Abstract: This is a discussion about the Bangla character representation in computer system and the sorting algorithm for Bangla words as well. Until recently a number of works have been conducted but no standard has been established. In our paper we have highlighted the limitations of those previous works and proposed a suitable solution. This will be convenient for Bangla word processing, voice synthesis, character recognition and compiler construction, etc.

Keywords: Bangla Character Representation, Bangla Text sorting.

1. INTRODUCTION

Bangla is a very rich language and approximately 10% of world’s populations speak in Bangla [1]. Hence, the computerization of this language is the inevitable need today, but unfortunately we have advanced a very little in this regard. Even there is no standard in the Bangla keyboard layout or computer representation of Bangla characters. More significantly, for the development of Bangla database systems a standard for representation of Bangla characters and an expedient, efficient, versatile sorting algorithm is a must.

In our paper we are proposing a method to represent Bangla characters internally in the computer systems, which will provide the scope of efficient sorting of Bangla words. To represent any character set, at least two criteria must be fulfilled. First, it should work along with the typical and standard word processors with ease of writing, saving and retrieval. Secondly, it should be able to undergo further mathematical computation in an efficient manner. All the present keyboard interfaces can easily meet the first criterion as they are smoothly being used in MS-WORD and other word processors. But the word format used in various word processors is not suitable for sorting, matching etc. Because the way the character strings are stored in physical devices is not convenient for any mathematical computation such as sorting. Hence, a logical and suitably defined representation of Bangla character set should be evolved. Since we are just a few years ahead from the inception of computerization in our country, if any standard for Bangla is not introduced, with the passage of time it will be very difficult for us to convert all the works to a standard. In the following pages we will introduce all the pros and cons of Bangla character representation and propose a standard for it.

2. The Bangla Language

In the written form of Bangla there are 11 vowels and 39 consonants. Moreover, there are 10 short forms of vowels called vowel modifiers (ie, Kar), 7 short forms of consonants called consonant modifiers (ie, Fala) [1]. Beside these, there are more than about 253 compound characters composed of 2, 3 or 4 consonants (200 compound characters composed of 2 consonants, 51 compound characters composed of 3 consonants and 2 compound characters composed of 4 consonants) [4]. In accordance with the order of Bangla Academy standard, vowels and corresponding vowel modifiers and their placement within words are listed in Table1.

<table>
<thead>
<tr>
<th>Vowels</th>
<th>Vowel Modifiers</th>
<th>Placement (with respect to the character to be modified)</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>আ</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>আ</td>
<td>চ</td>
<td>right</td>
<td>সাহস</td>
</tr>
<tr>
<td>ই</td>
<td>চ</td>
<td>left</td>
<td>বিশ</td>
</tr>
<tr>
<td>ঈ</td>
<td>চ</td>
<td>right</td>
<td>দীক্ষ</td>
</tr>
<tr>
<td>উ</td>
<td>ন</td>
<td>below</td>
<td>সূচনা</td>
</tr>
<tr>
<td>ঊ</td>
<td>ন</td>
<td>below</td>
<td>অলংকর</td>
</tr>
<tr>
<td>ঋ</td>
<td>ন</td>
<td>below</td>
<td>বৃত্তি</td>
</tr>
<tr>
<td>এ</td>
<td>ল</td>
<td>left</td>
<td>সোতার</td>
</tr>
<tr>
<td>ঐ</td>
<td>ল</td>
<td>left</td>
<td>ইশোধ</td>
</tr>
<tr>
<td>ও</td>
<td>লতা</td>
<td>8 at left, । at right</td>
<td>রোনন</td>
</tr>
<tr>
<td>ঔ</td>
<td>লতা</td>
<td>8 at left, । at right</td>
<td>মোঃ</td>
</tr>
</tbody>
</table>

According to the standard of Bangla Academy consonants are ordered as follows:

| অ, ই, উ, ও,  এ,  ঐ,  ও | ক | খ | গ | ঘ | ঙ | চ | ছ | জ | ঝ | ঞ | ড | ঢ | ণ | ত | থ | দ | ধ | ন | ব | ভ | ম | য | র | ল | লতা | রোনন | মোঃ |

Consonant modifiers (ie, Fala) with their corresponding consonants are listed in Table-2 [9].

Besides the vowel, consonant and their modified form we have a special character Hoshonto ( নমুনা )
Table-2: Consonant Modifiers

<table>
<thead>
<tr>
<th>Consonants</th>
<th>Consonant Modifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ন</td>
<td>ন</td>
</tr>
<tr>
<td>ব</td>
<td>ব</td>
</tr>
<tr>
<td>ম</td>
<td>ড</td>
</tr>
<tr>
<td>য</td>
<td>ই</td>
</tr>
<tr>
<td>র</td>
<td>অ, ঐ</td>
</tr>
<tr>
<td>ল</td>
<td>ল</td>
</tr>
</tbody>
</table>

Unlike English Bangla words are not only composed of individual characters placed one after another. In Bangla 2 or 3 consonant can be merged together to form a single compound character. Some examples are given in Table-3:

Table-3: Compound Characters

<table>
<thead>
<tr>
<th>Number of characters</th>
<th>Compound character</th>
<th>Decomposed form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>খা</td>
<td>ন + ড</td>
<td>ছুঁশ</td>
</tr>
<tr>
<td>3</td>
<td>পুরব</td>
<td>জ + জ + ব</td>
<td>উল্লাগ</td>
</tr>
<tr>
<td>4</td>
<td>দেহিণ</td>
<td>ন + ত + র + ন</td>
<td>বাঁধ্যু</td>
</tr>
</tbody>
</table>

3. SORTING PROBLEM OF BANGLA TEXT

English words are composed of individual alphabets and so the sorting of English words is quite simple. To sort two English words we start the comparison from the first letters of both the words and proceed towards the end of the words comparing characters pair by pair. On the basis of the first dissimilar pair of characters, a sorting decision is made.

For example, the sorting of two English words ‘ASAD’ and ‘ASIF’ is shown below:

```
   Start  End
   A     S     A       D
         A     S     I     F
```

Here the first dissimilar pair is ‘A’ and ‘I’. So decision is to be made from the comparison of these two alphabets. Since ‘A’ precedes ‘I’, ‘ASAD’ is to be placed before ‘ASIF’ in the sorted list.

In case of Bangla, the scenario is quite different. Bangla words cannot be sorted using such a simple algorithm. In Bangla words vowel and consonant modifiers are placed before, after, above or below any character. Moreover there are frequent uses of compound characters.

For the convenience of typing, all of the currently available Bangla word processors are organized in such a way that, the typist often has to type the modifiers before the consonants.

Moreover, some modifiers such as ১ and ৭ are fragmented into ১ + ১ and ৭ + ৭ respectively. Keystrokes are stored in the file following the same sequence.

For example, in case of typing সোনালী we first type ১, then ১, then ১ and so on. And in the same order the characters and modifiers are stored in the file. Here two modifiers ১ and ১ are associated with ১ but actually there is a single modifier ১১ with ১. This results in inconsistency in sorting.

Suppose two Bangla words সকাল and সোনালী are to be sorted. This will be done as described below:

Here ১ is first compared with ১. Since ১ precedes ১, ১১ comes before সকাল in the sorted list. Obviously this sorting is not correct. Because in the word সোনালী, ১ has the vowel modifier ১১ but in case of সকাল, ১ has no modifier. Hence ১১ should precede সোনালী in the sorted list if we are to follow the standard of Bangla dictionary.

4. POSSIBLE SOLUTIONS

From the proper scrutiny we find that the key problem for sorting is with the vowel modifiers and the compound characters. An efficient way of typing and a proper representation of the characters and modifiers can lead to the solution.

4.1 Method 1: as described in [1]

In order to maintain proper sorting Rahman and Iqbal [1] have proposed an internal representation of Bangla words where a dummy character is placed after the character, which has no modifier. Moreover, it is also ensured that there would be no dummy character between the constituent parts of a compound character. Again, vowel modifiers are included in the character set and they can be typed before or after the characters but for internal representation every time they are to be shifted after the character. In case of compound characters, they are decomposed into their constituent components and stored accordingly. In Table-4 internal representation of few words are shown where ৬ represents the dummy character.

For sorting the words the relative order in the character set are arranged in the following way:

Null modifier < Vowel modifiers < Vowels < Consonants

This method has the following shortcomings:

- Previously extra vowel modifiers had to be accommodated in the keyboard, which is not needed according to our opinion.
- Shifting of the vowel modifiers adds extra overhead. The keyboard interface has to be complex enough to do this job.
In the keyboard mapping proposed by them, .SH is mapped to ‘[’, .SH is mapped to ‘{’, .SH is mapped to ‘]’ and .SH is mapped to ‘}’. But these ‘[’, ‘{’, ‘]’ and ‘}’ symbols are used in Bangla. So they cannot be removed.

Due to use of the dummy character, a large amount of disk space is consumed to store Bangla words.

Table-4: Internal representation of words by [1]

<table>
<thead>
<tr>
<th>Word</th>
<th>Internal Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>অকাশে</td>
<td>অ ে কাশে খা শ ে</td>
</tr>
<tr>
<td>ছাগম</td>
<td>ছ ও গ ে তা ম ে</td>
</tr>
<tr>
<td>কর্ম</td>
<td>ক ে ম ে ল</td>
</tr>
<tr>
<td>বর্গ</td>
<td>ব ে র গ ে</td>
</tr>
<tr>
<td>মেবক</td>
<td>ম ে ব ে ক ে</td>
</tr>
<tr>
<td>কাক</td>
<td>ক ে ক ে</td>
</tr>
</tbody>
</table>

4.2 Method 2: as described in [2]

According to the proposal of Palit and Sattar [2], the keyboard will accommodate vowels, consonants and vowel modifiers. Words are typed as they are spelled. No dummy character is used. The compound characters are divided into their constituent components and stored in the file. The shape of those components will vary based on their relative position in the compound character. All the shapes are stored in the Video ROM and distinct codes are assigned against them. Internal representations of some words are shown in Table-5

Table-5: Internal representation of words by [2]

<table>
<thead>
<tr>
<th>Word</th>
<th>Internal Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>করণ</td>
<td>ক ে র ে ন</td>
</tr>
<tr>
<td>ককি</td>
<td>ক ে কি</td>
</tr>
<tr>
<td>কন্ধ</td>
<td>খ ে ন ে র</td>
</tr>
<tr>
<td>উস্ত</td>
<td>উ ে ত</td>
</tr>
<tr>
<td>অকার</td>
<td>অ ে কা র</td>
</tr>
<tr>
<td>অক্কর</td>
<td>অ ে ক্কর</td>
</tr>
</tbody>
</table>

For proper sorting the following order is followed:

Vowels < Consonants associated with the preceding part of the compound characters < Vowel modifiers < Trailing parts of the compound character.

This method has the following drawbacks:

- Both vowels and vowel modifiers are accommodated in the keyboard while the accommodation of only the vowels is sufficient.
- Since different codes are assigned to different shapes of the constituent parts of the compound character, a wide range of shapes and their corresponding codes are to be maintained.

4.3 Method 3: the proposed method

4.3.1 Keyboard Mapping

We propose that only the vowels, consonants and necessary symbols (i.e., @, & , +, = etc) will be accommodated in the keyboard. Moreover a specific key, suppose the key for G, will be used for a link character Δ. Bangla characters are mapped to appropriate ASCII values in such a way that the ordering prescribed by Bangla Academy is strictly maintained. The vowel modifiers are very frequently used. To represent a character modified by a vowel modifier, the modified character may be followed by a link character which is again followed by the vowel corresponding to that vowel modifier. This frequent use of the link character exploits significant amount of memory space.

Since there are only 10 vowel modifiers, for each modifier we can easily assign 10 distinct ASCII values higher than those of the consonants (within the available range from 128 to 255). This omits the use of the link character without hampering sorting and thus saves memory space.

4.3.2 Way of Typing

The words will be typed as they are spelled. Suppose সে is to be typed. At first we have to type স, then we will press the link key G and then এ. When any vowel is typed after the link key, the keyboard interface will store the ASCII value of the vowel modifier corresponding to that typed vowel. The internal representation of সে in the file will be সো.

While typing compound characters, for example ৷, we have to type ৷ first ন, then the link key and then ত. Keyboard interface will store the ASCII values of ৷. Link character and ত respectively. Hence, internally ত will be stored in the file as ন Δ ত.

It is the responsibility of the display interface to show সে and ত appropriately on the display unit.

The internal representations of some words are shown in Table-6.

Table-6: Representation of words by proposed method

<table>
<thead>
<tr>
<th>Words</th>
<th>Internal Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>সোনালী</td>
<td>সোনালী</td>
</tr>
<tr>
<td>সকাল</td>
<td>সকাল</td>
</tr>
<tr>
<td>সম্ভব</td>
<td>স্বচ্ছ</td>
</tr>
<tr>
<td>স্থিতি</td>
<td>স্থিতি</td>
</tr>
<tr>
<td>অন্যান্ত</td>
<td>অন্যান্ত</td>
</tr>
<tr>
<td>অন্যান্ত</td>
<td>অন্যান্ত</td>
</tr>
</tbody>
</table>
4.3.3 Sorting

When Bangla words are internally represented according to our proposed format, the sorting can be easily done with the algorithm used in sorting English words. The simple algorithm for sorting two Bangla words is given below:

1) Sorting will start with comparison of the first characters of the words and proceed comparing pair by pair towards the end.
2) If the characters mismatch, the word containing the character with higher precedence will go first.
3) If two characters being compared are same, then the following steps are followed:
   i. If these characters are the last characters of the words, then the words are same.
   ii. If the character is the last character of the first word but not of the second, then the first word will be placed before the second.
   iii. Otherwise we have to check whether the characters are followed by the link character ∆. There may be three cases:
      Case-1 (None of the characters are followed by the link character ∆):
      With the next pair of characters the operations starting from step 2 are repeated.
      Case-2 (One of the characters is followed by the link character ∆):
      The word containing the character followed by no link character will go first.
      Case-3 (Both the characters are followed by the link character ∆):
      The operations starting from step 2 are repeated with the pair of characters followed by link character ∆.

For sorting, we will follow the same order as used in Bangla dictionaries:

Vowels < Consonants < Vowel Modifiers

4.3.3.1 Algorithm for Sorting Two Bangla Words

Two Bangla words are internally organized into two strings A1 and A2. This algorithm will sort this two words.

```
Sort_Bangla(A1, A2)
1  i ← 0
2  While TRUE do
3    i ← i + 1;
4    if A1[i] = A2[i] then
5      if A1[i] = NULL then
6        message(' The words are same')
7      end if
8    else continue
9  end if
10  if A1[i] = '∆' then
11    Place A2 before A1
12    exit
13  end if
14  if A2[i] = '∆' then
15    Place A1 before A2
16    exit
17  end if
18  if A1[i] = NULL and A2[i] ≠ NULL then
19    Place A1 before A2
20    exit
21  end if
22  end if
```

4.3.3.2 Examples of Sorting

Example 1: The sorting of the two Bangla words সনালী and সকাল

```
Start   End
```

Here the starting character of সনালী and সকাল is স, but m of সকাল is not followed by link character while in সনালী m is followed by the link character. Therefore, সকাল will go before সনালী in the sorted list.

Example 2: The sorting of the two Bangla words অস্মর and অস্মর –

```
Start   End
```

Here the first mismatch occurs in the pair অ র and অ র. Since অ precedes অ, after sorting অস্মর goes before অস্মর.

Example 3: The sorting of the two Bangla words সৃষ্টি and সৃষ্টিতা

```
Start   End
```

Here the first mismatch occurs in the pair সৃষ্টি and সৃষ্টিতা.
Here the last compared character pair is same and স্থি ends while স্নি has more characters. So, স্থি goes before স্নি.

### 4.3.4 Role of Display Interface

During typing, the words are stored in the file according to the internal representation we have shown. So the file will contain the vowels, consonants and link characters. The shapes of vowels, consonants and modifiers are stored in the Video ROM. While forming compound characters, consonants take different shapes as specified in [2]. The Video ROM will also contain these shapes. The display interface is to properly display the compound characters and the characters along with their modifiers. When the link character \( \Delta \) is encountered between two characters, the Display Interface extracts the appropriate shape of the second or first character and displays with proper relative placement.

For example, কে will be stored in the file as খ আ খ. The display interface, encountering the link character \( \Delta \) between খ and আ, extracts the corresponding vowel modifier (here এ) and displays কে.

Again, incase of হ, this is stored in file as হ আ হ. Finding link character between হ and আ, the display interface will display হ, in the display unit.

Similarly, when the display interface will encounter link character between হ and আ, it will display হ.

### 5. CONCLUDING THOUGHT

In this paper we have proposed the internal representation of Bangla words which will be convenient for sorting, matching and other mathematical computation. We have included only the vowels and consonants in the keyboard, not their modified form. This will provide ease in Bangla word processing.

From the proper study of Bangla words if it is found that the modified form of certain characters are more frequently used than the characters themselves, then the keyboard may accommodate those modified forms instead of the original characters. It will obviously minimize the key-strokes. If the character itself is accommodated then the user can get the modifier by pressing that key along with the link key and vice versa.

### References