L2/98-262

Feedback on BiDi

A. UTC #76 Unicode Collation Algorithm & Bidi (from Mark Davis, IBM)

From: kenw@sybase.com (Kenneth Whistler)
Reply-To: unicore@unicode.org
To: Multiple Recipients of Unicore <unicore@unicode.org>
Date: Thu, 23 Apr 1998 15:53:15 -0700 (PDT)
Subject: UTC #76 Unicode Collation Algorithm & Bidi

Forwarding undelivered mail from Mark Davis:

Date: Thu, 16 Apr 1998 07:43:28 -0800
From: Mark Davis <marked@best.com>
Reply-To: mark@macchiato.com
To: unicore@unicode.org
Subject: UTC #76 Unicode Collation Algorithm & BIDI

I request that the following be distributed at the UTC.

Mark

P.S. I am disappointed that the meeting was scheduled when I could not attend, since there are a number of issues that I am involved in. Unfortunately, I will not even be reachable by phone after tomorrow.

====================

A. The results of the BIDI ad hoc are on the ftp site at:

ftp://unicore:unicore@ftp.unicode.org/WorkingGroups/Bidi/MED/BIDIAlgorithm.rtf

Could someone please print this up for distribution at the meeting?

B. I don't see the Unicode Collation Algorithm on the agenda (on http://www.unicode.org/unicode/members/meetings/agenda-76p.html), although I thought it was to be discussed.

I have two proposed changes to DUTR #10 before it is made a UTR, based on feedback from some database people internally. Both are fundamentally clarifications, and help to reduce any preceived conflict with 14651.

1. Add just before Conformance:
"The UCA does not mandate the use of the precise data format or tailoring format. As long as the implementation provides the ability to produce the same results in the comparison of strings, the data format and tailoring format can vary."

Add to the end of Conformance Clause 1 and Clause 2:
", or equivalent data."

2. Add to the end of Tailoring (just before Implementation Notes)

Customization

Implementations of the algorithm may allow users (through an API) to specify variations on the algorithm that have a broad effect on the results, without tailoring all of the characters involved. Examples of these include:

1. Rearrangement of the order of particular scripts.
2. Allowing the choice of whether uppercase comes before lowercase, or vice versa.
3. Condensing particular values within a level, such as reducing the differences in tertiary tags to just case differences.
4. Removing particular levels. This can have two forms:
a. completely ignoring the level (as discussed under Searching in Implementation Notes).
b. treating lower-level differences as if they were combined with the next highest level. For example, the level 4 differences can be merged with the level 3 differences. In such a case the level 4 differences are not ignored; they are just treated as if they were level 3 differences along with the other level 3 differences.

B. BiDi Reference Code (from Andy Daniels, Apple)

From: David Goldsmith <goldsmith@apple.com>
Reply-To: unicore@unicode.org
To: Multiple Recipients of Unicore <unicore@unicode.org>
Cc: "Andy Daniels" <amd@apple.com>
Date: Thu, 23 Apr 1998 15:25:12 -0700 (PDT)
Subject: Fwd: Re: Fwd: BIDI reference code

Hi all,

Here are some (belated) minor comments by Andy Daniels of Apple on the corrections to the Bidi algorithm which were sent out in February. Andy is working on Apple's implementation.

David Goldsmith
Architect
Text and International Department
Apple Computer, Inc.
goldsmith@apple.com
---------------- Begin Forwarded Message ----------------
Date: 4/22/1998 7:17 PM
Received: 4/23/1998 8:43 AM
From: Andy Daniels, amd@apple.com
To: David Goldsmith, goldsmith@apple.com

>(B)
>
>[Proposed corrigendum B1 corrects a typographical error.
>Proposed corrigenda B2 and B3 clarify that embedding and
>override codes and their matching PDFs are not themselves
>set to the level of the text they enclose.]

[...]

>After the last sentences of rules E2, E3, O2, and O3, add:
>
>"The formatting code itself retains the pushed level, and is not
>assigned the new level.
>
>(B3)
>
>After the first sentence of rule T4, insert:
>
>"The PDF itself is assigned the popped level (the one in effect
>before the embedding or override)."

These proposed corrigienda make it a bit more of a nuisance to implement the last part of rule T6:

>"T6. In the following rules, an embedding or override code
>and its matching PDF act as if they were strong characters
>of the appropriate type. All unmatched PDFs are ignored.
>If two embeddings with the same level are adjacent, then the
>PDF terminating the first embedding and the code initiating
>the next embedding are ignored."

It's a bit simpler to check that the new embedding takes you to the level of the preceding embedding/override if the PDF is set to the higher embedding level. Then you only need to check the level that you're recording anyway instead of having to keep yet another piece of state around. It's a minor thing, obviously.

-- Andy. --
----------------- End Forwarded Message -----------------

C. BiDi Algorithm (from Jony Rosenne)

From: Jonathan Rosenne <rosenne@netvision.net.il>
Reply-To: unicore@unicode.org
To: Multiple Recipients of Unicore <unicore@unicode.org>
Date: Thu, 23 Apr 1998 21:51:14 -0700 (PDT)
Subject: Bidi Algorithm

1. Invalid levels:

This is a lot of unnecessary complications. Why noy just ignore invalid RLE's and LRE's? I know it will mess up the PFFs, but who cares?

I am not repeating my objection to specifying the algorithm in this case, I am just saying that if it is felt that it must be specified then it should be done in the simplest possible way. Adding a new category to handle something that is not supposed to happen - is this reasonable?

If the limit of 15 is too low, it can be increased without affecting existing implementations.

2. Overrides

I think it was a mistake to treat overrides as if they were embeddings. I think at the time we thought of it as a neat solution, but we gave no thought to the question of reaching the limit of levels under the assumption that it had been set so high that it is practically infinite (I still think it is).

But overrides are not embeddings. An override can be processed by simply overriding the category of all the characters up to the next PDF, and then dropping the override and the PDF. If the algorithm specification has to be changed, then I suggest changing it this way.

Jony