Tibetan Language Specification

Tibetan letters are composed of :

- 30 consonants
- 4 vowels which combine with the 30 consonants
- 3 superscribed letters which combine individualy with some consonants
- 4 subjoined letters which combined individualy with some consonants
- 5 prefixes which combine with some consonants
- 10 suffixes which combine with all consonants (or suffixes?)

Words are made of syllabs
A syllab can have between 1 and 4 letters (with or without a vowel)
Reading of a letter is made in a sequence : letter, letter superscript, letter subjoin, vowel.
Letter spelling is the result of the various combinations between letter, superscript, subjoin,vowel.

Possible combination for a letter : (consonant includes the 'a') :

- consonant — ex : la
- consonant + vowel — ex : la na-ro = lo
- consonant + superscript — ex : ra-ta la = rla
- consonant + superscript + vowel — ex : ra-ta la na-ro = rlo
- consonant + subscript — ex : la ra-ta = lra
- consonant + subscript + vowel — ex : la ra-ta na-ro = lro
- consonant + superscript + subscript — ex : ra-ta la wa-sur = rlwa
- consonant + superscript + subscript + vowel — ex : ra-ta la wa-sur na-ro = rlwo

A syllab is a combination of any of the above, between 1 and 4 letters.
Ex : sa pa-ta ya-ta na-ro da = spyod (transl)

Spelling rules :

Syllab spelling is the result of letter spelling + prefix and suffix/post_suffix
If a consonant is a subscript letter, spelling is : consonant-TA
If a consonant is a superscript letter, the root letter (letter below the superscript) spelling is : consonant-TA
If a consonant is a prefix, spelling is : consonant-O
If a consonant is a suffix or post-suffix, it lose its vowel for spelling

```
// Syllab_Identification
// Each letter is entered as a set of characters separated by space, following the spelling order :
// Cases :
// - single letter :                    letter
// - letter+vowel :                     letter(space)vowel
// - letter+subscript :                 letter(space)subscribedletter-ta
// - letter+subscript+vowel :           letter(space)subscribedletter-ta(space)vowel
// - letter+superscrip :                superscriptedletter(space)letter-ta
// - letter+superscript+vowel :         superscriptedletter(space)letter-ta(space)vowel
// - letter+subscript+superscript :     letter(space)letter-ta(space)letter-ta
// - letter+subscript+superscript+vowel :  letter(space)letter-ta(space)letter-ta(space)vowel

lettercount = 1
do while (lettercount <> «tchag »)
        read_letter(lettercount)
        identify_letter(lettercount)
        store_letter(lettercount)
        lettercount++
end

// RootLetter_Identification

case lettercount of
        1 : rootletter = letter[1] ; break
```

```
        2 : rootletter = letter[1] ; break
        4 : rootletter = letter[2] ; break
        3 : three_letter_rootletter_identification()
        other : print « letter count do not match – cannot assess root letter »
end caseof


// Three_letter_rootletter_identification

count = 1
do while (count <4)
        read_letter(count)
        if (letter(count) has a vowel and letter(count) <> 'a'), or if letter(count) has a superscribed or if letter(count)
        has a subjoined
        then rootletter = letter(count)
end
          if letter[3] = suffix and letter[3] <> 'sa'
          then rootletter = letter[2]
          if letter[3] = 'sa' and letter[2] = non_affix
          then rootletter = letter[2]
          if letter[3] = 'sa' and letter[1] = affix and letter[2] = affix and word belongs to [list1]
          then rootletter = letter[2]
          if letter[3] = 'sa' and letter[1] = affix and letter[2] = affix and word belongs to [list2]
          then rootletter = letter[1]
end

case lettercount of
        1 : ident_syllab()
        2 : check_letter[2] = valid_suffix
            if valid_suffix
            then word = letter[1] + letter[2]
                 ident_syllab()
            else print « invalid suffix »
        3 : if root_letter = 1
            then check_letter[2] = valid_suffix
                 check_letter[3] = valid_post_suffix
                 if valid_suffix and valid_post_suffix
                 then word = letter[1] + letter[2] + letter[3]
                       ident_syllab()
                 else print « invalid suffix or post-suffix »
            else if root-letter = 2
                 then check_letter[1] = valid_prefix
                      check_letter[3] = valid_suffix
                      if valid_prefix and valid_suffix
                      then word = letter[1] + letter[2] + letter[3]
                            ident_syllab()
                      else print « invalid prefix or suffix »
        4 : check_letter[1] = valid_prefix
            check_letter[3] = valid_suffix
            check_letter[4] = valid_post_suffix
            if valid_prefix and valid_suffix and valid_post_suffix
                 then word = letter[1] + letter[2] + letter[3]
                       ident_syllab()
                 else print « invalid prefix or suffix or post_suffix »

// END syllab identification
```