UCA: remove variable-weighting=IgnoreSP option

Author: Markus Scherer
Date: 2012-aug-29

Proposal

Remove the variable-weighting=IgnoreSP option from the UCA (UTS #10)

- In 3.6.2 Variable Weighting, remove option 4. IgnoreSP and adjust related text such as “There are five possible options for variable weighted characters”.
- Instead, add a brief note about IgnoreSP in UCA 6.1 & 6.2 like “UCA 6.1 and 6.2 also defined an IgnoreSP option which was a combination of Shifted with changing the set of variable collation elements to only those for space and punctuation.”
- Change the IgnoreSP column heading in Table 13 Comparison of Variable Ordering to “CLDR root”, or else remove this column. Adjust or remove the following explanatory text for that column, as appropriate.

Rationale

UCA 6.1 added the variable-weighting=IgnoreSP option with the intention to describe the LDML default behavior. However, it is unnecessary, confusing, and troublesome to document.

- It mixes the variable-weighting (alternate-handling) parameter with changing the range of variable primaries; those should be orthogonal.
- It was added to say what LDML does but it's useless for LDML itself: LDML changes the default-variable-primaries range, therefore in LDML the Shifted option has the exact same behavior that IgnoreSP would have.
    - Since LDML does not need IgnoreSP, the LDML spec (UTS #35) does not mention it.
    - LDML 22 does say that its default-variable-primaries range differs from that in the DUCET.
- UCA+allkeys.txt does not specify the set of variable primary weights for IgnoreSP, so non-LDML UCA implementers are left guessing which characters/weights are variable, if they want to implement IgnoreSP.
- allkeys.txt has syntax (the *) for default-variable primaries but no syntax for IgnoreSP-variable.
    - We could pick a character or weight and document it, but we would have to be careful to keep that up to date.
    - We could design some algorithm, but since the punctuation and symbol characters are not guaranteed to be in single, contiguous DUCET blocks, we would at least have to review the result each time.