Saxon - Bug #1842

Greek perispomeni and normalize-unicode

2013-07-14 21:16 - Michael Kay

Status:	Closed	**Start date:**	2013-07-14
Priority:	Normal	**Due date:**	
Assignee:	Michael Kay	**% Done:**	100%
Category:	XPath conformance	**Estimated time:**	0:00 hour
Sprint/Milestone:		**Spent time:**	0:00 hour
Legacy ID:		**Fix Committed on Branch:**	
Applies to branch:		**Fixed in Maintenance Release:**	

Description

Raised by Ryan Baumann on the SourceForge saxon-help list.

Various forms of characters with perispomeni seem to be handled incorrectly with normalize-unicode (running as XSLT 2.0 in Saxon HE 9.5.1.1).

normalize-unicode('ῇ̓','NFC') (U+03B7 U+0342 U+0313 U+0345) is ῇ̓ (U+1FC6 U+0313 U+0345)

correct NFC: ῇ̓ (U+1FC7 U+0313)

normalize-unicode('ῇ̓','NFD') (U+1FC7 U+0313) is ῇ̓ (U+03B7 U+0342 U+0345 U+0313)

normalize-unicode('ῇ̓','NFD') (U+1FC6 U+0313 U+0345) is ῇ̓ (U+03B7 U+0342 U+0313 U+0345)

Other instances of incorrect NFC normalization (normalize-unicode on these characters is idempotent):

ῇ̔ (U+1FC6 U+0314 U+0345) should be ῇ̔ (U+1FC7 U+0314)

ῷ̔ (U+1FF6 U+0314 U+0345) should be ῷ̔ (U+1FF7 U+0314)

Ὧ (U+1F69 U+0342) should be Ὧ (U+1F6F)

Ἆ (U+1F08 U+0342) should be Ἆ (U+1F0E)

Checked against both Java's java.text.Normalizer and Perl's Unicode::Normalize as my references for "correct" NFC normalization.

The problem seems to be fairly general for any character which has a pre-combined perispomeni form. There are probably others than just what's here, you can see the results of running java.text.Normalizer against a large corpus of Ancient Greek that has already been passed through normalize-unicode in this commit:

https://github.com/ryanfb/idp.data/commit/bcb7dd6223fb50c48f62027761e8deced2574ed7

-Ryan

History

#1 - 2013-07-14 21:24 - Michael Kay

My initial suspicion was that this might be a question of which Unicode version is in use, but using tables generated from Unicode 4.0.0 as against Unicode 6.2.0 does not appear to give any difference in the results, and indeed inspection of the UnicodeData.txt file suggests no obvious difference between versions for the relevant characters.

What does seem relevant is that (taking an example) the entry for 1F0E in UnicodeData.txt shows a decomposition to 1F08 0342, where 1F08 itself can be further decomposed to 0391 0313. It looks as if the Saxon data tables generated from UnicodeData.txt don't take this double decomposition into account.

#2 - 2013-07-14 21:40 - Michael Kay

Note: The original Java code from the Unicode consortium on which the Java implementation is based has been withdrawn:

http://www.unicode.org/reports/tr15/Normalizer.html

It's therefore possible that the code contains bugs which have not been fixed.

#3 - 2013-07-14 23:17 - Michael Kay

It appears that Saxon 9.1 got this right. Saxon 9.1 generated the normalization data into a Java module UnicodeData.java rather than into the XML data file normalizationData.xml which is used in more recent releases.

#4 - 2013-07-15 00:24 - Michael Kay

Noted that in Saxon 9.1, decompose(codepoints-to-string((7944,834))) yields (913, 787, 834) whereas in 9.5, it yields (913, 834, 787). The failure to sort the modifiers into canonical order then causes a failure to compose the pairs during the composition phase.

Debugging shows that in 9.1, data.canonicalClass(834) is 230, while in 9.5, the same expression returns 220. This seems to account for the difference in the result of the sort. 834 = 0x342, for which Unicode 3.0 UnicodeData.txt and subsequent versions all have

0342;COMBINING GREEK PERISPOMENI;Mn;230;NSM;;;;;N;;;;;

where the field 230 is the canonical class. So the data tables in 9.5 appear to be wrong.

#5 - 2013-07-15 00:39 - Michael Kay

There seems to be a problem with either the writing or the reading of the run-length-encoded CanonicalClassValues list, which was newly introduced when the change was made to put the data in XML format.

#6 - 2013-07-15 10:06 - Michael Kay

- *Status changed from In Progress to Resolved*

The data was being read incorrectly; an implicit run-length was being interpreted as zero rather than 1. A patch has been committed for 9.4, 9.5, for Saxon-CE 1.1 (bug 1843), and for the development branch.

#7 - 2013-08-29 18:45 - O'Neil Delpratt

- *Status changed from Resolved to Closed*
- *% Done changed from 0 to 100*
- *Fixed in version set to 9.5.1.2*

Bug fix applied in the Saxon maintenance release 9.5.1.2