Języki dalekowschodnie i pakiet CJK

Michał Piskorski
Instytut Orientalistyczny Uniwersytetu Warszawskiego:
Zakład Japonistyki i Koreanistyki, Sekcja Koreanistyki
Zakład Zastosowań Informatycznych
micpis@mercury.ci.uw.edu.pl

Streszczenie

Artykuł ten jest próbą zaprezentowania problematyki związanej z zapisem i przetwarzaniem tekstów w językach dalekowschodnich oraz przybliżenia pakietu CJK autorstwa Wernera Lemberga, którego zadaniem jest wspomaganie składu tekstów w tych językach w LaTeX 2_ε. Opis dotyczy pakietu w wersji 4.3.0. W chwili obecnej została udostępniona wersja 4.4.0, która oprócz języków: chińskiego, japońskiego i koreańskiego umożliwia również skład tekstów w języku tajskim. Pakiet CJK jest rozpowszechniany w ramach Ogólnej Licencji Publicznej GNU.

Wstęp

O tym, że przy pomocy sytemu TeX można tworzyć wyrafinowane formy składu tekstu przekonywać nie trzeba. Wraz z narzędziami do szeroko pojętego profesjonalnego składu tekstu szybko powstawały rozszerzenia umożliwiające tworzenie dokumentów w językach narodowych. Ich rozwój jest wciąż żywy, co możemy zaobserwować również w naszym kraju. Niejako konsekwencją rozwoju cywilizacyjnego w sferze pokonywania barier odległości czy dostępu do systemów wymiany informacji stało się zapotrzebowanie na rozwój narzędzi wspomagających przetwarzanie tekstów zapisanych w wielu różnych językach naturalnych. Takie wyzwanie stanęło również przed TeXem. W LaTeXu do składu dokumentów w różnych językach europejskich można z powodzeniem skorzystać z pakietu Babel, natomiast doskonałym wsparciem przy tworzeniu tekstów zapisanych w językach orientalnych jest pakiet CJK.

Chyba każdy, kto tworzył dokumenty w językach europejskich, napotkał na mniejsze czy większe problemy związane z obsługą znaków diakrytycznych, mimo że ich liczba — nawet jeśli zsumować te znaki we wszystkich językach europejskich — jest stosunkowo niewielka. Jeżeli zauważymy, że w grupie języków CJK: chińskiego, japońskiego i koreańskiego mamy do czynienia z co najmniej kilkoma tysiącami ideogramów praktycznie wykorzystywanych do zapisu tekstu, a ich ogólna liczba szacowana jest na około 85 tysięcy (patrz: [1], str. 58), to staje się oczywiste, że problematyka związana ze sposobem zapisu tych tekstów oraz ich przetwarzaniem nie jest rzeczą trywialną. Co więcej, jest to wyzwanie nie tylko dla specjalistów określających standardy czy informatyków tworzących oprogramowanie, ale również dla użytkowników, który muszą opanować nieraz złożone metody wprowadzania tekstu.

Pisząc ten artykuł przyjąłem założenie, że jego odbiorca posiada niezbędne umiejętności do samodzielnej edycji tekstów orientalnych lub co najmniej dysponuje gotowym tekstem w postaci elektronicznej. Dla mniej wtajemniczonych pragnę jednak krótko przedstawić najbardziej podstawowe różnice i podobieństwa występujące w językach CJK, które w naturalny sposób wywarły wpływ na zagadnienia związane z ich komputerowym przetwarzaniem. Zainteresowanych edycją tekstów japońskich odsyłam do znajdującego się w tym zeszycie artykułu *Teksty wielojęzyczne w edytorze GNU Emacs*, którego autorem jest Janusz S. Bień.

Nieco o językach dalekowschodnich

Jak już wspomniałem, we wszystkich językach CJK wykorzystywane są ideogramy chińskie, przy czym w języku chińskim[1] i japońskim stosowane są one na co dzień, natomiast w Korei — na Północy po zakończeniu Wojny Koreańskiej, a na Południu od kilkunastu lat — możemy zaobserwować ich skuteczną eliminację na rzecz rodzimego alfabetu. Mimo tego, że w większości wypadków dany ideogram określa to samo pojęcie w każdym z krajów, a co więcej ideogramy wykorzystane są w jednakowych złożeniach, trzeba powiedzieć, że różnice ich użycia w poszczególnych krajach są nie tylko ilościowe, ale polegają również na pewnych modyfikacjach graficznych tego samego znaku. Niekiedy zdarza się również, że ten

[1] W artykule tym pomijam wszelkie różnice występujące w Chińskiej Republice Ludowej i na Tajwanie, w tym dotyczące tradycyjnego i uproszczonego sposobu zapisu ideogramów.

sam ideogram używany jest w różnych postaciach graficznych nawet w jednym kraju.

W Korei i Japonii oprócz ideogramów chińskich mamy do czynienia także z alfabetami, przy czym w Japonii są to dwa alfabety: *hiragana* i *katakana*, w których poszczególne znaki reprezentują niepodzielne na litery znaki sylabowe, a w Korei alfabet *hangŭl* złożony z liter, przy czym w tekście litery składane są w sylaby, tak że każda sylaba mieści się w polu kwadratu i wyglądem bardzo przypomina ideogram chiński. Liczba kombinacji tworzących sylabę wynosi 11172, jednakże wybrany ich podzbiór 2350 sylab praktycznie wystarcza w większości zastosowań.

Standardowe zestawy znaków

Uporządkowanie i sklasyfikowanie ogromnej liczby ideogramów chińskich, które są stosowane w językach CJK było konieczne nie tylko do zastosowań informatycznych, lecz również edukacyjnych. To właśnie potrzeba wyselekcjonowania określonego zestawu znaków, których opanowanie w danym kraju jest niezbędne bądź wskazane w procesie edukacyjnym, była źródłem powstania unormowanych zestawów znaków, a te z kolei były najczęściej pierwowzorem kodowych zestawów znaków, opracowanych dla potrzeb przechowywania i przetwarzania informacji w postaci elektronicznej. Kodowe zestawy znaków zawierają z reguły dużo więcej znaków niż zestawy, które powstały dla potrzeb edukacyjnych, a w przypadku języków koreańskiego i japońskiego obejmują również litery i znaki sylabowe alfabetów, przy czym dla języka koreańskiego — ze względu na graficzny sposób tworzenia sylab — w kodowym zestawie znaków znajdują się dodatkowo znaki przedstawiające sylaby złożone z liter.

W kręgu języków CJK nie powstał jeden powszechnie przyjęty standard analogiczny do *ASCII* na Zachodzie, choć zdobywający popularność *Unicode* jest dużym krokiem w tym kierunku.

Pakiet CJK na płycie TeX Live 5

Pakiet CJK autorstwa Wernera Lemberga[2] jest dystrybuowanym na zasadach Ogólnej Licencji Publicznej GNU narzędziem do wspomagania składu tekstów zapisanych w językach: chińskim, japońskim i koreańskim za pomocą LaTeX 2ε. Artykuł ten koncentruje się wokół dystrybucji zawartej na płycie TeX Live 5, która różni się nieco od dostępnych w archiwach CTAN. Pakiet dostępny na płycie zawiera wszelkie niezbędne modyfikacje w plikach konfiguracyjnych (`.fd`, `.map`) w celu dostosowania do zawartych na tej płycie fontów dalekowschodnich. Niemniej dystrybucja na płycie TeX Live 5, poza mniejszą liczbą dostępnych fontów, jest w pełni funkcjonalną wersją pakietu wspierającego skład tekstów zapisanych przy użyciu najbardziej popularnych metod kodowania.

W niektórych wypadkach możliwości pakietu (głównie w przypadku tekstu kodowanego w `UTF-8`) wykraczają poza wsparcie składu jedynie pism w językach CJK. Nie należy jednak tych możliwości nadużywać, gdyż pakiet nie jest optymalizowany pod kątem użycia znaków innych, niż należących do grupy CJK, a więc nie uwzględnia np. podcięć kernowych czy ligatur między fontami składowymi.

Z przeprowadzonych przeze mnie prób wynika, że nie występują problemy przy współpracy pakietu CJK zarówno z pakietem Babel jak i pakietem Polski (również przy przetwarzaniu PLaTeXem), co jest szczególnie istotne z punktu widzenia składu tekstów w języku polskim z odwołaniami do tekstu w językach orientalnych.

Definiowanie środowiska CJK. Do dyspozycji mamy dwa tryby korzystania z pakietu CJK — tryb bezpośredni oraz tryb z przetwarzaniem wstępnym.

W trybie bezpośrednim pakiet CJK wywoływany jest standardowo w preambule przez:

```
\usepackage{CJK}
```

przy czym mogą być definiowane dwa typy środowisk, których opis różni się wystąpieniem gwiazdki po CJK:

```
\begin{CJK}[<kodowanie fontu>]
                        {<Kodowanie>}{<rodzina>}
...
\end{CJK}
```

oraz

```
\begin{CJK*}[<kodowanie fontu>]
                        {<kodowanie>}{<rodzina>}
...
\end{CJK*}
```

W środowisku CJK* pomijane są spacje oraz nowe linie, jeżeli występują bezpośrednio po znaku CJK. Jest to typowa konwencja w tradycji chińskiej i japońskiej. Gdy tekst zapisany znakami innymi niż CJK jest wpisany pomiędzy znaki CJK, powinien zostać oddzielony spacjami, które nie zostaną pominięte. Umożliwia to polecenie `\CJKtilde`, które definiuje nowe znaczenie znaku tyldy jako spacji o wymiarze 0,25 szerokości znaku CJK. Powrót do standardowej definicji tyldy następuje przez `\standardtilde`. Z kolei tradycja koreańska (CJK bez gwiazdki) jest podobna do przyjętej dla języków europejskich i w związku z tym nie ma potrzeby specjalnego traktowania znaku spacji.

[2] `ftp://ftp.ffii.org/pub/cjk`

Wewnątrz środowiska możliwe jest przełączanie między tymi typami za pomocą poleceń `\CJKspace` oraz `\CJKnospace`.

W zaprezentowanych wyżej wywołaniach występują trzy parametry definiujące zachowanie środowiska:

`<kodowanie>` wskazuje na kodowy zestaw znaków i związany z nim sposób kodowania. Najczęściej używane wartości to: `GB`, `JIS`, `KS`, które oznaczają narodowe zestawy znaków kodowane zgodnie z `EUC` (*Extended Unix Code*). Wartości odpowiadające innym popularnym metodom kodowania to: `Bg5`[3], `Bg5+`, `GBK`, `SJIS` oraz `UTF8`.

Lista wartości z podstawowym opisem znajduje się w dokumentacji pakietu. W celu dokładnego poznania różnych kodowych zestawów znaków i metod ich kodowania polecam zaś książkę *CJKV Information Processing*[1].

`<kodowanie fontu>` określa układ znaków w foncie. Zdefiniowane tu są następujące wartości:

`' '` (pusty — wartość domyślna), `pmC`, `dnp`, `wn` oraz `HL`. Z punktu widzenia użytkownika płyty TEX Live 5 — ze względu na zawarte na niej obwiedniowe fonty CJK — szczególnie istotnymi będą `' '` (pusty), wykorzystywany do składu tekstów w języku chińskim, `dnp` dla tekstów w języku japońskim oraz `HL` dla tekstów w języku koreańskim.

`<rodzina>` jest wartością zdefiniowaną w odpowiednich plikach deklaracji fontów (`.fd`). Fonty na płycie zostały tak dobrane przez autora pakietu, aby miały podobny krój i harmonizowały z zachodnim *Times*. Pakiet CJK operuje fontami w sposób zgodny z NFSS, przy czym układ fontu określony jest przez dwucyfrową liczbę poprzedzoną literą „C”, np. font zgodny ze standardem *Unicode* ma układ C70. Na płycie możemy m.in. znaleźć deklarację fontu C70 rodziny *song* w pliku `c70song.fd`.

W pracy orientalisty stosunkowo rzadko dochodzi do posługiwania się tylko tekstem w oryginale. Niestety praktyka wykazuje, że gdy tylko zajdzie potrzeba stworzenia dokumentu w języku polskim oraz jednym z języków dalekowschodnich, zaraz pojawia się problem znalezienia odpowiedniego kodowania zawartych w nim znaków. Co więcej, w większości wypadków problemem będzie również zapis tekstów w różnych językach CJK jednocześnie w jednym dokumencie.

[3] Dla dokumentów kodowanych w `Bg5`, `Bg5+`, `GBK` oraz `SJIS` wskazane jest wykorzystanie przetwarzania wstępnego, gdyż kody zawierają znaki zakłócające poprawną pracę TEXa. Odpowiednie konwertery dostępne są razem z pakietem.

Środowisko CJK zdefiniowane w podany wcześniej sposób oczekuje określonego kodowania znajdującego się w nim tekstu. Załóżmy, że naszym zadaniem będzie skład w językach: japońskim[4] i koreańskim oraz tłumaczenia w języku polskim. Jeżeli posiadamy teksty w oryginale, z których każdy zapisany jest w postaci elektronicznej w akceptowanym przez pakiet CJK kodowaniu (np. japoński w `EUC-JP`, a koreański w `EUC-KR`), zadanie możemy wykonać nawet w prostym edytorze:

- zapisujemy tekst tłumaczenia oraz deklarujemy środowiska dla języka japońskiego i koreańskiego (wskazując kodowanie w jakim mamy zapisane teksty w oryginale)
- korzystamy z polecenia `\input` lub wklejamy teksty w oryginale (ale bez określania kodowania przy kopiowaniu)
- po każdym zamknięciu środowiska CJK odnawiamy deklarację kodowania polskich diakrytyków przez `\inputencoding`
- zapisujemy dokument w kodowaniu stosownym do określonego w `\inputenc`

jak w przykładzie poniżej[5]:

```
\documentclass{article}
\usepackage[OT4]{polski}
\usepackage[latin2]{inputenc}
\usepackage{CJK}
% aby wykorzystać fonty typu 1 z~HLaTeXa:
\usepackage{pshan}
\begin{document}
Dąży pięć wioślarek do źródeł Czarnej Hańczy.

\begin{CJK*}[dnp]{JIS}{min}
%dołączamy plik japanese.tex w~kodowaniu EUC-JP
\input{japanese}
\end{CJK*}

\inputencoding{latin2}
tłumaczenie w~języku polskim

\begin{CJK}[HL]{KS}{pmj}
%dołączamy plik korean.tex w~kodowaniu EUC-KR
\input{korean}
\end{CJK}

\inputencoding{latin2}
tłumaczenie w~języku polskim
\end{document}
```

Z powyższego przykładu wynika, że nie jest to metoda, która pozwala na łatwą edycję tekstów dalekowschodnich. Co więcej, wielokrotne odwoływanie się do oryginału, nawet w przypadku pojedynczych wyrazów, wymaga definiowania nowego środowiska i wczytywania zewnętrznego pliku. Jeżeli zaś

[4] Skład tekstów w języku chińskim jest w dużej mierze analogiczny do japońskiego.

[5] Uwaga: wykorzystanie mechanizmu tablic przekodowania jest źródłem błędu przy przetwarzaniu.

tekst jest bezpośrednio umieszczony w dokumencie, jest nieczytelny i nie może być edytowany.

Emacs dobry na wszystko

W przypadku, gdy korzystamy z pakietu CJK do wyboru mamy dwa rozwiązania, które umożliwiają wygodny skład tekstów wielojęzycznych w LaTeXu: wykorzystanie do zapisu kodowania `emacs-mule` lub `UTF-8`. Oczywiście do składu tekstów w językach innych niż CJK, np. polskim, wciąż koniecznie jest zastosowanie odpowiednich pakietów. W dalszej części artykułu przedstawię bliżej sposób przygotowania dokumentów w obu kodowaniach oraz proces adaptacji fontów true type w standardzie *Unicode* dla potrzeb LaTeXa.

Edytorem, który pozwala na łatwe wprowadzenie tekstu oraz jego odpowiednie zakodowanie jest GNU Emacs, który począwszy od wersji 20 ma wbudowany Mule (*Multilingual Environment*), rozszerzający funkcje Emacsa o możliwość edycji i zapisu tekstów wielojęzycznych. Należy tu jednak zaznaczyć, że w celu uaktywnienia metod wejściowych należy zainstalować bibliotekę *LEIM* oraz odpowiednie fonty, np. *GNU International Fonts*, aby znaki były widoczne na ekranie. Dokonanie zmian w pliku konfiguracyjnym Emacsa zgodnie z instrukcją w pakiecie CJK umożliwia wygodną edycję z wykorzystaniem AUC TeXa.

Emacs 20.x nie pozwala na użycie kodowania `UTF-8`. Aby zapewnić możliwość edycji w pełnym zakresie standardu *Unicode* oraz zapisu dokumentu w kodowaniu `utf-8`, polecam zainstalowanie pakietu *oc-unicode* (`ftp://ftp.cs.ust.hk/pub/ipe/oc-unicode-0.72.2.tar.gz`).

Emacs-mule. Dla tekstów kodowanych w `emacs-mule` stosujemy tryb z przetwarzaniem wstępnym, jako że LaTeX nie potrafi bezpośrednio dokonać składu dokumentu zapisanego w tym kodowaniu. W pakiecie znajduje się plik `cjk-enc.el`, który najlepiej umieścić w emacsowym katalogu ze źródłami w lispie i załadować przy starcie edytora. Polecenie `M-x cjk-write-file` (a w przypadku dokumentów złożonych `M-x cjk-write-all-files`) powoduje utworzenie, analogicznego do TeXowego, pliku z rozszerzeniem `.cjk` (dla dokumentów złożonych — dotyczy to również dołączanych plików).

Jeżeli wykorzystujemy mechanizm przetwarzania wstępnego dla kodowania `emacs-mule`, nie występuje konieczność określania środowisk CJK lub CJK*, a nawet deklaracji samego pakietu CJK, gdyż dodatkowo, w pierwszej linii dokumentu, zostają automatycznie umieszczone wywołanie pakietu i domyślne opcje środowiska CJK.

Aby skorzystać z fontów skalowalnych, które znajdują się na płycie TeX Live 5, trzeba jednak jawnie podać kodowanie fontu dla odpowiedniego zestawu znaków poleceniem `\CJKfontenc{<kodowanie>}{<kodowanie fontu>}`, a dla języka koreańskiego również wywołać pakiet *pshan*.

Poniższy tekst to przykładowy dokument, który zostanie poddany przetwarzaniu wstępnemu:

```
\documentclass{article}
\usepackage[OT4]{polski}
% aby wykorzystać fonty typu 1 z~HLaTeXa:
\usepackage{pshan}
% kodowanie fontów dla języka koreańskiego
\CJKfontenc{KS}{HL}
% kodowanie fontów dla języka japońskiego
\CJKfontenc{JIS}{dnp}
% furigana - napisy nad ideogramem
\usepackage[CJK,overlap]{ruby}
% zmiana odstępu między furigana i~ideogramem
\renewcommand{\rubysep}{-0.3ex}

\begin{document}
\noindent Dąży pięć wioślarek
do źródeł Czarnej Hańczy.

\noindent 이것은 \LaTeX{}로 작성된 한국어 문서입니다.

\noindent czyli: ,,To jest dokument w~języku
koreańskim złożony w~\LaTeX{}u.''

\CJKtilde

\noindent これは~\LaTeX~で\ruby{作}{さく}
\ruby{成}{せい}した日本語のテキストです。

\standardtilde

\noindent czyli: ,,To jest tekst w~języku
japońskim złożony w~\LaTeX{}u.''
\end{document}
```

Nowością jest tu wprowadzenie pakietu *ruby*, który pozwala na umieszczenie zapisu czytania znaku (*furigana*) nad ideogramem chińskim. Zabieg ten jest stosowany głównie w Japonii, gdy w tekście znajdują się rzadko używane ideogramy, których odczytanie mogłoby stanowić pewną trudność. Dzięki opcji `overlap` kilkusylabowy zapis czytania znaku może rozciągać się nad znakami sąsiadującymi z opisywanym. Parametr `rubysep` określa odstęp między zapisem czytania znaku i znakiem opisywanym.

A oto wynik składu:

Dąży pięć wioślarek do źródeł Czarnej Hańczy.

이것은 LaTeX로 작성된 한국어 문서입니다.

czyli: „To jest dokument w języku koreańskim złożony w LaTeXu."

さく せい
これは LaTeX で作成した日本語のテキストです。

czyli: „To jest tekst w języku japońskim złożony w LaTeXu."

Unicode krok po kroku

Dla kodowania UTF-8 nie stosujemy przetwarzania wstępnego, a wywołanie pakietu CJK oraz określenie parametrów środowiska deklarowane są jawnie. Do składu tekstu w przykładzie poniżej użyjemy fontu w zdefiniowanej nieco dalej rodzinie *arial*.

```
\documentclass{article}
\usepackage[OT4]{polski}
\usepackage[latin2]{inputenc}
\usepackage{CJK}
\begin{document}
\begin{CJK}{UTF8}{arial}

<tekst CJK w~kodowaniu UTF-8>

\end{CJK}
\end{document}
```

Konieczność stworzenia w miarę uniwersalnego zestawu znaków dla potrzeb przetwarzania tekstów wielojęzycznych przyczyniła się do zauważalnej ekspansji *Unicode*'u. Coraz więcej edytorów akceptuje ten standard, a na rynku pojawił się dość pokaźny zbiór fontów w tym układzie. Niestety tylko nieliczne zawierają znaki CJK. Nie należy się temu dziwić — opracowanie kroju dla kilkudziesięciu tysięcy znaków jest zadaniem poważnym. Najczęściej spotykanymi fontami w standardzie *Unicode* są fonty true type (mogą mieć rozszerzenie `ttf` lub `ttc`). Font zawierający ideogramy chińskie i inne znaki używane w językach CJK ma zwykle objętość nie mniejszą niż 10MB, w przypadku bardziej „bogatych”, zwykle komercyjnych fontów rozmiar może sięgać nawet 30MB.

Jeżeli mamy do dyspozycji font z zestawem znaków zgodnym z Unicode 2.0 (w wersji Unicode 1.x część znaków znajdowała się w innym miejscu niż w wersji 2.0, wiele innych nie było zaś zdefiniowanych w ogóle), to możemy przeprowadzić pewne operacje, dzięki którym — z wykorzystaniem pakietu CJK — będziemy mogli go użyć w tekście złożonym w LaTeXu. Wszystkie potrzebne narzędzia znajdziemy bezpośrednio na płycie TeX Live5.

Dość obszerny font *Arial Unicode MS*[6] nieodpłatnie udostępnia Microsoft Corp. Znajdziemy go na stronie `http://office.microsoft.com/2000/downloaddetails/aruniupd.htm`. Font zapisujemy np. w katalogu `/fonts/truetype/ms/arialuni`.[7]

Wpierw utworzymy zestaw plików `tfm`, który pozwoli TeXowi dokonać poprawnego składu. Każdy z plików otrzyma nazwę `arunixx.tfm`, gdzie `xx` jest liczbą z zakresu `00` $\sim$ `ff` i stanowi starszy bajt dwubajtowej reprezentacji znaku w standardzie *Unicode*. Korzystamy z programu `ttf2pk`, który odwołuje się do pliku `/ttf2pk/unicode.sfd` definiującego sposób podziału fontu na 255-cio znakowe fonty składowe:

```
ttf2tfm arialuni aruni@unicode@
```

Wygenerowane fonty umieszczamy np. w katalogu `/fonts/tfm/ms/arialuni` i dokonujemy wpisu do `/fontname/special.map`:

```
@c Arial Unicode MS
aruni00   ms   arialuni
aruni01   ms   arialuni
...
aruniff   ms   arialuni
```

Ponieważ pakiet CJK posiada domyślną deklarację fontu C70 rodziny *song* w pliku `c70song.fd`, możemy na jej podstawie utworzyć analogiczną definicję rodziny *arial* z odwołaniem do fontu *Arial Unicode MS* w pliku `c70arial.fd`:

```
\DeclareFontFamily{C70}{arial}{}
\DeclareFontShape{C70}{arial}{m}{n}
                        {<-> CJK * aruni}{}
\DeclareFontShape{C70}{arial}{bx}{n}
                 {<-> CJKb * aruni}{\CJKbold}
```

Trzecia linia wskazuje na sposób tworzenia pisma grubego przez trzykrotne drukowanie znaku z domyślnym przesunięciem 0,015em. Jest to bardzo użyteczna metoda, gdyż fonty CJK dość rzadko posiadają rzeczywistą wersję pisma grubego.

Plik `/ttf2pk/ttfonts.map` uzupełniamy jeszcze o wpis:

```
aruni@Unicode@   arialuni
```

i przy próbie wyświetlenia pliku `.dvi` powinien nastąpić proces automatycznego generowania fontów `pk` z wywołaniem programu `ttf2pk`.

Utworzymy teraz, odpowiedni do plików `tfm`, szereg fontów składowych typu 1, które wykorzystamy przy generowaniu plików postscriptowych. Sięgamy po program `ttf2pfb` i uruchamiamy go każdorazowo przy tworzeniu kolejnego fontu składowego. Dzięki opcji `-a` otrzymujemy również pliki z wektorami kodowania (`.enc`) dla poszczególnych fontów składowych:[8]

```
ttf2pfb -a -plane 0x00 -f ArialUni
                -o aruni00.ps arialuni.ttf
...
ttf2pfb -a -plane 0xff -f ArialUni
                -o aruniff.ps arialuni.ttf
```

[6] Dziekuję Panu Adamowi Twardochowi za wskazanie tego fontu.

[7] Po umieszczeniu nowych plików w strukturze katalogów TeXa należy pamiętać o przprowadzeniu aktualizacji bazy danych przez wykonanie `mktexlsr`.

[8] Program uruchamiamy z zapisem opcji w jednej linii.

Aby otrzymać postać binarną tych fontów musimy skorzystać z programu `t1asm`. Wywołujemy go w następujący sposób:

```
t1asm -b aruni00.ps aruni00.pfb
...
t1asm -b aruniff.ps aruniff.pfb
```

Wszystkie fonty składowe typu 1 umieszczamy np. w katalogu `/fonts/type1/ms/arialuni`.

Zanim jednak uruchomimy `dvips`-a, należy dokonać rejestracji fontu w plikach konfiguracyjnych. W tym celu tworzymy plik `arialuni.map` o następującej zawartości:

```
aruni00 ArialUni00 <aruni00.pfb
...
aruniff ArialUniff <aruniff.pfb
```

a następnie uzupełniamy plik `config.ps` przez dodanie linii:

```
p +arialuni.map
```

Należy pamiętać, że `dvips` trzeba wywoływać z opcją `-j0`. Niestety przy częściowym ładowaniu fontów otrzymujemy komunikat o wystąpieniu błędu. Tego problemu oczywiście nie ma, gdy korzystamy z odpowiednich fontów typu 1 znajdujących się na płycie TeXLive 5 przy opisanym wcześniej kodowaniu `emacs-mule`.

Zobaczmy teraz efekty naszej pracy. Poniższy przykład został zapisany w kodowaniu `utf-8` i złożony przy użyciu zaadoptowanego fontu *Arial Unicode MS*:

Dąży pięć wioślarek do źródeł Czarnej Hańczy.
이것은 LaTeX로 작성된 한국어 문서입니다.
czyli:
,,To jest dokument po koreańsku złożony w LaTeXu."
これはLaTeXで作成した日本語のテキストです。
czyli:
,,To jest tekst w języku japońskim złożony w LaTeXu."

Zakończenie

W większości wypadków zastosowanie pakietu CJK dla tekstu kodowanego w `emacs-mule` jest wystarczające. Nie należy też pochopnie rezygnować z tego kodowania na rzecz `utf-8` z kilku powodów. Po pierwsze trzeba pamiętać, że standard *Unicode* opisuje jedynie pewien abstrakcyjny zestaw znaków kodowych, nie definiuje natomiast reprezentacji graficznej tych znaków. Jest to bardzo istotne z punktu widzenia składu tekstów dalekowschodnich. Dany znak kodowy, nawet jeżeli we wszystkich językach CJK wiąże się z ideogramem określającym to samo pojęcie, posiada reprezentację graficzną, która pod względem typograficznym różni się w pismach chińskim, japońskim i koreańskim. Zależnie od języka konsekwencją może się być również użycie ideogramu tradycyjnego bądź uproszczonego. Dlatego przy wyborze fontu, trzeba zwrócić uwagę na to, dla potrzeb jakiego pisma został stworzony. Po drugie, ze względu na konstrukcję transformacji *Unicode*'u do `utf-8`, pakiet CJK automatycznie zostaje uaktywniony jedynie dla znaków o kodach wyższych niż 0x80. Jeżeli zatem wewnątrz środowiska CJK znajdzie się tekst w języku polskim, to diakrytyki zostaną złożone określonym w środowisku fontem, natomiast pozostałe litery fontem jaki TeX używa poza środowiskiem CJK. Można oczywiście wykorzystać typowy dla LaTeXa mechanizm deklaracji fontu, aby podstawowym został pierwszy font składowy (jak w przykładzie), nie rozwiązuje to jednak problemu braku odmiany grubej. Również jak wspominałem pakiet nie uwzględnia podcięć kernowych czy możliwych ligatur między fontami składowymi, gdyż taki problem nie wystepuje w językach CJK.

Zdarza się jednak, że dzięki kodowaniu `utf-8` można dokonać składu tekstu dalekowschodniego, jaki nie byłby możliwy w kodowaniu `emacs-mule`. Osobiście znalazłem się w takiej sytuacji, gdy w tekście koreańskim znajdowały się znaki ujęte w standardzie *Unicode*, a które nie są uwzględnione w kodowaniu `EUC-KR`, a więc również w `emacs-mule`.

Ostatnie zmiany w pakiecie CJK, które znajdą się wkrótce w oficjalnej dystrybucji, umożliwiają włączanie i wyłączanie przetwarzania wstępnego w jednym dokumencie. W praktyce oznacza to, że będzie można przetwarzać dokumenty zawierające fragmenty tekstu w kodowaniu `emacs-mule` i `utf-8`, jak to ma miejsce w tym artykule.

Bibliografia

[1] Ken Lunde, *CJKV Information Processing*, O'Reilly, Sebastopol 1999.

[2] *Szturm na wieżę Babel. Panorama języków Azji i Afryki*, Festiwal Nauki 19–27.IX.1998, Instytut Orientalistyczny UW, Warszawa 1998, ISBN 83-86483-75-X.

[3] Janusz S. Bień, *Teksty wielojęzyczne w edytorze GNU Emacs*, bieżący numer Zeszytu

[4] Roman Czyborra, *Unicode Transformation Formats*, `http://czyborra.com/unicode/utf`

[5] Unicode Consortium, *Code charts (pdf)*, `http://www.unicode.org/charts/`

◇ Michał Piskorski
micpis@mercury.ci.uw.edu.pl