The akshar package

Vu Van Dung

Version 0.1 — 2020/05/17

Contents

Ind	ex	2
3	Implementation	1
2	User guide	1
1	Introduction	

1 Introduction

When dealing with processing strings in the Devanagari script, normal LATEX commands usually find some difficulties in distinguishing "normal" characters, like 雨, and "special" characters, for example of or of. Let's consider this example code:

- ∖ExplSyntaxOn
- ² \tl_set:Nn \l_tmpa_tl { की}

2 tokens.

- 3 \tl_count:N \l_tmpa_tl \c_space_token tokens.
- $_4$ \ExplSyntaxOff

The output is 2, but the number of characters in it is only one! The reason is quite simple: the compiler treats $\hat{\ }$ as a normal character, which it isn't.

To tackle that, this package provides expl3 functions to "convert" a given string, written in the Devanagari script, to a sequence of token lists. each of these token lists is a "true" Devanagari character. You can now do anything you want with this sequence; and this package does provide some front-end macros for some simple actions on the input string.

2 User guide

a

क्ष

b

कौ

C

क्ष्य

d

3 Implementation

- $_{\scriptscriptstyle 1}$ (@@=akshar)
- 2 (*package)

Declare the package.

```
3 \RequirePackage{fontspec}
4 \ProvidesExplPackage {akshar} {2020/05/17} {0.1}
    {Support for syllables in the Devanagari script (JV)}
6 \tl_const:Nn \c__akshar_joining_tl { []}
7 \tl_const:Nn \c__akshar_diacritics_tl {000000000000}
& \tl_new:N \l__akshar_input_tl

  \tl_new:N \l__akshar_tmp_tl

10 \bool_new:N \l__akshar_prev_joining_bool
11 \seq_new:N \l__akshar_char_seq
12 \prg_generate_conditional_variant:Nnn \tl_if_in:Nn { No } { TF }
13 \cs_new:Npn \__akshar_str_getchar:nn #1 #2
    {
14
      \seg clear:N \l akshar char seg
15
      \bool_set_false:N \l_ akshar_prev_joining_bool
16
      \tl_set:Nn \l__akshar_input_tl {#1}
      \tl_map_variable:NNn \l__akshar_input_tl \l__akshar_map_tl
18
          \tl_if_in:NoTF \c__akshar_diacritics_tl {\l__akshar_map_tl}
              % It is a diacritic.
              \seq_pop_right:NN \l__akshar_char_seq \l__akshar_tmp_tl
          \seq_put_right:Nx \l__akshar_char_seq { \l__akshar_tmp_tl \l__akshar_map_tl }
            }
            {
              \tl if eq:NNTF \l akshar map tl \c akshar joining tl
                {
28
                  % It is the joining character
                  \seq_pop_right:NN \l__akshar_char_seq \l__akshar_tmp_tl
                  \seq_put_right:Nx \l__akshar_char_seq
                    { \l_akshar_tmp_tl \l_akshar_map_tl }
                  \bool_set_true:N \l__akshar_prev_joining_bool
                }
                {
                  % It is a normal character
                  \bool_if:NTF \l__akshar_prev_joining_bool
                    {
                      % but previously there is a joining character
                      \seq_pop_right:NN \l__akshar_char_seq \l__akshar_tmp_tl
                      \seq_put_right:Nx \l__akshar_char_seq
                        { \l__akshar_tmp_tl \l__akshar_map_tl }
                      \bool_set_false:N \l__akshar_prev_joining_bool
                    }
                    {
                      % Previously: nothing special.
                      \seq_put_right:Nx \l__akshar_char_seq { \l__akshar_map_tl }
                }
            }
50
          % Plus two just to guard against breaking too soon
          \int_compare:nNnT {\seq_count:N \l__akshar_char_seq} > {#2 + 2}
53
54
              \tl_map_break:
55
      \seq_item:Nn \l__akshar_char_seq {#2}
59 \NewDocumentCommand \mystrchar {mm}
         _akshar_str_getchar:nn {#1} {#2}
    }
63 (/package)
```

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Α	N
akshar internal commands:	\NewDocumentCommand 59
\lakshar_char_seq 11,	
15, 23, 24, 30, 31, 40, 41, 47, 52, 57	P
<pre>\cakshar_diacritics_tl 7, 20</pre>	prg commands:
\lakshar_input_tl 8, 17, 18	<pre>\prg_generate_conditional</pre>
<pre>\cakshar_joining_tl 6, 27</pre>	variant:Nnn 12
\lakshar_map_tl	\ProvidesExplPackage 4
<pre>\lakshar_prev_joining_bool</pre>	R
10, 16, 33, 37, 43	\RequirePackage 3
<pre>\akshar_str_getchar:nn 13, 61</pre>	
\lakshar_tmp_tl	S
9, 23, 24, 30, 32, 40, 42	seq commands:
	\seq_clear:N 15
В	\seq_count:N 52
bool commands:	\seq_item:Nn 57
\bool_if:NTF	\seq_new:N 11
\bool_new:N 10	\seq_pop_right:NN 23, 30, 40
\bool_set_false:N 16, 43	\seq_put_right:Nn 24, 31, 41, 47
\bool_set_true:N 33	
С	T
cs commands:	tl commands:
\cs_new:Npn 13	\tl_const:Nn 6, 7
/c3_new.wpn	\tl_if_eq:NNTF 27
I	\tl_if_in:Nn 12
int commands:	\tl_if_in:NnTF 20
\int_compare:nNnTF 52	\tl_map_break: 54
	\tl_map_variable:NNn 18
M	\tl_new:N
\mystrchar 59	\tl_set:Nn 17