appropriate font packs on demand when a document containing non-embedded CJK characters is opened. Note that these fonts are licensed solely for use with Adobe Readers.

Professional CJK typesetting requires at least two font families: serif and sans serif. As of Adobe Acrobat Reader 4, Asian font packs, except for Chinese Simplified, included both families, but newer packs include only a serif family. However, newer versions of Adobe Reader can automatically substitute a missing CJK font by another CJK font installed in the operating system, so displaying both families is possible on most platforms.

If the CJK fonts included in Asian font packs are to be used, there is no need to embed the fonts when making PDF output. The PDF file should contain the font names and code points only. Some 'generic' font names are given in Table 1, which can be handled by Acrobat Reader 4 and later. However, these names depend on the PDF viewers.[11] Note that the names are not necessarily true font names. For example, Ryumin-Light and GothicBBB-Medium are the names of commercial (rather expensive) Japanese fonts. They are installed in every genuine (expensive) Japanese PostScript printer. PDF readers and PostScript-compatible low-cost printers accept these names but use compatible typefaces instead.

While TEX generates DVI output only, pdfTEX generates both DVI and PDF output. But Omega and pTEX do not have counterparts generating PDF

[7] Available from the FreeType project, `http://www.freetype.org`.

[8] Available from the Korean TEX Users group, `http://ftp.ktug.or.kr/pub/ktug/freetype/contrib/ttf2pk-1.5-20020430.patch`.

[9] Available as `http://chem.skku.ac.kr/~wkpark/project/ktug/ttf2pk-freetype2_20030314.tgz`.

[10] Asian font packs for Adobe Acrobat Reader 5.x and Adobe Reader 6.0, Windows and Unix versions, can be downloaded from `http://www.adobe.com/products/acrobat/acrrasianfontpack.html`. For Mac OS, an optional component is provided at the time of download.

[11] For example, these names are hard coded in the executable file of Adobe (Acrobat) Reader, and each version has different names.

Table 1: Generic CJK font names

	Serif	Sans Serif
Chinese Simplified	STSong-Light	STHeiti-Regular
Chinese Traditional	MSung-Light	MHei-Medium
Japanese	Ryumin-Light	GothicBBB-Medium
Korean	HYSMyeongJo-Medium	HYGoThic-Medium

output yet. One solution is DVIPDFMx [2], an extension of dvipdfm,[12] developed by Shunsaku Hirata and one of the authors, Jin-Hwan Cho.

10 Conclusion

We have shown how Omega, with CJK-ΩTP, can be used for the production of quality PDF documents using the CJK languages.

CJK-ΩTP, as it stands, is poorly tested and documented. Especially needed are examples of Chinese typesetting, in which the present authors are barely qualified. In due course, we hope to upload CJK-ΩTP to CTAN.

[12] The utility dvipdfm is a DVI to PDF translator developed by Mark A. Wicks. The latest version, 0.13.2c, was released in 2001. Available from `http://gaspra.kettering.edu/dvipdfm/`.

References

[1] ASCII Corporation. ASCII Nihongo TEX (Publishing TEX). `http://www.ascii.co.jp/pb/ptex/`.

[2] Jin-Hwan Cho and Shunsaku Hirata. The DVIPDFMx Project. `http://project.ktug.or.kr/dvipdfmx/`.

[3] John Plaice and Yannis Haralambous. The Omega Typesetting and Document Processing System. `http://omega.enstb.org`.