L2/07-174

The Case Folding Solution for the Arabic Script

1. The implementation of the Arabic letters in their atomic form is of utmost importance and must be implemented as soon as possible.
2. It has many benefits including the universality to the Arabic coverage of the block, limiting the block explosion, providing the ease in data entry operations especially on limited devices.
3. Further more, the issues regarding the development of OCR products that may emit the partial recognized character sequences is almost impossible without the support for separate nuqta characters.
4. But encoding them introduces some normalization issues.
5. Let's not call the two possible encodings are normalization forms.
6. These are SPECIAL CASES of the same characters!
7. The normalization transformations are not of the transient nature, but these transformations are! A user is expected to type in a hybrid of the both of the forms.
8. Let's start from the character. A character may be in either of the two cases
    a. Collapsed Case
    b. Spread Case
    c. All the base characters containing a diacritic (nuqta) are in collapsed case
    d. All the base characters without a diacritic (nuqta) are in spread case
    e. All the diacritics (nuqta) are in spread case
9. A series of the characters, that is string, may be in either of the cases:
    a. Collapsed Case: Only having collapsed characters
    b. Spread Case: Only having spread characters
    c. Hybrid: A mixture of collapsed and spread characters
10. There exists two string functions for converting in either cases:
    a. Collapse(S): Will return collapsed case of a string S
    b. Spread(S): Will ALWAYS return spread case of string S
11. Collapse(S)
    a. May not always return Collapsed Case if for some sequence of characters, there do not exist any corresponding collapsed case character. In such a case, the returned string will be in Hybrid Case.
    b. Will look for spread characters having a series of diacritics after them and then will look their mapping for transformation. If mapping does not exist, leaves as it is.
    c. Has idempotent property. That is: Collapse(Collapse(S)) = Collapse(S)
    d. Is a context dependent function
    e. Is a buffer length preserving function
12. Spread(S)
    a. Will always return the Spread Case.
    b. Will look for composite characters and emits decomposed sequences.

c. Has idempotent property. That is: Spread(Spread(S)) = Spread(S)
d. Is context independent
e. Is a buffer length preserving function

13. It might be the case that Collapse(S) === Spread(S)
    a. Will happen when:
        i. S has ONLY the base characters which are already in Spread case
        ii. Or S has ONLY the base characters and diacritic (nuqta) sequences which do not have corresponding collapsed characters
        iii. If either (i) or (ii) alone, or in conjunction are satisfied, then Collapse(S) === Spread(S)
14. Sometimes Collapse(H) === H but in any case Spread(H) != H where H is a hybrid string.
15. Such an implementation operation does not harm the existing conventions.
16. The existing text will be completed in Collapsed Case.

7.a: Collapsed Case Characters

ب، پ، ج، ڈ

7.b: Spread Case Characters

ا، ر، و، ب◌، ب◌پ، ح◌ب، د◌ٹ

8.a: Collapsed Case String

پ اک س ت ان = پاکستان

8.b: Spread Case String

ب◌پ اک س ب◌تان◌ = پاکستان

8.c: Hybrid Case String

پ اک س ت ا ں◌ = پاکستاں

12: Collapse(S) === Spread(S) (Sometimes may be the case)

Collapse(اردو) === Spread(اردو)

13.a: Collapse(H) === H (sometimes may happen where H is a hybrid string)

Collpase(پاکستاں) == پاکستاں

13.b: Spread(H) != H (ALWAYS is the case where H is a hybrid string)

Spread(پاکستاں) == پاکستاں