The `otibet`-package
Writing tibetan text for Ω

Norbert Preining

May 16, 2010

1 Introduction

This package should ease the pain when writing tibetan text for the typesetting system Ω, a Unicode-capable descendent of TEX. This package makes heavy use of Ω translation process, known as ΩTP. Included in the package there are the necessary ΩTP-files to type tibetan text in a certain transcription and after processing the file with Ω getting the tibetan glyphs.

Together with this package comes a font `tibetan.mf` for METAFONT, which has to be used (cf. sec. 8).

This package is capable of rendering all the basic tibetan compound glyphs (as `rka`) in their usual form. Complex Hindi consonant stacks are made up from the basic characters by stacking them one above the other.

2 Input Encoding, Transcription

Two transcriptions for reading the input are supported, the "Wylie's Transcription" and the one used in the Unicode standard (cf. [1]). These two only differ in how to write the inverted characters. For these characters there will be a double entry under transcription in the following table, standing for "Wylie/Unicode". Each one of this input encodings can be selected as an option (see chap. 9).

Transl.	Tib.	txa/tta		pha	
		thxa/ttha		ba	
ka		dxa/dda		ma	
kha		dxha/ddha		tsa	
ga		nxa/nna		tsha	
nga		ta		dza	
ca		tha		wa	
cha		da		zha	
ja		na		za	
nya		pa		'a	

ya	ཡ་	sha	ཤ་	ha	ཧ་
ta	ཏ་	shxa/ssa	ཥ་	a	ཨ་
la	ལ་	sa	ས་	/	།
				.,	་

For the distinction between praefixed ‘g’ and main ‘g’, as in g.yag, use ‘{g}’ (‘{g}yag’ gives གཡག while ‘gyag’ gives གྱག).

3 Special characters

Some special characters are accessible by the use of control-sequences:

`\om`	0F00	
`\endsym,\gtertsheg`	0F14	
`\anusvara`	0F71	◌ं
`\hrih`	---	
`\dme`	---	
`\hung`	---	
`\swasti`	---	

4 Unicode characters not implemented

The following Unicode characters are at the moment not implemented:

`\vowelii`	0F73	◌	`\nyizlanaada`	0F82	◌
`\voweluu`	0F75	◌	`\snaldan`	0F83	◌
`\vowelr`	0F76	◌	`\halanta`	0F84	◌
`\vowelrr`	0F77	◌	`\paluta`	0F85	◌
`\vowell`	0F78	◌	`\lcirtags`	0F86	◌
`\vowelll`	0F79	◌	`\yangrtags`	0F87	◌
`\vowelee`	0F7B	◌	`\lcetsacan`	0F88	◌
`\voweloo`	0F7D	◌	`\mchucan`	0F89	◌
`\rnambcad`	0F7F	◌	`\grucanrgyings`	0F8A	◌
`\reversedi`	0F80	◌	`\grumedrgyings`	0F8B	◌
`\reversedii`	0F81	◌			

5 Output Encoding

The package `otibet` internally uses the Unicode encoding of the tibetan glyphs (`U+0F00--U+0FBF`), although not every glyph can (at the moment) be rendered (depending on the font). The internal processing is done in Unicode and finally the output is recoded to the (very) special arrangement of the characters in the font `tibetan` (cf. sec. 8).

6 Specialities

A tsheg is automatically inserted at a space and deleted at nonsense places. This option can be turned of with the option "`notsheg`" (not implemented yet!).

7 Consonant Clusters

All the basic tibetan glyphs are rendered in there usual form. If a cluster of consonants cannot be rendered, the basic glyphs are just stacked one over the other. So you get

- from "rgyo" a "  "
- from "ltsr" a "  ".

8 The font `tibetan`

The font `tibetan` is based on Sirlins fonts `gtib` and `gtibsp`, which in turn is based on `pk`-files named `tib.300pk` and `tibsp.300pk`. A few contributions of the author should fill the gaps to a font covering all Unicode characters.

9 Options to the package

There are some options which control the behaviour of the package:

- `wylie-input, unicode-input`: These two options select the input encoding, the transcription. See chap. 2 for details. The `wylie-input` is the default.
- `tibetan-output, latin-output`: These two options select the output encoding and the used font.
- `notsheg`: Inhibits the putting of a tsheg after syllabes. *Not implemented yet.*
- `realunicode`: Maybe in the future a text will come in Unicode, not in `ASCII`, so you can select it. *Not implemented yet.*

10 System Requirements

You need Ω and a large `buff_size;-)`.

References

[1] The Unicode Consortium. *The Unicode Standard, Version 2.0.* Addison-Wesley, 1996.