Bidirectional Reordering Algorithms
L2/01-218
Steve Atkin
IBM
Ryan Stansifer
Florida Tech
Error

• Paper contains an error
  – No difference between the C/C++ and Java implementations
  – Paper will be corrected
    • Tables 3 and 7, rows 7 and 11
  – Explanation
    • Test case run using Hebrew input rules instead of Arabic input rules
Bidi Revisited

• We studied existing bidi algorithms
  – FriBidi, PGBA, UCDATA, ICU, Java, Unicode

• We created an alternative reference
  – Functional language (Haskell)
  – Mathematical description

• We tested conformance
  – Standardized testing conventions
  – Provided additional test cases
Currently

• All use an imperative language
  – C, C++, Java
  – Complex code

• Few public test cases exist

• Conformance
  – Difficult to determine
Printed Description

• Issue 1
  – “From the highest level found in the text to the lowest odd level on each line, reverse any contiguous sequence of characters that are at that level or higher.”
  • English is imprecise
Printed Description

• Two possible interpretations
  – Next level to be processed is one less than the current level.
  – Next level to be processed is the next lowest level actually present in the text.

• Recommendation
  – Provide clarification, possibly an example
Input Conventions

• Issue 2
  – Not every implementation uses the same input conventions.
    • Nevertheless, they all support the standard types.
  – We modified the input conventions when the algorithm had no such support.
    • Unicode data table – changed attributes
  – Simplified comparing output.
<table>
<thead>
<tr>
<th>Type</th>
<th>Arabic</th>
<th>Hebrew</th>
<th>Mixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>a - z</td>
<td>a - z</td>
<td>a - z</td>
</tr>
<tr>
<td>AL</td>
<td>A - Z</td>
<td></td>
<td>A - M</td>
</tr>
<tr>
<td>R</td>
<td>A - Z</td>
<td></td>
<td>N - Z</td>
</tr>
<tr>
<td>AN</td>
<td>0 - 9</td>
<td></td>
<td>5 - 9</td>
</tr>
<tr>
<td>EN</td>
<td>0 - 9</td>
<td></td>
<td>0 - 4</td>
</tr>
</tbody>
</table>
Input Conventions

• Recommendation
  – Work with implementers to adopt a standard set of input conventions
    • UCDATA
    • FriBidi
    • PGBA
Test Cases

• Issue 3
  – Inadequate public test cases
  – Unicode - no public test cases
  – PGBA - Arabic and Hebrew test cases
  – Haskell Bidi - Arabic, Hebrew, and Mixed
    • Based on ambiguities and other problem areas

• Recommendation
  – Make Unicode’s test cases publicly available
Test Results

• Issue 4
  – PGBA and FriBidi do not claim conformance
  – Both accept Unicode as input
  – PGBA and FriBidi
    • Differences
    • Tables 7, 8, and 9
Test Results

• Why are there differences?

• Incorrect assumptions
  • AL = R
  • AL, is not a strong type

• Recommendation
  – Find out why they do not wish to conform.
  – May require Unicode to make changes to the Bidi algorithm.
Normalization

• Issue 5
  – Unexpected interaction between normalization and the Bidi algorithm.
  – Logical order
    • U0627,U2116,U0031,U0032,U0033
    • 123? ?
  – NFKC
    • U0627,U004E,U006F,U0031,U0032,U0033
    • No123 ?
Normalization

• Issue 5
  – Problems with quantities
  – Logical order
    • U0627,U00BC
    • ¼ ?
  – Normal form KC
    • U0627,U0031,U2044,U0034
    • 4/1 ?
Normalization

• Recommendation
  – Document the interaction between normalization and the Bidi algorithm.
    • UAX #9
Normalization

• Canonicalization safe?
  – Appears to be safe
  – Requires
    • precomposed character’s type = decomposed base character’s type
Boundary Neutrals

• Issue 6
  – Some boundary neutrals should never be reordered, even if an implementation retains formatting codes.
  – Lang tags
    • Logical order
      – U0624,UE0001,UE0075,UE0072,U0623
      – Lang ur - Urdu
    • Display order
      – U0623,UE0072,UE0075,UE0001,U0624
      – Lang ru – Russian
Boundary Neutrals

• Recommendation
  – Document this interaction in UAX #9
Domain Names

• Issue 7
  – Applying the bidi algorithm to domain names produces strange results.

• What happens if we use the Bidi algorithm?
  – ABC.ibm.com (logical order, Arabic)
  – ibm.com.CBA (display order)
  – Period used as punctuation in Bidi algorithm

• Bidi algorithm mangles domain hierarchy
  – Left to right reading
  – Specific to general
Domain Names

• Suppose you apply algorithm to each label
  – NOP--123 (logical order, Hebrew)
  – --123PON (display order)
  – European terminator
• Bidi assumptions inappropriate
• In domain names “-”
  – White space, predominate usage
  – 123--PON (display order)
Domain Names

• Recommendation
  – Treat “-” as white space in domain names
  – Consider allowing an alternative output
    • Rule L2
  – Use controls (LRO, RLO, LRE, RLE, PDF)
    • Embed controls in the domain name for display purposes only
    • Keep output in logical order