L2/11-377

Response to PRIs
* 205 Proposed addition of AL MARK and LEVEL DIRECTION MARK
* 188 Proposed Update UAX #9: Unicode Bidirectional Algorithm
* 185 Extension of UBA for improved display of URL/IRIs

Preliminary proposal for UBA v.2

Kent Karlsson
2011-10-24

Introduction

The current bidi algorithm does not perform well in far too many instances for it to be generally recommendable as it is. It is sometimes said that the current bidi algorithm was made for "ordinary text", i.e. paragraphs of text in a natural language like this paragraph. However, it fails much too often when it comes to such things as parentheses or quotes (which may contain text in a different script of different directionality than the surrounding text) or even plain sentence structure indicated by punctuation (full stop, question mark, comma, ...). Instead it focus much attention to handle signs near numbers. But it does so in quite an arbitrary manner, and further, the placement is locale dependent or even an individual preference, and should not at all be codified in the bidi algorithm. So not even for ordinary text does the current bidi algorithm give a sensible result (examples are easy to find elsewhere and are not given here). It is even worse for formal syntaxes that are not natural language, such as URLs, simple linear math expressions, or XML source files. Even though doing a bidi LRO override (via a higher level protocol; a setting that should be available in text editors, including web browsers) would help, especially for those who are not familiar with bidi scripts, that can be overbearing for those who are familiar with bidi scripts. In particular, doing such an override is not suitable for "ordinary text". The points below outline a set of remedies for these problems.

Note that even though the current bidi algorithm does provide for "higher level protocol overrides", in particular HL3, this has not been as popular as it should be, and instead there has been a focus on "doing bidi exactly as in UBA **without** any modifications" (such as UBA HL3 settings), or at most allowing HTML documents to modify bidi according to the HTML *dir* attribute (including the *bdo* element), which map to inserting certain bidi control characters. But nothing beyond that. Inserting invisible control characters "by hand", and especially doing so systematically is just not workable. Also adding HTML bidi markup systematically (for quotes, parentheticals, etc.) is unlikely to happen.

For “ordinary text” the bidi algorithm should give good results in the vast majority of cases also without bidi control characters (or directional markup in HTML). This calls for a UBA version 2 (as hinted in PRI 205, but for a much more restricted examle). In addition at least some special syntaxes should have predefined recommendation for “higher level” bidi changes.

Notational note

In discussion and description of the bidi algorithm, some abbreviations are used both for referring to characters and for referring to bidi categories. In contrast to UBA (v.1), in this proposal there is no 1-1 correspondence for DDE (new), ELS (new), and PDF with respect to characters and bidi categories. E.g., many characters will have the bidi category PDF, not just the bidi control character POP DIRECTIONAL FORMATTING (PDF).

Proposal

The proposal is divided up in a number of related points.

1. Add a bidi category DDE, *Derived Direction Embedding*.

Rationale: The DDE bidi category is intended for certain graphical characters (see below). DDE works like LRE or RLE, but which one depends on the implicit directionality of the embedding. The implicit directionality of an embedding is derived from the first (in logical order) bidi strong character after a bidi DDE category character but before any bidi DDE, LRE, RLE, LRO, RLO, or PDF category character. The derived direction defaults to the enclosing directionality if no bidi strong character is encountered in the specified range.

((**1b.** Add a bidi category EDE, *Enclosing Direction Embedding*? Don’t yet have a clear use case for this one...))

2. Add a bidi category ELS, *Embedding Level Separator* (instead of LDM, Level Direction Mark, suggested in PRI 205).

Rationale: The ELS bidi category is intended for certain graphical characters (see below), certain character strings (see below), and a new bidi control character. Characters of this category (original value, not after bidi reassignment steps) are (re)set to the enclosing explicit embedding level. Thus this category works similar to the bidi category S, but does not go all the way up to the paragraph level, but just to the enclosing embedding/(override) level. Note that this category is not only of the embedding direction (as the PRI proposed LDM), but decreases the bidi level to the closest enclosing embedding/override level. This is in order to guarantee the proper behaviour of ELS bidi category characters in all cases. Note that CHARACTER TABULATION is of bidi category S, better called *Paragraph Level Separator* (rather than just *Separator*).

3. Add a bidi control character with the bidi category ELS.

Rationale: Some graphical characters that are not (will not be) of ELS bidi category, should in some cases be made ELS also in the absence of a higher-level protocol to set them to the bidi category ELS. For instance / in date expressions that are expected to be used in either bidi direction. This can then be achieved by placing an ELS control on either side of each / in the date expression.

4a. Abolish rule X9.

Rationale: The bidi categories will now be applied also to non-control characters, which must not be removed.

4b. Abolish the part of rule X6 that starts with "besides". SEE ALSO POINT 6. (BN?)

Rationale: The DDE/LRE/RLE/LRO/RLO, ELS, and PDF characters must get the **enclosing** embedding level so that they are not moved around illogically.

5. Make all characters of general category Ps bidi DDE. Make all characters of general category Pi default bidi DDE. Make all characters of general category Pe bidi PDF. Make all characters of general category Pf default bidi PDF. INTERLINEAR ANNOTATION SEPARATOR to be a DDE and INTERLINEAR ANNOTATION TERMINATOR to be a PDF. (Note though that "complex ruby", http://www.w3.org/TR/ruby/, has a more general mechanism, <rt> would by default be be DDE, and </rt> would be PDF, similarly for rb or at least rb to be ELS , so that the order of base elements and ruby elements is consistent, namely that of the enclosing bidi embedding. See also point 8b below.)

Rationale: A parenthetical or quote should be kept together in a single span, and not be split up in several discontiguous spans, and each parenthetical/quote be bidi analysed on its own. Only if there is no strong indication to the contrary, should it assume the direction of the enclosing text.

6. At stage L (or, better, already at initial level assignment, and preferably also for category S, provided that this assignment is not changed later by any bidi step), (re)set the level of DDE/LRE/RLE/LRO/RLO, ELS (as mentioned), and PDF to the **enclosing** embedding/override level (not always the level adjacent to the characters in question, but defaults to the paragraph level when there is no bidi embedding/override). Thus an ELS control character works just like the control character sequence <DDE,PDF> (and <LRE,PDF>, ...).

Rationale: After a beginning parenthesis (and similar characters), one should automatically start a bidi embedding, the bidi direction should be reevaluated, and the resulting direction “pushed”. Just before the ending parenthesis (and similar characters), the bidi level should be popped. The parentheses (and similar characters) themselves should however be seen as part of the surrounding text, and should therefore have the

embedding level of the surrounding text's explicit embedding level. This way bidi text will work in a more predictable manner, and heavily lessens the need for manual insertion of invisible bidi control characters. The latter is a process which is highly impractical in a text editor, and even with some support is very hard to get right.

7. Let the characters in the set Dash-Hyphen-"SMALL HYPHEN-MINUS"-"FULLWIDTH HYPHEN-MINUS"-Sm+"INTERLINEAR ANNOTATION ANCHOR" (but see point 9 below regarding Sm) behave as bidi ELS. Let the strings in the set <Space+bos, * HYPHEN-MINUS, Space+eos> behave as bidi <ELS,ELS,ELS>. Maybe the same for strings in the set <Space, SOLIDUS, Space>. (Where eos is end of string, including end of paragraph and end of bidi embedding (PDF) and where bos is beginning of string, including beginning of paragraph and beginning of bidi embedding.)

Rationale: This is in order to let in-paragraph lists maintain their order according to the current embedding level. An alternative would be to let these characters (and strings) assume the S bidi category, but that always uses the paragraph bidi level, disregarding embeddings, and that is likely to be inappropriate for in-line quotes and parentheticals.

8. Let the characters in the set Terminal_Punctuation, **if** followed by a character in Space+eos+Terminal_Punctuation behave as bidi ELS. Let the characters in the set "INVERTED/TURNED" Terminal_punctuation (such as INVERTED QUESTION MARK), **if** preceded by a character in Space+bos+"INVERTED/TURNED" Terminal_Punctuation behave as bidi ELS.

Rationale: This is in order to maintain the sequencing of sentence components. An alternative would be to let these characters assume the S bidi category, but that always uses the paragraph bidi level, disregarding embeddings, and that is likely to be inappropriate for in-line quotes and parentheticals.

8b. (Not for UTC, but... In HTML, all begin and end tags should be interpreted (in addition to their normal HTML meaning) as at least bidi ELS. Note that some tags are for bidi interpreted as DDE/LRE/RLE/LRO/RLO (for begin tags) and PDF (for end tags), which (now, according to this proposal) imply ELS functionality. *Rationale*: Keep text that has been marked as connected together and not split into discontiguous pieces. HTML display engines should allow for UBA v.1 as well as UBA v.2 as a user setting with the latter as default.)

9. Let all characters that have general category Sm, **except** the six listed below (which are letter-like), behave as bidi ELS.

```
03F6;GREEK REVERSED LUNATE EPSILON SYMBOL;Sm;0;ON;;;;;N;;;;;
2118;SCRIPT CAPITAL P;Sm;0;ON;;;;;N;SCRIPT P;;;;
2141;TURNED SANS-SERIF CAPITAL G;Sm;0;ON;;;;;N;;;;;
2142;TURNED SANS-SERIF CAPITAL L;Sm;0;ON;;;;;N;;;;;
2143;REVERSED SANS-SERIF CAPITAL L;Sm;0;ON;;;;;N;;;;;
2144;TURNED SANS-SERIF CAPITAL Y;Sm;0;ON;;;;;N;;;;;
```

Rationale: This is in order to maintain the sequencing of simple linear math expressions. An alternative would be to let these characters assume the S bidi category, but that always uses the paragraph bidi level, disregarding embeddings. However, the excluded six symbols are letter-like, and should not act as sequence separators. These six would be better as general category Lo and bidi L.

10. Special syntax (a.k.a. Higher Level Protocols in Unicode) will still need special handling for bidi, depending on which special syntax that is at hand. Here are some examples of modifications to bidi in order to handle these better in bidi (the suggestions here are tentative):

- (Known to be) date format with /: / is ELS.
- (Known to be) date format with -: - is ELS.
- (Known to be) date format with .: . is ELS.
- (Known to be) date format with ARABIC DATE SEPARATOR: ARABIC DATE SEPARATOR is ELS (but this should be the default for ARABIC DATE SEPARATOR).
- (Known to be) Unix file name: / is ELS.
- (Known to be) Windows file name: : and \ are ELS.
- Domain name: . and - are ELS, implicit DDE before and implicit PDF after; maybe have implicit DDE+PDF if the domain name is not part of an email address or an IRI.
- Email address: < is DDE (Derived Direction Embedding) (implicit if missing); > is PDF (implicit if missing); ., -, and @ are ELS.
- IRI: Domain name part: handle as domain name; rest: ":", "/", "?", "#", "@", "!", "$", "&", "'", "*", "+", ",", ";", "=" are ELS. (%-escapes?) Implicit DDE+PDF around the IRI (or use EDE, Enclosing Direction Embedding?).
- XML source: < is LRE or RLE (depending on global preference); > is PDF, :, = are ELS; " after = is DDE, other " is PDF; whitespace between attributes: ELS; tabs to be considered as space; newlines (all varieties): space (WS) or "paragraph" boundary (B) depending on global preference.
- XHTML, HTML source: as XML source; but in addition , , any table related begin/end markup (in particular <td>, </td>) are seen as paragraph separators; attribute dir='...' be heeded for encompassed text (not for encompassed markup, incl. attribute values).

- HTML/XML display (not source display): all begin and end tags should be interpreted (also, i.e. in addition to their normal HTML interpretation as begin table, begin link, etc. etc.) as at least ELS (some begin tags interpreted (also) as DDE/LRE/RLE/LRO/RLO which implies ELS functionality, some end tags interpreted (also) as PDF, which implies ELS functionality).

Rationale: Some textual expressions aren't ordinary sentences (with their structure, using terminal punctuation, parenthesis, quotes, etc.), but instead have a different way of being parsed. An example above was date expressions with / where the / (solidus) is more of a syntactic separator than usual, and the direction should not depend on the constituents of the date expression (digits, month names), but depend solely on the embedding level and its direction. While one could insert bidi controls also for the more complex cases; e.g. (Unix) file names being very similar to date expression using /; many of them are also more complex (and file name should not contain bidi control characters) and may occur in places where they may be autodetected (if not outright explicit) to be of one or other more complex expression, such as extracted from mark-up, or in certain UI positions (such as an address bar). In these cases it is helpful to act as if the syntax had a more expected influence on the bidi algorithm. E.g., having ELS at expected places w.r.t. the syntax, and also insert implicit LRE, RLE, or DDE, and PDF around, or even inside, these expression.

11. Questionable bidi categories: AL, ES, CS, EN, AN, ET.

The existence of these bidi categories seem to be an attempt to automate something that should not be automated, not in that way at least. They add complexity, for an unexplained use-case, and seem arbitrary; I have the impression that they are quite arbitrary and far from natural even to people familiar with the bidi algorithm and even more unnatural and arbitrary for people “just” used to reading text in a bidi script. These categories should be simplified away, and replaced by L, R, ON, and ELS as appropriate. This would make the result of the bidi algorithm easier to predict, more natural, and would make the algorithm itself easier to understand for all (and for “ordinary” users in particular). The difference these categories induces in UBA v.1 should be handled by other, more explicit, means.

AL:
All AL should be R.

ES:

```
002B;PLUS SIGN;Sm;0;ES;;;;;N;;;;;
002D;HYPHEN-MINUS;Pd;0;ES;;;;;N;;;;;
207A;SUPERSCRIPT PLUS SIGN;Sm;0;ES;<super> 002B;;;;N;;;;;
207B;SUPERSCRIPT MINUS;Sm;0;ES;<super> 2212;;;;N;SUPERSCRIPT HYPHEN-
  MINUS;;;;
208A;SUBSCRIPT PLUS SIGN;Sm;0;ES;<sub> 002B;;;;N;;;;;
208B;SUBSCRIPT MINUS;Sm;0;ES;<sub> 2212;;;;N;SUBSCRIPT HYPHEN-MINUS;;;;
2212;MINUS SIGN;Sm;0;ES;;;;;N;;;;;
FB29;HEBREW LETTER ALTERNATIVE PLUS SIGN;Sm;0;ES;<font> 002B;;;;N;;;;;
FE62;SMALL PLUS SIGN;Sm;0;ES;<small> 002B;;;;N;;;;;
FE63;SMALL HYPHEN-MINUS;Pd;0;ES;<small> 002D;;;;N;;;;;
FF0B;FULLWIDTH PLUS SIGN;Sm;0;ES;<wide> 002B;;;;N;;;;;
FF0D;FULLWIDTH HYPHEN-MINUS;Pd;0;ES;<wide> 002D;;;;N;;;;;
```

Several of these would be ELS (for * HYPHEN-MINUS if surrounded by spaces) according to the proposal above. * HYPHEN-MINUS should otherwise be ON (neutral). See point 9 above.

CS:

```
002C;COMMA;Po;0;CS;;;;;N;;;;;
002E;FULL STOP;Po;0;CS;;;;;N;PERIOD;;;;
002F;SOLIDUS;Po;0;CS;;;;;N;SLASH;;;;
003A;COLON;Po;0;CS;;;;;N;;;;;
00A0;NO-BREAK SPACE;Zs;0;CS;<noBreak> 0020;;;;N;NON-BREAKING SPACE;;;;
060C;ARABIC COMMA;Po;0;CS;;;;;N;;;;;
202F;NARROW NO-BREAK SPACE;Zs;0;CS;<noBreak> 0020;;;;N;;;;;
2044;FRACTION SLASH;Sm;0;CS;;;;;N;;;;;
FE50;SMALL COMMA;Po;0;CS;<small> 002C;;;;N;;;;;
FE52;SMALL FULL STOP;Po;0;CS;<small> 002E;;;;N;SMALL PERIOD;;;;
FE55;SMALL COLON;Po;0;CS;<small> 003A;;;;N;;;;;
FF0C;FULLWIDTH COMMA;Po;0;CS;<wide> 002C;;;;N;;;;;
FF0E;FULLWIDTH FULL STOP;Po;0;CS;<wide> 002E;;;;N;FULLWIDTH PERIOD;;;;
FF0F;FULLWIDTH SOLIDUS;Po;0;CS;<wide> 002F;;;;N;FULLWIDTH SLASH;;;;
FF1A;FULLWIDTH COLON;Po;0;CS;<wide> 003A;;;;N;;;;;
```

Several of these, IF followed by space, would be ELS (except SOLIDUS, FRACTION SLASH, and * NO-BREAK SPACE) according to the proposal above. If not followed by space, they should be ON (neutral). FRACTION SLASH should be ELS (as other math operator symbols). See point 8 above.

EN:

…

Turn into L, in order to simplify matters.

AN:

```
0600;ARABIC NUMBER SIGN;Cf;0;AN;;;;;N;;;;;
0601;ARABIC SIGN SANAH;Cf;0;AN;;;;;N;;;;;
0602;ARABIC FOOTNOTE MARKER;Cf;0;AN;;;;;N;;;;;
0603;ARABIC SIGN SAFHA;Cf;0;AN;;;;;N;;;;;
0604;ARABIC SIGN SAMVAT;Cf;0;AN;;;;;N;;;;;
06DD;ARABIC END OF AYAH;Cf;0;AN;;;;;N;;;;;
```

These are invisible **control** characters, but strangely have the bidi category “Arabic digit” (AN). Note that all other Cf are bidi BN.

```
0660;ARABIC-INDIC DIGIT ZERO;Nd;0;AN;;0;0;0;N;;;;;
0661;ARABIC-INDIC DIGIT ONE;Nd;0;AN;;1;1;1;N;;;;;
0662;ARABIC-INDIC DIGIT TWO;Nd;0;AN;;2;2;2;N;;;;;
0663;ARABIC-INDIC DIGIT THREE;Nd;0;AN;;3;3;3;N;;;;;
0664;ARABIC-INDIC DIGIT FOUR;Nd;0;AN;;4;4;4;N;;;;;
0665;ARABIC-INDIC DIGIT FIVE;Nd;0;AN;;5;5;5;N;;;;;
0666;ARABIC-INDIC DIGIT SIX;Nd;0;AN;;6;6;6;N;;;;;
0667;ARABIC-INDIC DIGIT SEVEN;Nd;0;AN;;7;7;7;N;;;;;
0668;ARABIC-INDIC DIGIT EIGHT;Nd;0;AN;;8;8;8;N;;;;;
0669;ARABIC-INDIC DIGIT NINE;Nd;0;AN;;9;9;9;N;;;;;
```

However, 06F0;EXTENDED ARABIC-INDIC DIGIT ZERO...06F9;EXTENDED ARABIC-INDIC DIGIT NINE are not AN but EN.

It would be easier (conceptually) to consider all decimal digits (that are not bidi "R") as bidi "L" or at least EN (if EN is kept).

Note that /, -, and ARABIC DATE SEPARATOR (the latter now AL) should be ELS, if not by default or higher level protocol, by inserting ELS bidi control character around /, - or ARABIC DATE SEPARATOR characters (in a textual date datum, or preinserted via a date format).

```
066B;ARABIC DECIMAL SEPARATOR;Po;0;AN;;;;;N;;;;;
066C;ARABIC THOUSANDS SEPARATOR;Po;0;AN;;;;;N;;;;;
```

The latter two would be more logical to have as ON (neutral), just like just other decimal and thousands separators (full stop or comma, without spaces around).

```
10E60;RUMI DIGIT ONE;No;0;AN;;;1;1;N;;;;;
...
10E7E;RUMI FRACTION TWO THIRDS;No;0;AN;;;;2/3;N;;;;;
```

As mentioned above, these are better considered as bidi L, just as EN should be L instead.

ET:

```
0023;NUMBER SIGN;Po;0;ET;;;;;N;;;;;
0024;DOLLAR SIGN;Sc;0;ET;;;;;N;;;;;
0025;PERCENT SIGN;Po;0;ET;;;;;N;;;;;
00A2;CENT SIGN;Sc;0;ET;;;;;N;;;;;
00A3;POUND SIGN;Sc;0;ET;;;;;N;;;;;
00A4;CURRENCY SIGN;Sc;0;ET;;;;;N;;;;;
00A5;YEN SIGN;Sc;0;ET;;;;;N;;;;;
00B0;DEGREE SIGN;So;0;ET;;;;;N;;;;;
00B1;PLUS-MINUS SIGN;Sm;0;ET;;;;;N;PLUS-OR-MINUS SIGN;;;;
058F;ARMENIAN DRAM SIGN;Sc;0;ET;;;;;N;;;;;
0609;ARABIC-INDIC PER MILLE SIGN;Po;0;ET;;;;;N;;;;;
060A;ARABIC-INDIC PER TEN THOUSAND SIGN;Po;0;ET;;;;;N;;;;;
066A;ARABIC PERCENT SIGN;Po;0;ET;;;;;N;;;;;
09F2;BENGALI RUPEE MARK;Sc;0;ET;;;;;N;;;;;
09F3;BENGALI RUPEE SIGN;Sc;0;ET;;;;;N;;;;;
09FB;BENGALI GANDA MARK;Sc;0;ET;;;;;N;;;;;
0AF1;GUJARATI RUPEE SIGN;Sc;0;ET;;;;;N;;;;;
0BF9;TAMIL RUPEE SIGN;Sc;0;ET;;;;;N;;;;;
0E3F;THAI CURRENCY SYMBOL BAHT;Sc;0;ET;;;;;N;THAI BAHT SIGN;;;;
17DB;KHMER CURRENCY SYMBOL RIEL;Sc;0;ET;;;;;N;;;;;
2030;PER MILLE SIGN;Po;0;ET;;;;;N;;;;;
2031;PER TEN THOUSAND SIGN;Po;0;ET;;;;;N;;;;;
2032;PRIME;Po;0;ET;;;;;N;;;;;
2033;DOUBLE PRIME;Po;0;ET;<compat> 2032 2032;;;;N;;;;;
2034;TRIPLE PRIME;Po;0;ET;<compat> 2032 2032 2032;;;;N;;;;;
20A0;EURO-CURRENCY SIGN;Sc;0;ET;;;;;N;;;;;
20A1;COLON SIGN;Sc;0;ET;;;;;N;;;;;
20A2;CRUZEIRO SIGN;Sc;0;ET;;;;;N;;;;;
20A3;FRENCH FRANC SIGN;Sc;0;ET;;;;;N;;;;;
20A4;LIRA SIGN;Sc;0;ET;;;;;N;;;;;
20A5;MILL SIGN;Sc;0;ET;;;;;N;;;;;
20A6;NAIRA SIGN;Sc;0;ET;;;;;N;;;;;
```

```
20A7;PESETA SIGN;Sc;0;ET;;;;;N;;;;;
20A8;RUPEE SIGN;Sc;0;ET;<compat> 0052 0073;;;;N;;;;;
20A9;WON SIGN;Sc;0;ET;;;;;N;;;;;
20AA;NEW SHEQEL SIGN;Sc;0;ET;;;;;N;;;;;
20AB;DONG SIGN;Sc;0;ET;;;;;N;;;;;
20AC;EURO SIGN;Sc;0;ET;;;;;N;;;;;
20AD;KIP SIGN;Sc;0;ET;;;;;N;;;;;
20AE;TUGRIK SIGN;Sc;0;ET;;;;;N;;;;;
20AF;DRACHMA SIGN;Sc;0;ET;;;;;N;;;;;
20B0;GERMAN PENNY SIGN;Sc;0;ET;;;;;N;;;;;
20B1;PESO SIGN;Sc;0;ET;;;;;N;;;;;
20B2;GUARANI SIGN;Sc;0;ET;;;;;N;;;;;
20B3;AUSTRAL SIGN;Sc;0;ET;;;;;N;;;;;
20B4;HRYVNIA SIGN;Sc;0;ET;;;;;N;;;;;
20B5;CEDI SIGN;Sc;0;ET;;;;;N;;;;;
20B6;LIVRE TOURNOIS SIGN;Sc;0;ET;;;;;N;;;;;
20B7;SPESMILO SIGN;Sc;0;ET;;;;;N;;;;;
20B8;TENGE SIGN;Sc;0;ET;;;;;N;;;;;
20B9;INDIAN RUPEE SIGN;Sc;0;ET;;;;;N;;;;;
212E;ESTIMATED SYMBOL;So;0;ET;;;;;N;;;;;
2213;MINUS-OR-PLUS SIGN;Sm;0;ET;;;;;N;;;;;
A838;NORTH INDIC RUPEE MARK;Sc;0;ET;;;;;N;;;;;
A839;NORTH INDIC QUANTITY MARK;So;0;ET;;;;;N;;;;;
FE5F;SMALL NUMBER SIGN;Po;0;ET;<small> 0023;;;;N;;;;;
FE69;SMALL DOLLAR SIGN;Sc;0;ET;<small> 0024;;;;N;;;;;
FE6A;SMALL PERCENT SIGN;Po;0;ET;<small> 0025;;;;N;;;;;
FF03;FULLWIDTH NUMBER SIGN;Po;0;ET;<wide> 0023;;;;N;;;;;
FF04;FULLWIDTH DOLLAR SIGN;Sc;0;ET;<wide> 0024;;;;N;;;;;
FF05;FULLWIDTH PERCENT SIGN;Po;0;ET;<wide> 0025;;;;N;;;;;
FFE0;FULLWIDTH CENT SIGN;Sc;0;ET;<wide> 00A2;;;;N;;;;;
FFE1;FULLWIDTH POUND SIGN;Sc;0;ET;<wide> 00A3;;;;N;;;;;
FFE5;FULLWIDTH YEN SIGN;Sc;0;ET;<wide> 00A5;;;;N;;;;;
FFE6;FULLWIDTH WON SIGN;Sc;0;ET;<wide> 20A9;;;;N;;;;;
```

A few of these should be ELS (like PLUS-MINUS SIGN), the rest either ON (like for #) or L (currency signs, except the one for shekel). Most currency signs should be treated the same as currency denotations like USD or rp. Most of them are based on Latin letters, and should be treated for bidi purposes the same as Latin letters. There are a few exceptions, like the one for shekel which should be R. * PERCENT, * PER MILLE, and * PER TEN THOUSAND should be ON (or split into L for the ordinary ones and R for the Arabic ones).

Rationale: The questionable bidi categories seem to be a very failed attempt to automate something (in relation to strings with digits) that should not be automated **in the bidi algorithm**. The effect should instead be achieved by different logical order input, not by the reordering these categories cause in UBA v.1. Note that CLDR offers having locale-dependent formats for dates, times, and currency related expressions with numbers. These are more flexible, and more systematic. The format should result in the same order of fields whether true currency symbols or letters or a mix of true currency symbols and letters, e.g. US$, is used to denote the currency (*provided* the suggestions above are followed, so the bidi algorithm does not mess things up; the bidi algorithm v.1 messes this up). In short, this aspect of UBA v.1 is a(nother) major failure and needs to be cleaned away. (Other failures of UBA v.1 are that it does not even handle “ordinary” natural language text (including simple linear math expressions) well out of the box, and that there has been too much emphasis on using UBA (v.1) unaltered by higher level protocols, so formal syntaxes are mishandled and messed up rather than emphasising

adaptation of the bidi algorithm to handle formal syntaxes in an as readable manner as possible.)

12. Do not add AL MARK (proposed, see PRI 205).

Rationale: AL MARK is a proposed format control characters having the bidi category AL (and that is the point of this control character). The bidi category AL is here (point 11) proposed to be abolished and replaced by R. The AL MARK would then work exactly as RLM.

Summary

This proposal is inspired by the proposal to fix certain aspects of the bidi algoritm, as set forth in PRI 205 and PRI 185. The proposals here pick up on the suggestion to have a UBA v.2 (since the UBA (v.1) is too stabilised to be fixed). The proposals here suggest to fix not only the issues set forth in PRI 205 and 185, but take a more general approach.

The more general approach is needed in order to get results that are more predictable and more readable. Firstly by making normal sentence structure play a more significant role in the v.2 bidi algorithm, so that normal sentence structure, including parentheticals and quotes is respected and not confusingly reordered away and messed up. Secondly by removing arbitrary reorderings near digit sequences, something that is more locale-dependent, or even individual preference, than something bidi script dependent. This (basically arbitrary) reordering does not work well in a more general context where currency symbols and other units need not be expressed using a particular subset of characters, but may use other types of characters (such as letters) as well.

The higher degree of predictability and readability applies also when facing the necessity for letting higher order protocols affect the working of the bidi algorithm (usually by using the new category ELS more liberally than is done by default, and possibly adding more embedding). Of course this is provided that this is done in proper manner, relative to which special syntax is to be catered for, so that the result really is more readable.