L2/20-006

Proposal suggesting formal edits to UAX #14

For consideration by Unicode Technical Committee

2020-01-06
Marcel Schneider (charupdate@orange.fr)

We should always say what we see.
Above all we should always
—which is most difficult—
see what we see.

Charles Péguy

Introduction

Submitted in response to action item 161-A1, this proposal is derived from L2/19-317 ***Proposal to update some statements about space characters in*** **Unicode Standard Annex #14: Unicode Line Breaking Algorithm,** where I indistinctly suggested both material and editorial changes in a single move, not noticing that the latter are usually grouped into an extra section, and downplaying the former in the process. This now focuses on formal edits only. Material changes are suggested alongside but separately in *Proposal to make material changes to UAX #14,* short {Material}**.**

Rather than standalone, the suggestions in this paper are merely complemental. They make sense only if crucial material edits are accepted, as otherwise they would be at risk of contributing to deceive the reader by fueling the belief that UAX #14 is providing reliable information about the topics under discussion.

By coincidence, this proposal is also part of Unicode 13.0 beta feedback.

Scope

Those parts of UAX #14 that are the matter of the previous proposal are considered in the first place. The related edits are numbered 1.1 through 1.7 in section *1 Focused edits*.

L2/19-317 also contains instances where more parenthesized abbreviations are added after listed character names. These are completed and grouped into section *2 Abbreviation support*, as they are based on assumptions made in *Proposal to extend support for abbreviations* (short {Abbreviations}), also scheduled for simultaneous submission. That is due to the complexity of some concerns related to the underlying data.

Trying to do a better job is mandatory, however, both for the sake of internal consistency and to ensure that UAX #14 seamlessly integrates with keyboard layout documentation. Therefore, collected edit suggestions as far as they got noticed all over the document are listed in section *3 Collected edits*, where they are in linear order. Some of them are raising concern in my opinion.

Technical

This proposal refers to the current latest UAX #14 Unicode 12.0.0 (2019-02-15, revision 43).

Highlighting is yellow for new text, lime green for reused, and ~~purple & barred~~ for deleted. That color scheme aims at distinguishing moved, copy-pasted or case-converted strings, from those that are added from scratch. Using another color for deletions (plus line through) is for easier fast-reading.

The “Change to” subsection and “Rationale” as found in L2/19-317 are merged into a single suggestion block preceded by a right-pointing arrow, close to the layout used in Markus Scherer’s L2/19-364.

As a matter of style, these proposals usually don’t place final punctuation before a closing quotation mark unless it is part of the quotation or of the quoted string, because I’m convinced that the advantage of unambiguity and clarity far outweighs the graphical downside.

1. Focused edits

These are located in section **GL: Non-breaking (“Glue”) (XB, XA).** Other edits suggested in L2/19-317 are part of either material changes (see {Material}) or abbreviation support (see next section).

1.1
In particular, when NO-BREAK SPACE follows SPACE, there is a break opportunity after the SPACE and the NO-BREAK SPACE will go as visible space onto the next line.

➔ For readability, a comma would be useful after “break opportunity after the SPACE,”.

1.2

00A0	NO-BREAK SPACE (NBSP)
202F	NARROW NO-BREAK SPACE (NNBSP)
180E	MONGOLIAN VOWEL SEPARATOR (MVS)

➔ For consistency, rows 2 and 3 could be switched so as to sort the list by code points. (Revision 9 already “sorted some lists by code points”.) Additionally, that will help prevent the adverse effect of the misleadingly close names of NBSP and NNBSP. As a bonus, the MVS will be raised:

00A0	NO-BREAK SPACE (NBSP)
180E	MONGOLIAN VOWEL SEPARATOR (MVS)
202F	NARROW NO-BREAK SPACE (NNBSP)

1.3
as in the case of a title and a name “Dr.<NBSP>Joseph Becker”.

➔ For readability, a colon would be useful after “name:”.

1.4
When SPACE follows NO-BREAK SPACE, there is no break

➔ This paragraph is about NO-BREAK SPACE, but the subject of this sentence is SPACE. I’d suggest making NO-BREAK SPACE the subject of this sentence too, like it is the subject of the preceding sentence, and adapting the verb accordingly:
When NO-BREAK SPACE precedes SPACE ~~follows~~, there is no break

1.5
This character has no visible glyph and its presence indicates that adjoining characters are to be treated as a graphemic unit, therefore preventing line breaks between them.

➔ For readability, a comma would be useful after “glyph,”.

1.6
The use of *grapheme joiner* affects other processes, such as sorting, therefore, U+2060 WORD JOINER should be used if the intent is to merely prevent a line break.

➔ For readability, the comma after "sorting" should be replaced with a semicolon.

➔ The conditional clause comes logically first, and it has also precedence over the recommendation as it expresses the user's expectation, that the recommendation is assumed to meet.

Hence I'd suggest rearranging this sentence to:

The use of *grapheme joiner* affects other processes, such as sorting; therefore, if the intent is to merely prevent a line break, U+2060 WORD JOINER should be used instead.

1.7
All of these space characters have a specific width, but otherwise behave as breaking spaces. In setting a justified line, none of these spaces normally changes in width, except for THIN SPACE when used in mathematical notation.

➔ Item 1.4 of {Material} suggests moving the second clause to a new first paragraph. As already noted, this proposal supposes acceptance of {Material}. The third clause ("In setting a justified line") has the same grammatical subject than the fourth clause, so it might be slightly reworded. A revision note for revision 15 already uses the phrase "when lines are justified" rather than "in setting a justified line". ("Added a note on the behavior of U+200B and U+3000 when lines are justified.")

Result after clause removal and rewording:

All of these space characters have a specific width, ~~but otherwise behave as breaking spaces.~~ and ~~none of these spaces~~ they do not normally change~~s~~ in width ~~In setting a justified~~ when lines are justified ~~line~~, ~~none of these spaces normally changes in width,~~ except for THIN SPACE when used in mathematical notation.

2. Abbreviation support

Many of the suggestions in this section are valid only if the simultaneously submitted *Proposal to extend support for abbreviations,* short {Abbreviations}, can be positively reviewed. The reason is that for being able to occur in a UAX, abbreviations must stand in NameAliases.txt.

Actually, a number of abbreviations do exist but are not in NameAliases.txt; others do exist but do not comply to Unicode namespace constraints, or did exist during a time in history, or did never exist but seem desirable for convenience and for equity. {Abbreviations} is trying to assess the pertinence and extension of additional abbreviation support.

Adding the abbreviations after character names in example list tables of UAX #14 may be considered purely editorial, provided that the material part, which is about the underlying data, is appropriately addressed elsewhere, for instance in {Abbreviations}.

2.1 AL: Ordinary Alphabetic and Symbol Characters (XP)

0600..0604	ARABIC NUMBER SIGN..ARABIC SIGN SAMVAT
06DD	ARABIC END OF AYAH
070F	SYRIAC ABBREVIATION MARK
2061..2064	FUNCTION APPLICATION..INVISIBLE PLUS
110BD	KAITHI NUMBER SIGN

➔ Several reasons are encouraging to expand the two collapsed ranges:
- Better documentation;
- Cross-script/community equity;
- Providing more useful abbreviations;
- Unproblematic, web page not length-sensitive.

➔ In UAX #14, among the blank or invisible characters listed as examples without having their parenthesized abbreviation appended to their name, five Arabic format characters are collapsed on a single line: ARABIC NUMBER SIGN, ARABIC SIGN SANAH, ARABIC FOOTNOTE MARKER, ARABIC SIGN SAFHA and ARABIC SIGN SAMVAT. So are four mathematical format characters: FUNCTION APPLICATION, INVISIBLE TIMES, INVISIBLE SEPARATOR and INVISIBLE PLUS. Unequally collapsing character ranges in documentation is problematic. Applied to the general-use whitespace range U+2000..U+200A, that scheme would result in the following table collapsing to:

1680	OGHAM SPACE MARK
2000.. 200A	EN QUAD..HAIR SPACE
205F	MEDIUM MATHEMATICAL SPACE
3000	IDEOGRAPHIC SPACE

(That would be the more consistent with Arabic or mathematical format characters as no abbreviations are provided neither.) Obviously, ranges should not be collapsed in documentation, except for losslessly collapsable ranges like dingbat digits, numbered emoji modifiers, and the regional indicator alphabet.

➔ SYRIAC ABBREVIATION MARK has a widely used abbreviation present in NameAliases.txt, in the Core Specification, in the Code Charts as an informative alias, and inside the dashed-box glyph. Although, unlike “CGJ”, “SAM” does not occur elsewhere in UAX #14, it should be provided here. The only point in leaving it out would be to prevent SAM from standing out as if Syriac was favored over the other locales. That same speculation about UAX #14 could at present be made as of Mongolian. I’d suggest making an end of diplomatic balancing, by uniformly providing abbreviations for every character whose name users may wish to abbreviate. I think that characters named basically after their local name, like SANAH, SAFHA and SAMVAT in this table, are not abbreviation-prone.

➔ As explained in {Abbreviations}, where they are moot, the suggested abbreviations of the invisible mathematical format characters are initialisms of names partly redesigned for that purpose in order to get unequivocal strings. More precisely, “TIMES” has been replaced with “MULTIPLICATION SIGN” so as to avoid the initialism “IT”, and “PLUS” has been expanded to “PLUS SIGN” in order to get around the initialism “IP” in this context. However, these probably need to run through another process prior to being added here. Provisionally this item provides the expected maximum until an eventual downcut:

0600~~..0604~~	ARABIC NUMBER SIGN (ANS)~~..ARABIC SIGN SAMVAT~~
0601	ARABIC SIGN SANAH
0602	ARABIC FOOTNOTE MARKER (AFM)
0603	ARABIC SIGN SAFHA
0604	ARABIC SIGN SAMVAT
06DD	ARABIC END OF AYAH (AEOA)
070F	SYRIAC ABBREVIATION MARK (SAM)

2061~~..2064~~	FUNCTION APPLICATION (FA)~~..INVISIBLE PLUS~~
2062	INVISIBLE TIMES (IMS)
2063	INVISIBLE SEPARATOR (IS)
2064	INVISIBLE PLUS (IPS)
110BD	KAITHI NUMBER SIGN (KNS)

2.2 BA: Break After (A)

Breaking spaces

1680	OGHAM SPACE MARK
[…]	[…]
3000	IDEOGRAPHIC SPACE

➔ All of these spaces somewhere either have a conformant abbreviation (9) or a non-conformant one that can be easily redesigned (3), or a historic one that can be revived. OGHAM SPACE MARK standing out by not having its abbreviation in a table featuring abbreviations for all other characters is problematic, as is hiding all abbreviations for that reason. Rather than shifting the problem, I'd suggest solving it instead.

1680	OGHAM SPACE MARK (OGSP)
2000	EN QUAD (NQSP)
2001	EM QUAD (MQSP)
2002	EN SPACE (ENSP)
2003	EM SPACE (EMSP)
2004	THREE-PER-EM SPACE (THPMSP)
2005	FOUR-PER-EM SPACE (FPMSP)
2006	SIX-PER-EM SPACE (SPMSP)
2008	PUNCTUATION SPACE (PSP)
2009	THIN SPACE (THSP)
200A	HAIR SPACE (HSP)
205F	MEDIUM MATHEMATICAL SPACE (MMSP)
3000	IDEOGRAPHIC SPACE (IDSP)

2.3 BA: Break After (A)

Tabs

0009	TAB

➔ This table falls out of current practice as exemplified with U+000B:

000B	LINE TABULATION (VT)

While I don't suggest using "(HT)" for U+0009, providing the ISO 6429 name seems better than replacing it with even the most current of the two existing abbreviations. Please see also item 3.45 below.

0009	CHARACTER TABULATION (TAB)

2.4 BA: Break After (A)

Breaking Hyphens

058A	ARMENIAN HYPHEN
2010	HYPHEN
2012	FIGURE DASH
2013	EN DASH

➔ HYPHEN is particular in that it is the breaking counterpart of NON-BREAKING HYPHEN that is going to have its abbreviation standardized if category 1 in {Abbreviations} is accepted, but also in that in almost all fonts it is extremely confusable with HYPHEN-MINUS, and is useful only in those fonts where the latter's glyph has an uncommon design to account for related information in the Standard, that helped motivate very few designers to give U+002D and U+2010 very dissimilar glyphs, at odds with current end-user preferences.

➔ Obviously other dashes and hyphens could likewise use some abbreviation, but I refrained from expanding this proposal to that extent:

058A	ARMENIAN HYPHEN
2010	HYPHEN (HY)
2012	FIGURE DASH
2013	EN DASH

2.5 BK: Mandatory Break (A) (Non-tailorable)

2028	LINE SEPARATOR
2029	PARAGRAPH SEPARATOR

➔ These are abbreviated "LS" and "PS", respectively. Please se*e Proposal to synchronize seven glyphs in the Code Charts,* and section *A.7 Line and paragraph separators* in {Abbreviations}.

2028	LINE SEPARATOR (LS)
2029	PARAGRAPH SEPARATOR (PS)

2.6 GL: Non-breaking ("Glue") (XB/XA) (Non-tailorable)

034F	COMBINING GRAPHEME JOINER
2007	FIGURE SPACE
2011	NON-BREAKING HYPHEN

➔ The abbreviations of these three characters have dissimilar status, as CGJ is already used elsewhere in the document and should be provided here, while FSP is straightforward as it is inside the dashed-box glyph, in contrast to NBHY that as such stands nowhere in the actual Standard even in the Code Charts, but had been in UAX #14 fairly long, then was removed (instead of being standardized). Now it is advocated in {Abbreviations}, see subsection *2.3 Inclusion of hyphens.*

➔ The case of U+034F COMBINING GRAPHEME JOINER tends to prove that in at least one instance, the lack of a parenthesized abbreviation in UAX #14 has no obvious reason. NameAliases.txt provides "CGJ" as this character's abbreviation. The Core Specification (TUS) makes massive use of "CGJ" (24 instances). The Code Chart of the block *Combining Diacritical Marks* shows "CGJ" both inside the dashed box and in an annotation reading "commonly abbreviated as CGJ". And UAX #14 itself is using the initialism to refer to the

character in Table 1. As a consequence, UAX #14 should provide the abbreviation along with the name when introducing the character, like it does for a number of other characters, namely LF, VT, FF, CR, SP, NEL, NBSP, SHY, MVS, ZWSP, ZWJ, NNBSP, WJ and ZWNBSP.

➔ Provided that "FSP" and "NBHY" will make it into NameAliases.txt, they are suggested for addition along with the already-standard "CGJ":

034F	COMBINING GRAPHEME JOINER (CGJ)
2007	FIGURE SPACE (FSP)
2011	NON-BREAKING HYPHEN (NBHY)

2.7 5.3 Use of Hyphen

➔ In the wake, the two instances of the string descriptor "<SHY, NON-BREAKING HYPHEN>" in subsections 5.3 and 5.4 can be restored to their original form "<SHY, ~~NON-BREAKING HYPHEN~~ NBHY>".

2.8 5.4 Use of Soft Hyphen

➔ The string descriptor "tugg<U+2010>/ gummi" may be changed to "tugg<~~U+2010~~HY> / gummi", or rather to "~~tugg~~bränn<~~ U+2010~~HY> / ~~gummi~~nässla" if suggestions 3.74 and 3.75 are accepted.

3. Collected edits

This list does not claim to be comprehensive (even if it is so in my understanding). Some of the suggestions needed to be moved to {Material}. But even here, a number of items are raising much concern, in particular (the long) item 3.2.

3.1 Summary

This annex presents the Unicode line breaking algorithm along with detailed descriptions of each of the character classes established by the Unicode line breaking property.

➔ "*Unicode ~~l~~Line ~~b~~Breaking ~~a~~Algorithm*" is titlecased in all other instances, and it probably should be so here, too.

3.2 1 Overview and Scope

For most Unicode characters, considerable variation in line breaking behavior can be expected, including variation based on local or stylistic preferences. For that reason, the line breaking properties provided for these characters are informative.

1 ➔ As per LineBreak.txt, the section "Data file" in UAX #14, and Table 3.2 in the Core Specification, the "Line_Break" property is a normative property, and therefore all values, also known as classes, are normative. None are "informative" as claimed in the quoted snippet. UAX #14 itself states under the heading "Data file" in section *5 Line Breaking Properties*: "The line break property assignments from the data file are normative."

That fact is unrelated to whether a given class is tailorable or not. In fact, most line breaking classes are, but that does not preclude these classes from being normative.

Consistently, in section 5,1, the headings of the eleven non-tailorable line breaking classes comprise the string "(non-tailorable)", not the string "(normative)".

2 ➔ The one “Line Break” property in the Unicode Standard being of type “enumerated” does not make it multiple properties. Actually, the text quoted above is still legacy wording inherited from early versions and never extensively updated, despite rising awareness that those early drafts fell short of providing clear and reliable information. From earliest *revision 0.3* (1998) through *version Unicode 5.0.1 Draft* (2007), the document was even called “Line Breaking Properties”, and the fifth section, initially the “Specification”, got soon today’s heading “Line Breaking Properties”.

Correcting terminological and semantic mistakes proves to be highly incremental, at least in UAX #14, and to span over relatively long periods of time, as with these three examples:

Version of UAX #14	**Section 5 Heading**	**Sample phrases in section 5**	
Revision 4 (as presented at IUC 13 [B14]) 1998-07-30	5.0 Specification	[…] list Unicode characters grouped by their line breaking property […]	
Revision 5 (PDF file) 1999-08-20	5.0 Specification	[…] to fix the membership of character classes for each line breaking property.	The classification by properties defined here is used as input […]
Revision 6.0 1999-11-15	5~~.0 Specification~~ Line Breaking Properties	[…] to ~~fix~~ summarize the membership of character classes for each line breaking property.	The classification by properties defined here is used as input […]
Version 4.0.0 2003-04-17 Revision 14	5 Line Breaking Properties	[…] to summarize the membership of character classes for each value of the line breaking property.	The classification by properties defined here is used as input […]
Version 4.1.0 2005-08-29 Revision 17	5 Line Breaking Properties	[…] to summarize the membership of character classes for each value of the line breaking property.	The classification by propert~~ies~~ values defined ~~here~~ in this section and in the data file is used as input […]
Version Unicode 12.0.0 2019-02-15 Revision 43	5 Line Breaking Properties	[…] ~~to~~ summarizes the membership of character classes for each value of the line breaking property.	The classification by property values defined in this section and in the data file is used as input […]
Upcoming soon? [see 3.27]	*5 Line Breaking Propert~~ies~~ Values*	*[…] summarizes the membership of character classes for each value of the line breaking property.*	*The classification by property values defined in this section and in the data file is used as input […]*

3 ➔ Section 2 *Definitions* of *UAX #14* states (bold added): “Table 1 lists all of line breaking **classes** by name, also giving their **class** abbreviation and their **status as tailorable or not.** The examples and brief indication of line breaking behavior in this table are merely typical, not exhaustive. Section 5.1, *Description of Line Breaking **Properties***, provides a detailed description of each line breaking **class**, including detailed overview of the line breaking behavior for characters of that **class.**”

As a result, we can make an attempt to decrypt the confusing language of UAX #14: "Line breaking **properties**" are actually line breaking **classes;** "**normative** or **informative** line breaking properties" are actually **non-tailorable** or **tailorable** line breaking classes.

5 ➔ To keep arguing that a line breaking class is informative if it is tailorable, and normative if it is not, the file LineBreak.txt would need to undergo a number of material changes:

Line	Original LineBreak.txt v12.1.0	Synched with UAX #14 version Unicode 12.1.0
12	This file is a normative contributory data file in the Unicode Character Database.	This file is partly a normative and partly an informative contributory data file in the Unicode Character Database.
18	Non-tailorable:	~~Non-tailorable:~~ Normative:
20	Tailorable:	~~Tailorable:~~ Informative:

These useless and highly confusing changes are not going to happen. Correcting UAX #14 so as to keep that one in synch with UCD is far more sustainable than the other way around, since in UAX #14 the needed changes are most probably considered merely formal edits correcting quiproquos without affecting the underlying content.

6 ➔ Nowhere in UAX #14 is "**line breaking property**" (singular) defined otherwise than in LD5. And in 4 out of 6 instances elsewhere it has actually the meaning as defined. Only under SY (see 3.61) and XX (see 3.63) is it used instead of "**line breaking class**". Where the term is pluralized to "**line breaking properties**", 17 out of 18 instances of that string can be safely replaced with "line breaking classes". The one exception is discussed in suggestion 3.103 below.

7 ➔ For all those reasons, I suggest replacing the terms "~~properties~~" with "classes", "~~provided for~~" with "assigned to", and "~~informative~~" with "tailorable":

For most Unicode characters, considerable variation in line breaking behavior can be expected, including variation based on local or stylistic preferences. For that reason, the line breaking ~~properties~~ classes ~~provided for~~ assigned to these characters are ~~informative~~ tailorable.

3.3 1 Overview and Scope

Some characters are intended to explicitly influence line breaking. Their line breaking behavior is therefore expected to be identical across all implementations. As described in this annex, the Unicode Standard assigns normative line breaking properties to those characters.

➔ If the previous suggestion is applied, "~~normative~~ line breaking ~~properties~~" should be replaced with "non-tailorable line breaking classes".

3.4 1 Overview and Scope

The Unicode Line Breaking Algorithm is a tailorable set of rules that uses these line breaking properties in context to determine line break opportunities.

➔ Although many of these rules are tailorable, the set as a whole is still normative, since UAX #14 is normative.

➔ Focusing on the rules: Since some are tailorable and some are not, pointing that difference may seem more meaningful. That will allow keeping the term "tailorable", just moving it to the right place.

➔ The "properties" here are actually classes again.

➔ These classes are used by the rules, rather than by the set, so the verb should be plural.

➔ As a result, I suggest changing this to:

The Unicode Line Breaking Algorithm is a ~~tailorable~~ normative set of tailorable rules and non-tailorable rules that use~~s~~ these line breaking ~~properties~~ classes in context to determine line break opportunities.

3.5 2 Definitions

LD1*. *Line Fitting: The process of […].

➔ I think that the defined terms should not be titlecased as headings. The heading is actually the identifier, here LD1, and consistently the identifier bears the hyperlink with the anchor for easy reference. The defined term is already italicized and boldened to stand out, and is followed by a colon. Titlecase in addition is confusing. A heading is sort of a name (of a chapter, a section), whereas the defined term is the object of the definition, not its name (which is, again, the identifier, here "LD1").
Therefore I suggest leaving only the first word titlecased, and lowercasing all other words of the defined terms in this section.

3.6 2 Definitions

LD2*. *Line Break: The position in the text where one line ends and the next one starts.

➔ I don't think that the definite article is appropriate when generically referring to a "position in the text". This mistake has probably happened under the influence of LD1 and LD4 defining processes. In a given implementation there is only one of each process, while there are multiple line breaks in a document. Only in "the text" is the definite article okay because this specification does not focus on multiple documents. But it does on multiple line breaks.

➔ As it appears in the next suggestion (3.7), as a matter of good style this document is very careful about not repeating words too often. Yet although I'm aware that in various languages it is a part of racy style, I don't think that the phrase "where one line ends and the next one starts" is particularly graceful, as in neither instance the word "one" is objectively straightforward. (Additionally, Word highlights "one line" as a grammar flaw.)

➔ The word "break" in the defined term is lowercased according to 3.5:

LD2*. *Line ~~B~~b break: ~~The~~ A position in the text where ~~one~~ a line ends and the next ~~one~~ line starts.

3.7 2 Definitions

LD3*. *Line Break Opportunity: A place where a line is allowed to end.

➔ I don't think that varying terms in a technical standard for solely stylistic purposes is a good idea. In the definitions section it is even less. The object is correctly called "a position in the text" both in the comment and in the preceding definition LD2. So it should be here.

➔ The words "break" and "opportunity" in the defined term are lowercased according to 3.5:

LD3*. *Line ~~B~~b break ~~O~~o pportunity: A ~~place~~ position in the text where a line is allowed to end.

3.8 2 Definitions

LD4*. *Line Breaking: The process of selecting one among several line break opportunities such that the resulting line is optimal or ends at a user-requested explicit line break.

➔ If there is a hard line break in a line, the line breaking process does not run for that line, since there is no point in selecting anything else for that line to end.

➔ The word "breaking" in the defined term is lowercased according to 3.5:

LD4*. *Line ~~B~~b reaking: The process of selecting one among several line break opportunities such that the resulting line is optimal, if it does not ~~or~~ end~~s~~ at a user-requested explicit line break.

3.9 2 Definitions

***LD5*. Line Breaking Property:** A character property with enumerated values, as listed in *Table 1*, and separated into normative and informative values.

➔ Accordingly, here too, "~~*normative*~~ *and* ~~*informative*~~" should be replaced with "*non-tailorable and tailorable*", or now for fluidity rather the other way around (lowercase according to 3.5):

***LD5*. Line ~~B~~breaking ~~P~~property:** A character property with enumerated values, as listed in *Table 1*, and separated into ~~normative~~ tailorable and ~~informative~~ non-tailorable values.

3.10 2 Definitions

***LD7*. Mandatory Break:** A line must break following a character that has the mandatory break property.

➔ This is the first instance improperly using the singular of the word "property" instead of "class" or "property value".

➔ Please see also the proposed material changes in {Material}, item 2.2.

➔ The word "break" in the defined term is lowercased according to item 3.5:

***LD7*. Mandatory ~~B~~break:** A line must break following a character that has the mandatory break property value.

3.11 2 Definitions

***LD7*.** […] where **B** is the character with the mandatory break property.

➔ The misuse of "property" may seem handy, and this is probably the ultimate reason why it got established in this UAX. But there is no point in defining terminology for a specification if the specification is not going to use that terminology as defined. Correct wording takes very little effort:

[…] where **B** is the character ~~with~~ of the mandatory break ~~property~~ class.

3.12 2 Definitions

LD7. […] Such a break is also known as a *forced* break and is indicated in the rules as **B !**, where **B** is the character with the mandatory break property.

➔ When regular expressions with spaces like in "B !" are used inline, the spaces should be NO-BREAK SPACE, not SPACE, since the line breaking class EX (that would make the space here non-breaking) is currently not implemented. Therefore, this needs to be turned into (HTML):

```
[…] indicated in the rules as <b>B– !</b>, where […]
```

3.13 2 Definitions

***LD9*. Indirect Break:** A line break opportunity exists between two characters of the given line breaking classes *only* if they are separated by one or more spaces.

➔ This definition is unambiguous only so far as an indirect break is related to the character U+0020 SPACE. Without that meta-analysis, the bare semantics is ambiguous since "spaces" may refer to any breaking spaces as in line breaking rule LB12a, where "spaces and hyphens" refers to the class [SP BA HY] (HY to catch the HYPHEN-MINUS), or it may refer exclusively to U+0020 SPACE as in LB14, where "spaces" refers to the class [SP].

➔ As it stands, this definition is at the intersection of the current definition of "spaces" as "characters of gc=Zs", and the meta-analysis ruling out anything else than U+0020 SPACE. As a result, it is wrong. It can be fixed by uppercasing "SPACE" so as to disambiguate "SPACE" as a character name from the generic meaning of "space".

➔ Please see also the proposed material change in {Material}, item 2.4.

➔ The word “break” and all other words in the defined term are lowercased according to 3.5:

LD9. Indirect ~~B~~break: A line break opportunity exists between two characters of the given line breaking classes *only* if they are separated by one or more ~~space~~SPACEs.

3.14 2 Definitions

LD10. Prohibited Break: No line break opportunity exists between two characters of the given line breaking classes, even if they are separated by one or more space characters.

➔ By using the full-length concept “space characters” this rule insinuates that it refers to all characters of gc=Zs. But probably “SPACE” should be uppercased according to 3.13.

➔ The word “break” and all other words in the defined term are lowercased according to 3.5:

LD10. Prohibited ~~B~~break: No line break opportunity exists between two characters of the given line breaking classes, even if they are separated by one or more ~~space~~SPACE characters.

3.15 2 Definitions

Table 1 lists all of line breaking classes by name, also giving their class abbreviation and their status as tailorable or not.

➔ The definite article “the” seems to be missing between “all of” and “line breaking classes”.

3.16 Table 1

BK	*Mandatory Break*	NL, PARAGRAPH SEPARATOR	Cause a line break (after)
CR	*Carriage Return*	CR	Cause a line break (after), except between CR and LF
LF	*Line Feed*	LF	Cause a line break (after)
[…]			
NL	*Next Line*	NEL	Cause a line break (after)

➔ The parentheses around “after” can be removed. In a specification, even obvious things are expected to be spoken out. Hence the point in parenthesizing this preposition in these rows is irrelevant.

3.17 Table 1

CM	*Combining Mark*	Combining marks, control codes	Prohibit a line break between the character and the preceding character

➔ I’d suggest simplifying the content of the right cell by replacing “~~between the character and the preceding character~~” with “before”, unless there is a real risk of confusing “before” with “before the base character” for combining marks.

3.18 Table 1

WJ	*Word Joiner*	WJ	Prohibit line breaks before and after

➔ WORD JOINER, encoded as GL for Unicode 3.2.0 (2002), was assigned this new class one version later for Unicode 4.0.0 (2003), and its legacy counterpart, previously GL too, immediately followed up.

While in the WJ class description both characters are listed, the change was only partially reflected in Table 1. To fix it, add "**, ZWNBSP**" in the third column. (Related: 3.20)

3.19 Table 1

ZW	*Zero Width Space*	ZWSP	Provide a break opportunity

➔ Append "**after**" to "break opportunity", like what is done elsewhere.

3.20 Table 1

GL	*Non-breaking ("Glue")*	CGJ, NBSP, ZWNBSP	Prohibit line breaks before and after

➔ Consistently with 3.18, remove "~~, ZWNBSP~~" from the third column.

➔ Fix error in titlecase of the descriptive name, as in hyphenated words every part obeys to the rule applicable to standalone words.

3.21 Table 1

B2	*Break Opportunity Before and After*	Em dash	Provide a line break opportunity before and after the character

➔ Since there is one opportunity before, and another one after, "a line break opportunity" should be pluralized.

➔ For consistency with the preceding and following table sections and for simplicity, I'd suggest removing "the character" throughout in this table section:

Provide ~~a~~ line break opportunities~~y~~ before and after ~~the character~~

3.22 Table 1

CJ	*Conditional Japanese Starter*	Small kana	Treat as NS or ID for strict or normal breaking.

➔ This is the only instance in this table where the subject is the implementation, not the characters. I'd suggest synching this by replacing "~~Treat as~~" with "**Act like**" as in 4 other instances (2 in this table).

3.23 4 Conformance

The methods by which a line layout process chooses optimal line breaks from among the available break opportunities is outside the scope of this specification. The behavior of a line layout process in situations where there are no suitable break opportunities is also outside of the scope of this specification.

➔ Making two sentences and varying between "outside" and "outside of" seems to me not streamlined enough, and less suitable for a Unicode Standard Annex, than contracting into one sentence and using only the form "outside ***of***", preferred in formal writing:

The methods by which a line layout process chooses optimal line breaks from among the available break opportunities, and ~~is outside the scope of this specification. T~~the behavior of a line layout process in situations where there are no suitable break opportunities, are ~~is also~~ outside of the scope of this specification.

3.24 4.1 Conformance Requirements

Note: Locale-sensitive line break specifications can be expressed in LDML [UTS35]. Tailorings are available in the Common Locale Data Repository [CLDR].

➔ This is a note (incidentally the first one), but unlike all other notes it is not indented. So it should be.

➔ To indent notes, <blockquote> tags are used. The stylesheet reduces their default margin but does not suggest that they should be used for anything else than quotations. There is even a class "tus" for quotations from *TUS*. That is consistent with semantic web requirements where <blockquote> denotes a quotation and must not be used merely for its default indent, per W3C specification. I'd suggest using a and give it the same margin. The class "note" is already used with for italicizing the note headings, so I'd suggest adding in reports-v2.css(127):

```
div.note { margin: 20px; }
```

and in the process, replacing "HEADERS" with "HEADINGS" at line 102, and all three instances of "headers" with "headings" (113, 124, 229).

Then adding that around this note, and substituting it to <blockquote> around all other 20 notes and the one example at line 821. Since there are also quotations and (quoted) examples where the <blockquote> shall remain, replacing ~~<blockquote>~~ with `<div class="note">` will occur only at these lines, and ~~</blockquote>~~ with `</div>` a few lines below each instance:

821, 932, 999, 1026, 1674, 1983, 2036, 2275, 2321, 2486, 2622, 2675, 2799, 2812, 2894, 2957, 2980, 3006, 3330, 3818, and 4076.

3.25 4.1 Conformance Requirements

3. *If the implementation tailors the behavior of Section 6.2, Tailorable Line Breaking Rules, that fact must be disclosed.*

➔ I don't think that a section can stand for the rules contained in that section to such an extent that an implementation may be told tailoring the behavior of the section rather than the behavior of the rules. Hence, I'd change this to:

If the implementation tailors ~~the behavior of~~ rules from Section 6.2 […]

3.26 4.1 Conformance Requirements

As is the case for all other Unicode algorithms, this specification is a logical description—particular implementations can have more efficient mechanisms as long as they produce the same results.

➔ Since this em dash does not open a parenthetical, and for the break in the reading flow a period would do it, this dash has more of a special touch. The intent is probably to ease the prosaic standards style. But the clauses on either side seem a bit too long for this punctuation to work smoothly. With a period, this would read:

[…] this specification is a logical description. ~~—p~~Particular implementations can have […]

3.27 5 Line Breaking Properties

➔ This heading should be corrected as hinted in the last table row under point 2 of item 3.2:

5 Line Breaking Propert~~ies~~y Values

3.28 5 Line Breaking Properties

Note that the mnemonic names for the line break classes are intended […]

➔ This is the first of 47 instances of “line break class(es)” instead of “line breaking class(es)” as it stands in 40 instances. Although “line break class” relies on the “`Line_Break Property`” and the name of the containing UCD file “LineBreak.txt”, it contradicts definition LD2 defining “line break” as an actual instance, and definition LD4 defining “line breaking” as a process. In my understanding, a fictional “line break class” would be a class of line breaks, such as the class of direct breaks, or the class of mandatory breaks, but these are no character property values. Therefore, referring to a “line break class” when talking about characters is semantically incorrect. For consistency I’d suggest correcting all of the wrong instances at once. Please compare to item 3.50 about not skipping the -ing in “line breaking behavior”.

3.29 5 Line Breaking Properties

[…] neither as exhaustive descriptions of their membership nor as indicators of their entire range of behaviors in the line breaking process.

➔ For readability, a comma would be useful after “membership,”.

3.30 *Data File*

➔ This is the first level-4 heading, and it is closer to the preceding than to the following paragraph, which for a heading looks unusual. The bug results from a rule in the stylesheet specifying an 8px top margin for all headings. Deleting that rule puts the h4s at equal distance from above and below.

➔ However, instead of deleting the rule, setting its value to *38 pixels* greatly improves the readability of the whole UAX, making it look less compressed. Checking other UAXes results in the same conclusion.

Suggested edit: In <https://www.unicode.org/reports/reports-v2.css>(103), prepend “3” to “8px”.

3.31 *Data File*

This is a semicolon-delimited, two-column, plain text file, with code position and line breaking class.

➔ That does not account for collapsed ranges. As-is, it was accurate until Unicode 6.0.0. Appending a parenthesized plural -s to “position(s)” could fix that.

3.32 *Data File*

A comment at the end of each line indicates the character name.

➔ This seems equally outdated with respect to the latest data file containing more information per line. Since Unicode version 7.0.0, the list is shortened by collapsing ranges. To account for that:

A comment at the end of each line indicates the General_Category, for a range the number of code positions, and the (first and last) character name.

3.33 *Data File*

The descriptions of the line break classes in this UAX include examples of representative or interesting characters for each class, but for the complete list always refer to the data file.

➔ The last clause suggests that the lists are found in the file, that would then be ordered by line breaking classes like DerivedAge.txt is ordered by Unicode versions. However, to get any such list, LineBreak.txt needs to be parsed first. That fact should be reflected somehow (“line breaking classes” according to item 3.28):

“The descriptions of the line breaking classes in this UAX include examples of representative or interesting characters for each class, but ~~for~~ the complete list is always ~~refer to~~ obtained by parsing the data file.”

3.34 Future Updates

As scripts are added to the Unicode Standard and become more widely implemented, line breaking classes may be added or the assignment of line breaking class may be changed for some characters.

➔ I'd suggest streamlining the last phrase and make it flow smoother by moving "assignment":
[…] or the ~~assignment of~~ line breaking class assignment may be changed […]

3.35 5.1 Description of Line Breaking Properties

Each line breaking class is marked with an annotation in parentheses with the following meanings:

➔ I see two ways of rewording this logically: Either "[…] an annotation in parentheses with one of the following meanings", or end the sentence after "parentheses." and start a new one: "These annotations have ~~with~~ the following meanings:" (I'd prefer the first option.)

3.36 AL: Ordinary Alphabetic and Symbol Characters (XP)

Ordinary characters require other characters to provide break opportunities; otherwise, no line breaks are allowed between pairs of them.

➔ The word "~~otherwise,~~" is too much here, since when other characters occur between two characters of class AL, the latter do not form a pair any longer.

3.37 AL: Ordinary Alphabetic and Symbol Characters (XP)

In some Far Eastern documents, […].

➔ I don't think that "~~Far Eastern~~" is appropriate in the Unicode Standard, rather than "East Asian". Also, in this UAX, the former occurs only twice (the other instance is pointed in item 3.105), compared to 13 instances of the latter. Both are a legacy dating 20 years back to revision 6. TUS 12.0.0 never uses "Far Eastern". I'd suggest using the more international term throughout.

3.38 AL: Ordinary Alphabetic and Symbol Characters (XP)

Note: Use ZWSP as a manual override to provide break opportunities around alphabetic or symbol characters.

➔ The preposition "around" is misdefining the actual use case. Also the code point is missing compared to the other instance of a note giving guidance for a manual override. Suggested:
Note: Use U+200B ZWSP as a manual override to provide a break opportunity~~ies around alphabetic or symbol~~ between a pair of characters of line breaking class AL.

3.39 AL: Ordinary Alphabetic and Symbol Characters (XP)

These format characters […] breaks after.

➔ This paragraph is indented using `<blockquote>` as if it were a quotation, yet it does not seem to need styling setting it off from the rest of the body text. I'd suggest deleting this pair of `<blockquote></blockquote>` tags. (Cf. 3.24 about changing markup for indentation.)

3.40 AL: Ordinary Alphabetic and Symbol Characters (XP)

Major exceptions to the general pattern of alphabetic and symbolic characters having line break class **AL** include**:**

➔ The following list enumerates ten other line breaking classes. These are not actually “Major ~~exceptions~~”, but “Major alternatives”.

➔ Correcting “line break class” to “line breaking class” almost throughout is suggested in 3.28.

3.41 AL: Ordinary Alphabetic and Symbol Characters (XP)
HL for Hebrew letters

➔ This is probably the first instance where a character class is cited without the hyperlink to its description. 371 class abbreviation occurrences have a hyperlink, the rest don’t. Some of the regexes have classes with hyperlinks, some don’t. I’d suggest providing hyperlinks consistently throughout UAX #14, every time a class is cited by its abbreviation, particularly in isolation like here.

3.42 AL: Ordinary Alphabetic and Symbol Characters (XP)
HL for Hebrew letters
AI or ID, based on the East Asian Width property of the character
ID for certain pictographic symbols
CJ for small hiragana and katakana
SA for complex context scripts
JL, JV, JT, H2 or H3 for Hangul characters

➔ Actually this list is broken into lines using
 tags, and indented (mis)using <blockquote>:

```
    <blockquote>
      <p>HL for Hebrew letters<br>
AI or ID, based on the East Asian Width property of the character<br>
ID for certain pictographic symbols<br>
CJ for small hiragana and katakana<br>SA for complex context scripts<br>JL, JV,
JT, H2 or H3 for Hangul characters</p>
    </blockquote>
```

I suggest applying unordered list markup instead, as that would be both more respectful towards the content, and more conformant to W3C recommendations:

```
  <ul>
    <li>HL for Hebrew letters</li>
    <li>AI or ID, based on the East Asian Width property of the character</li>
    <li>ID for certain pictographic symbols</li>
    <li>CJ for small hiragana and katakana</li>
    <li>SA for complex context scripts</li>
    <li>JL, JV, JT, H2 or H3 for Hangul characters</li>
  </ul>
```

3.43 BA: Break After (A)
Breaking Spaces
Breaking spaces are a subset of characters with General_Category Zs. Examples include:

➔ Since the following list, quoted at suggestion 2.2, is the full list, I’d change this to:
[…] ~~Examples include~~ The full list is:

3.44 BA: Break After (A)

Breaking Spaces

The OGHAM SPACE MARK may be rendered visibly between words but it is recommended that it be elided at the end of a line.

➔ For readability, a comma would be useful after “words,”.

3.45 BA: Break After (A)

Tabs

➔ Beside redesigning the example as described above (2.3), I’d suggest singularizing the heading, since it refers to characters, like the preceding one (“Breaking spaces are a subset of characters”), not to actual occurrences. There is only one tab character referred to here, so it should be singular.

➔ Further, in a heading like this one, a plain term is probably preferable over a colloquial abbreviation. Except that the new ISO 6429 name seems to be based on a misconception of what a “character” is supposed to represent. So rather than CHARACTER TABULATION, the correct long name is HORIZONTAL TABULATION like in Unicode 1.0, as opposed to the VERTICAL TABULATION.

I’d suggest changing the heading to:

Horizontal Tabulations

3.46 BA: Break After (A)

Conditional hyphens

➔ If SHY is the only character involved, this heading should be singular, too: “***Conditional hyphens***”

3.47 BB: Break Before (B)

Dictionary Use

The accent should not be separated from the syllable it marks by a break.

➔ This confusing sentence can easily be fixed by moving “by a break” after “separated”:

The accent should not be separated by a break from the syllable it marks ~~by a break~~.

3.48 BB: Break Before (B)

Dictionary Use

Note: It is hard to find actual examples in most dictionaries because the pronunciation fields usually occur right after the headword, and the columns are wide enough to prevent line breaks in most pronunciations.

➔ For better readability, a comma should be added amidst “dictionaries, because”.

3.49 B2: Break Opportunity Before and After (B/A/XP)

The EM DASH is used to set off parenthetical text. Normally, it is used without spaces. However, this is language dependent. For example, in Swedish, spaces are used around the EM DASH. Line breaks can occur before and after an EM DASH. Because EM DASHes are sometimes used in pairs instead of a single quotation dash, the default behavior is not to break the line between even though not all fonts use connecting glyphs for the EM DASH.

Some languages, including Spanish, use EM DASH to set off a parenthetical, and the surrounding dashes should not be broken from the contained text. In this usage there is space on the side where it can be broken. This does not conflict with symmetrical usages, either with spaces on both sides of the em-dash or with no spaces.

➔ This content appears disordered, as the first paragraph is talking about two topics in a row, and the second, about the first topic again. To fix it, the last sentence of the first paragraph should be made a third paragraph:

The EM DASH is used to set off parenthetical text. Normally, it is used without spaces. However, this is language dependent. For example, in Swedish, spaces are used around the EM DASH. Line breaks can occur before and after an EM DASH. ~~Because EM DASHes are sometimes used in pairs instead of a single quotation dash, the default behavior is not to break the line between even though not all fonts use connecting glyphs for the EM DASH.~~

Some languages, including Spanish, use EM DASH to set off a parenthetical, and the surrounding dashes should not be broken from the contained text. In this usage there is space on the side where it can be broken. This does not conflict with symmetrical usages, either with spaces on both sides of the em-dash or with no spaces. [PARAGRAPH BREAK]

Because EM DASHes are sometimes used in pairs instead of a single quotation dash, the default behavior is not to break the line between even though not all fonts use connecting glyphs for the EM DASH.

➔ Please see *{Material}*, item 2.9, for material changes suggested for the first clause of the moved sentence.

➔ Pointing the font issue after "used in pairs instead of a single quotation dash" rather than mentioning it at the end after "not to break the line between" would be more straightforward.

➔ In "the default behavior is not to break the line between", the "not" seems displaced, altering the meaning of the phrase.

➔ A comma is actually missing after "between,". But that would become irrelevant if the final period ends up at that position when the rest of the sentence is moved.

➔ As a result of the additional notes, I'd suggest rewording the new paragraph after the first clause to:

[…] even though not all fonts use connecting glyphs for the EM DASH, the default behavior is ~~not~~ to not break the line between ~~even though not all fonts use connecting glyphs for the EM DASH~~.

3.50 CJ: Conditional Japanese Starter

CSS Text Level 3 (which supports Japanese line layout) defines three distinct values for its line-break behavior:

➔ This is the unique instance of the hyphenated string "line-break" in UAX #14, compared to 5 instances of "line break behavior", and 28 instances of "line breaking behavior". Please compare to item 3.28 about not skipping the -ing in "line breaking class" either.

I'd suggest synching this instance by changing "line-break behavior" to "line breaking behavior", and changing the five other instances of "line break behavior" similarly to "line breaking behavior".

3.51 CL: Close Punctuation (XB)

The closing character of any set of paired punctuation should be kept with the preceding character, and the same applies to all forms of wide comma and full stop. This is desirable, even when there are intervening space characters, to prevent the appearance of a bare closing punctuation mark at the head of a line.

➔ The first "and" is unnecessary and only makes for a hasty style (like in this sentence).

➔ The comma after "desirable" is semantically wrong, since the statement that this is desirable *per se* is already implied. The point here is to tell *in what other case* it is so. Hence the first three words of the second sentence should not be set off as a clause.

[…] with the preceding character. ~~, and t~~The same applies to all forms of wide comma and full stop. This is desirable~~,~~ even when […]

3.52 CM: Combining Mark (XB) (Non-tailorable)

For most purposes, combining characters take on the properties of their base characters, and that is how the **CM** class is treated in rule **LB9** of this specification. As a result, if the sequence <0021, 20E4> is used to represent a triangle enclosing an exclamation point, it is effectively treated as EX, the line break class of the exclamation mark. If U+2061 CAUTION SIGN had been used, which also looks like an exclamation point inside a triangle, it would have the line break class of **AL**. Only the latter corresponds to the line breaking behavior expected by users for this symbol. To avoid surprising behavior, always use a base character that is a symbol or letter (Line Break **AL**) when using enclosing combining marks (General_Category Me).

➔ Please see {Material}, item 2.10, for proposed material changes to the last sentence.

➔ The parenthetical "Line Break **AL**" should be synched with "line breaking class **ID**" under the heading "AI: Ambiguous (Alphabetic or Ideograph)": (~~L~~line ~~B~~breaking class **AL**)

3.53 CP: Closing Parenthesis (XB)

This class contains just two characters, U+0029 RIGHT PARENTHESIS and U+005D RIGHT SQUARE BRACKET. Characters of class CP differ from those of the CL (Close Punctuation) class in that they will not cause a break opportunity when appearing in contexts like "(s)he." In all other respects the breaking behavior of CP and CL are the same.

➔ The number of "the breaking behavior" and "are" in the last sentence should be the same. To fix it, the verb ought to be turned into singular:

[…] In all other respects the breaking behavior of CP and CL ~~are~~ is the same.

3.54 JV: Hangul V Jamo (XA/XB)

➔ This is the first instance out of two where "XB/XA" (3 instances) is inverted: "(~~XA~~/XB/~~XA~~)". The second is in "ZWJ: Zero Width Joiner (XA/XB) (Non-tailorable)". The two may be corrected at once.

3.55 HY: Hyphen (XA)

Some additional context analysis is required to distinguish usage of this character as a hyphen from its usage as a minus sign (or indicator of numerical range). If used as hyphen, it acts like U+2010 HYPHEN, which has line break class **BA.**

➔ For completeness, the implied "if not" clause could be added explicitly:

[…] If used as hyphen, it acts like U+2010 HYPHEN, which has line break class **BA.** If not, it acts like U+2212 MINUS SIGN, which has line break class **PR.**

3.56 RI: Regional Indicator (B/A/XP)

Pairs of RI characters are used to represent a two-letter ISO 3166 region code.

➔ The number of the subject and the number of the object should be harmonized in either way:

Pairs of RI characters are used to represent ~~a~~ two-letter ISO 3166 region code~~s~~.

Or:

A ~~P~~pair~~s~~ of RI characters ~~are~~ is used to represent a two-letter ISO 3166 region code.

3.57 SG: Surrogate (XP) (Non-tailorable)
In UTF-8 and UTF-32 surrogate code points represent corrupted data and their line break behavior is undefined.

➔ A comma is required after “UTF-32,” and another comma after “data,”.

3.58 SP: Space (A) (Non-tailorable)
The space characters are used as explicit break opportunities; they allow line breaks before most other characters. However, spaces at the end of a line are ordinarily not measured for fit. If there is a sequence of space characters, and breaking after any of the space characters would result in the same visible line, then […]

➔ By using the generic concept “space characters” to (obviously yet implicitly) refer to several instances of the character SPACE, this description creates confusion. The plural here seems unnecessary. I’d suggest changing this to:
~~The space characters~~ SPACE ~~are~~ is used as an explicit break opportunity~~ies~~; ~~they~~ it allows a line break~~s~~ before most other characters. However, ~~spaces~~ SPACE at the end of a line ~~are~~ is ordinarily not measured for fit. If there is a sequence of ~~space~~ SPACE characters, and breaking after any of them ~~space characters~~ would result in the same visible line, then […]

3.59 SP: Space (A) (Non-tailorable)
[…] the line breaking position after the last space character in the sequence is the locally most optimal one.

➔ Uppercasing SPACE and making it unambiguous is discussed in 3.58.

➔ The attribute “optimal” is already a superlative. While we can call something “less than optimal” or “suboptimal”, we cannot tell about two things that one is more optimal than the other. Hence this should be changed to either of the following:
[…] the line breaking position after the last ~~space character~~ SPACE in the sequence is the locally ~~most~~ optimal one.
[…] the line breaking position after the last ~~space character~~ SPACE in the sequence is the locally most ~~optimal~~ suitable one.
[…] the line breaking position after the last ~~space character~~ SPACE in the sequence is the locally ~~most optimal~~ best one.

3.60 SP: Space (A) (Non-tailorable)
In other words, when the last character measured for fit is *before* the space character, any number of space characters are kept together invisibly on the previous line and the first non-space character starts the next line.

➔ The word “space” needs to be uppercased as a character name in all three instances, so as to disambiguate the string "SPACE character(s?)" from the string "space character(s?)" as used in the next note.

➔ The term “character” when referring to SPACE seems unnecessary.

➔ The definite article should be indefinite in the phrase “is *before* ~~the~~ a ~~space~~ SPACE ~~character~~”.

➔ A comma goes amidst the clauses:
any number of ~~space~~ SPACEs ~~characters~~ are kept together invisibly on the previous line, and the first non-~~space~~SPACE character starts the next line.

3.61 SY: Symbols Allowing Break After (A)

The **SY** line breaking property is intended to provide a break opportunity after, […]

➔ This is one of the two instances—mentioned in point 6 of item 3.2—where the singular "line breaking ~~property~~" is mistakenly used instead of "line breaking class".

3.62 WJ: Word Joiner (XB/XA) (Non-tailorable)

The word joiner character is the preferred choice for an invisible character to keep other characters together that would otherwise be split across the line at a direct break.

➔ I'd suggest moving "together" after "keep", deleting "the line at", uppercasing WORD JOINER as a name since it is singular and the other one (ZWNBSP) is not preferred, and deleting the first instance of "character" since it is only a replacement of missing uppercase for the character name.

➔ I note that WORD JOINER is not limited to preventing direct breaks. It works after SPACE as well, where it helps emulate an always justifying no-break space [3]. Hence the word "direct" needs to be deleted, and "line" added thereafter:

The ~~word joiner~~ WORD JOINER ~~character~~ is the preferred choice for an invisible character to keep together other characters ~~together~~ that would otherwise be split across ~~the line at~~ a ~~direct~~ line break.

3.63 XX: Unknown (XP)

The **XX** line break class consists of […] default to the class ID, and are listed in the description of that class. [] Unassigned code positions […] are assigned this line breaking property. The default behavior for this class is identical to class **AL.**

➔ This is one of 47 instances, pointed under 3.28, where "line break class(es)" is mistakenly used instead of "line breaking class(es)" as it stands in 40 instances.

➔ Above all this is the second instance mentioned in point 6 of item 3.2, where the singular "line breaking ~~property~~" is mistakenly used instead of "line breaking class". Given that there are also four instances of "class", the text surreptitiously seems to make the reader get habits with using "property" as a synonym of "class" or "property value". I can't see anything good behind this terminology mashing, and that's why I keep suggesting that this document be thoroughly copy-edited:

The **XX** line breaking class consists of […] default to the class ID, and are listed in the description of that class. [] Unassigned code positions […] are assigned this line breaking ~~property~~ class. The default behavior for this class is identical to class **AL.**

3.64 XX: Unknown (XP)

[…] by assigning characters the property ID or another class.

➔ This instance of singular "property" instead of "class" (see point 6 of item 3.2), is even internally inconsistent as it contradicts itself inside a five-words span. Nowhere in the document is "property" defined as a synonym of "class". But "property value" is. Correct is either:

[…] by assigning characters the property value **ID** or another class.

Or:

[…] by assigning characters the ~~property~~ class **ID** or another class.

The second option is the most straightforward.

3.65 ZW: Zero Width Space (A) (Non-tailorable)
This character is used to enable additional (invisible) break opportunities wherever SPACE cannot be used.

➔ The attribute “invisible” applied to “line break opportunities” probably means that no hyphen will show up when the opportunity is used for a line break. Per definition LD2 a line break opportunity is a position in the text (or “a place”) and is therefore invisible while unused. A space in the text does not unveil a break opportunity since it may be non-breaking. Talking about invisibility is talking about glyphs, not about break opportunities. Perhaps the function of this parenthetical since the earliest (available) draft (revision 0.3) is adverbial: “to invisibly enable additional breaks.”

➔ I think that the original draft got it right when using the verb “provide” rather than “enable”, since break opportunities do exist or they don’t. Something cannot be enabled without existing at all. What may be said “enabled” is the line break itself, but that is ambiguous as of whether it’s about making it possible, or making it happen. Trying to figure out why two revisions later (in revision 5) “provide” was replaced with “enable”, I guess that “enable additional breaks” was preferred over “provide additional break opportunities”, but that the awareness of the ambiguity, and of the difference between actual breaks and mere break opportunities was only able to keep “opportunities”, not to keep “provide”.

➔ As a result, this sentence should probably be set back (reusing “provide” from early drafts) and turned into:
This character is used to invisibly ~~enable~~ provide additional ~~(invisible)~~ break opportunities wherever SPACE cannot be used (and no hyphen is expected).

3.66 5.2 Dictionary Usage
Oxford English Dictionary (1st Edition) **si·lă'bl** where · is a slightly raised middle dot indicating the vowel of the stressed syllable (similar to Johnson’s acute).

➔ The cited acute is actually an acute in the web page so as to prevent font issues, but the explaining text identifies it by its code as a MODIFIER LETTER PRIME: “´ is an oversized U+02B9”:
[…] (similar to Johnson’s ~~acute~~ prime).

3.67 5.2 Dictionary Usage
Oxford English Dictionary (2nd Edition) has gone to IPA **'sIlə(ə)l** where **'** is U+02C8, I is U+026A, and ə is U+0259 (both times). The **'** comes before the stressed syllable. The () indicate the *schwa* may be omitted.

➔ Eliding obvious things is often just fine, but in this document I’d rather use full language:
The () indicate the second *schwa* may be omitted.
Or even:
The ~~()~~ parentheses indicate that the second *schwa* may be omitted.

3.68 5.2 Dictionary Usage
BBC English Dictionary **sIlə<u>l</u>** where <u>I</u> is <U+026A, U+0332> and ə is U+0259. The vowel of the stressed syllable is underlined.

➔ The source shows that the combining underline is emulated using underline markup, and that this is misplaced. Correct would be:
sIlə<u>l</u>” should be: “s<u>I</u>lə~~<u></u>~~l~~</u>~~

3.69 5.2 Dictionary Usage

Collins Cobuild English Language Dictionary **sllәbə°l̲** where l̲ is <U+026A, U+0332> and has the same meaning as in the *BBC English Dictionary*.

➔ Same error and same fix as in 3.68.

3.70 5.2 Dictionary Usage

Some dictionaries use a character that looks like a vertical series of four dots to indicate places where there is a syllable, but no allowable break.

➔ The word "boundary" seems to be missing after "syllable".

3.71 5.3 Use of Hyphen

The rules for treating hyphens in line breaking vary by language. In many instances, these rules are not supported as such in the algorithm, but the correct appearance can be realized by using a *non-breaking hyphen.*

➔ The phrase "as such" is superfluous here. It resolves to: "these rules are not supported *as rules* in the algo." That supposes that they are supported as something else, which they aren't. The lenifying effect of such phrases is fairly current but is not needed here. Actually the principle of using appropriate characters applies throughout, even to spaces, unlike what UAX #14 goes to great lengths to do without reaching any useful goal. There is no need to beg the reader's indulgence, unless the promises to automatically fix defective text actually raised exaggerated expectations.

➔ The alienation effect generated by italicizing the concept of a "*non-breaking hyphen*" seems pointless unless it should appear as an exception, while it is only an implementation of the rule to always use appropriate characters. I'd suggest referring directly to NON-BREAKING HYPHEN instead, as is done for many other characters in the document:

[…] In many instances, these rules are not supported ~~as such~~ in the algorithm, but the correct appearance can be realized by using NON-BREAKING HYPHEN ~~a *non-breaking hyphen*~~.

3.72 5.3 Use of Hyphen

In standard Polish orthography, explicit hyphens are always promoted to the next line if a line break occurs at that location in the text. For example, if, given the sentence "Tam wisi czerwono-niebieska flaga" ("There hangs a red-blue flag"), the optimal line break occurs at the location of the explicit hyphen, an additional hyphen will be displayed at the beginning of the next line like this:

➔ If in Polish and Portuguese, the explicit hyphen is promoted to the next line, the additional hyphen is the hyphenation hyphen at line end, not the explicit hyphen at the next line start:

[…] an additional hyphen will be displayed at the ~~beginning~~ end of the ~~next~~ previous line […]

3.73 5.4 Use of Soft Hyphen

In German and Swedish, a consonant is sometimes doubled: Swedish "tuggummi"; hyphenates into "tugg- / gummi".

➔ The semicolon after the example is superfluous.

➔ Material changes are suggested in {Material}, item 2.12.

3.74 5.4 Use of Soft Hyphen

When a SHY is used to represent a possible hyphenation location, the spelling is that of the word without hyphenation: "tug<SHY>gummi".

➔ The indefinite article before “SHY” is pointless because “SHY” is a character identifier, not an abbreviation of a concept. One would say: “when a soft hyphen is used”, but here it goes: “when SOFT HYPHEN is used”.

➔ The term “location” instead of “position” is less accurate, or maybe downright inaccurate, and anyway here it induces a weird echo.

➔ The last phrase could be streamlined, while synching the example with {Material}, item 2.12:

When ~~a~~ SHY is used to represent a possible hyphenation ~~location~~ position, ~~the spelling is that of~~ the word’s spelling is otherwise unaltered ~~without hyphenation~~: “~~tug~~brän<SHY>~~gummi~~nässla”.

3.75 5.4 Use of Soft Hyphen

Sometimes it is desirable to encode text that includes line breaking decisions and will not be further broken into lines. If such text includes hyphenations, the spelling needs to reflect the changes due to hyphenation: “tugg<U+2010>/ gummi”, including the appropriate character for any inserted hyphen.

➔ The example should stand at the end, not in the middle. About picking another example, see 3.72.

➔ The slash representing a line break should be *surrounded* by spaces like in the preceding examples. (Here, the space before is missing.)

➔ About changing “<U+2010>” to “<HY>”, please see suggestion 2.8.

➔ The wording could be somewhat streamlined, for example as:

[…] the spelling needs to reflect the changes due to hyphenation, with any appropriately hard-coded hyphen: “~~tugg~~bränn<U+2010> / ~~gummi~~nässla”~~, including the appropriate character for any inserted hyphen~~.

3.76 5.4 Use of Soft Hyphen

Hyphenation, and therefore the SHY, can be used with the Arabic script. If the rendering system breaks at that point, the display—including shaping—should be what is appropriate for the given language.

➔ The first sentence sounds somewhat patronizing and should rather be streamlined by merging it with the first phrase of the second sentence. That could result in turning the snippet into:

When ~~H~~hyphenation~~, and therefore the SHY, can be~~ is used with the Arabic script, and~~. If~~ the rendering system breaks at ~~that point~~ SHY, the display—including shaping—should be what is appropriate for the given language.

3.77 5.4 Use of Soft Hyphen

However, when a word contains an explicit SHY, it is customarily treated as overriding the action of the hyphenator for that word.

➔ A word may contain more than one SHY. Using the plural (without adding “several”) would be safe, but for clarity, “one or more” may be added.

➔ The pronoun “it” is meant to refer to “SHY”, not to “word”, that it would normally refer to, since “A word” is the subject. From personal, this pronoun should be turned into demonstrative:

However, when a word contains ~~an~~ one or more explicit SHYs, ~~it~~ these ~~is~~ are customarily treated as overriding the action of the hyphenator for that word.

3.78 5.6 Tibetan Line Breaking

Phrases are often metrical—that is, written after every *N* syllables—and a new sentence can often start within the middle of a phrase.

➔ Since it appeared in version 4.1.0 (revision 17, 2005), this sentence yields three flaws that went uncaught despite it came under scrutiny at least in revision 23 when the term “phrases” stopped being scare-quoted.

➔ Tibetan phrases are **separated** rather than “**written** after every *N* syllables”, or they are written **in** *N* syllables each. I’d suggest using the verb of the next phrase to replace “**written**” with **starting.**

➔ Tibetan sentences **can** start, or they **often** start. Telling that they “**can often** start” seems pleonastic.

➔ Tibetan sentences can start **within** a phrase, or **in the middle of** a phrase; “**within the middle of** a phrase” is a pleonasm, again. This cumulation of pleonasms unveils that the author of this information was baffled (which is understandable from an outside point of view but shouldn’t flow into the Standard). More correct:
Phrases are often metrical—that is, ~~written~~ starting after every *N* syllables—and a new sentence can ~~often~~ start within ~~the middle of~~ a phrase.

3.79 5.6 Tibetan Line Breaking

Traditionally there is nothing akin to a paragraph in Tibetan text. It is typical to have many pages of text without a paragraph break—that is, without an explicit line break. The closest thing to a paragraph in Tibetan is a new section or topic starting with U+0F12 or U+0F08. However, these occur inline: one section ends and a new one starts on the same line, and the new section is marked only by the presence of one of these characters.

➔ Rather than attempting to synch the style (“nothing akin to”, “the closest thing”) with the rest of UAX #14, I’d first mention that the effect of the last “and” is to make the style breathless. It would be useful if not preceded by a comma nor followed by the redundant “the new section”. As it stands, there should rather be a sentence break between “the same line” and “the new section”.

➔ Since occasionally a line break may occur next to TIBETAN MARK RGYA GRAM SHAD or TIBETAN MARK SBRUL SHAD, the sentence after the colon should use “may”.

➔ I’d suggest moving “on the same line,” so as to be able to do some streamlining:
[…] However, these occur inline: on the same line, one section may end~~s~~ and a new one start~~s on the same line~~, ~~and the new section is~~ marked only by the presence of one of these characters.

3.80 6 Line Breaking Algorithm

➔ This is the last section heading with a comment in the source code repeating the heading:

```
<!--
   -
   -   6    Line Breaking Algorithm
   -
   -->
```

The point comes up when looking at the four SPACEs after the section number, unveiling that the preference for setting off the section numbers—fairly current in publishing for readability reasons—was real and only failed to transpose into display. A current fix elsewhere in the document, such as after definition and rule numbers, was to append " ". LD1 had even two of them, plus another one:

```
<p><i>LD1   Line Fitting:</i>  The process of determining […]
```

That changed for revision 34: 60 NBSPs out of 175 (named entities) were deleted in the document.

➔ I’d suggest using EM SPACE instead of SPACE after the section number for all section and subsection headings, and EN SPACE after all definition IDs and rule IDs. I’m currently using plain Unicode to do such spacing (also "➔<EMSP>"), but for visibility the following uses hex entities:

```
<h2>6  <a name="Algorithm" href="#Algorithm">Line Breaking
Algorithm</a></h2>
```

3.81 6 Line Breaking Algorithm
The formal statement of each line breaking rules consists either of a remap rule or of one or more regular expressions containing one or more line breaking classes and one of three special symbols indicating the type of line break opportunity:

➔ The word “rules” should be singular.

➔ Given that rule LB26 (see 3.99) and example 7 in subsection 8.2 (see 3.106) are careful about explaining what the pipe character stands for—a symbol used as soon as in LB6—I’d suggest providing that information right here instead (surrounding the pipe with spaces, for instance <NNBSP>).

➔ The keys of “^”, “?” and “*”, symbols used from LB24 on, could also be moved here alongside:
The formal statement of each line breaking rule~~s~~ consists either of a remap rule or of one or more regular expressions containing one or more line breaking classes (where “ | ” stands for logical OR, “^” after “[” for logical NOT, “?” is the “0 or 1” quantifier, and “*” is “0 or more”) and one of three special symbols indicating the type of line break opportunity:

3.82 6 Line Breaking Algorithm
× No break allowed at the indicated position

➔ I’d suggest synching terminology with LD10, which results in streamlined wording:
× ~~No b~~Break ~~allowed~~ prohibited at the indicated position

3.83 6 Line Breaking Algorithm
The examples for each rule use representative characters, where ‘H’ stands for an ideographs, ‘h’ for small kana, and ‘9’ for digits.

➔ For logic and consistency I’d suggest using singular throughout, titlecasing “Kana”, and adding alphabetic/abjad letters. I must confess that I’m unable to spot a single example using a small Kana, but I do find an instance of “a” (under LB22):
The examples for each rule use representative characters, where ‘a’ stands for a letter, ‘H’ ~~stands~~ for an ideograph~~s~~, ‘h’ for a small ~~k~~Kana, and ‘9’ for a digit~~s~~.

3.84 6.2 Tailorable Line Breaking Rules
LB1

➔ The hyperlinks fail to span over the rule identifiers, unlike what is done for the definition identifiers, because the closing tags were not moved to the end of the IDs when anchors and hyperlinks were added in revision 17:
`<p><i><a name="LB1"></a>LB 1  Assign a line breaking class […]`
In the actual source code, the close tags need to be moved so as to enable the users of UAX #14 to pick links right at rule start.

➔ About optionally adding <ENSP>, please see item 3.80.
`<p class="rule"><a name="LB1" href="#LB1">~~</a>~~<b>LB1</b></a>  Assign a line breaking class […]`

3.85 6.1 Non-tailorable Line Breaking Rules
LB2 *Never break at the start of text.*

sot ×.

➔ In UAX #14 this is the first centered regular expression. Since the document calls the HTML class attribute already over 1,200 times, I'd suggest adding also a class for centered (regexes) so as to streamline the source by replacing about 90 instances of:

```
<p style="text-align:center">[…]</p>
```

With:

```
<p style="text-align:center" class="regex">[…]</p>
```

In reports-v2.css:

```
.regex { text-align:center; }
```

3.86 6.1 Non-tailorable Line Breaking Rules

LB5 *Treat CR followed by LF, as well as CR, LF, and NL as hard line breaks.*

CR × LF
CR !
LF !
NL !

➔ A regex in the next rule (LB6) starts collapsing classes using logical OR ("|"). There seems to be no reason not to do the same here:

CR × LF
(CR | LF | NL) !
~~LF !~~
~~NL !~~

3.87 6.1 Non-tailorable Line Breaking Rules

LB9 *Do not break a combining character sequence; treat it as if it has the line breaking class of the base character in all of the following rules. Treat ZWJ as if it were CM.*

➔ Put "has" into conditional form like "were", and like "had" four lines below.

3.88 6.1 Non-tailorable Line Breaking Rules

LB9

At any possible break opportunity between CM and a following character, CM behaves as if it had the type of its base character. Note that despite the summary title, this rule is not limited to standard combining character sequences. For the purposes of line breaking, sequences containing most of the control codes or layout control characters are treated like combining sequences.

➔ I'd suggest replacing "Note that" with a paragraph break:

At any possible break opportunity between CM and a following character, CM behaves as if it had the type of its base character. ~~Note that d~~ [PARAGRAPH BREAK]
Despite the summary title, this rule is not limited to standard combining character sequences. For the purposes of line breaking, sequences containing most of the control codes or layout control characters are treated like combining sequences.

3.89 6.1 Non-tailorable Line Breaking Rules

LB11 *Do not break before or after Word joiner and related characters.*

➔ For consistency, "~~W~~*word joiner*" should be lowercase.

3.90 6.2 Tailorable Line Breaking Rules

LB12a *Do not break before NBSP and related characters, except after spaces and hyphens.*

[^SP BA HY] × GL

The expression [^SP, BA, HY] designates any line break class other than SP, BA or HY. The symbol ^ is used, instead of !, to avoid confusion with the use of ! to indicate an explicit break. Unlike the case for WJ, inserting a SP overrides the non-breaking nature of a GL.

➔ The negated members in the class are not followed by a comma inline neither.

➔ The key of the symbol ^ is now suggested to be placed at section start, so that at least the point about the exclamation mark can be deleted here.

➔ I'd suggest separating into two paragraphs the notational information and the information about the effect of the rule, because the two are unrelated.

➔ In a full sentence such as "Unlike what is the case for WJ", "for" is correct, but in isolation one rather refers to "the case of WJ".

➔ When preceding an initialism, the form of the indefinite article usually depends on the letter-by-letter spelling of the abbreviation, here "ess-pee":

The expression [^SP~~,~~ BA~~,~~ HY] designates any line break class other than SP, BA or HY. ~~The symbol ^ is used, instead of !, to avoid confusion with the use of ! to indicate an explicit break.~~ [PARAGRAPH BREAK]

Unlike the case ~~for~~ of WJ, inserting an SP overrides the non-breaking nature of a GL.

3.91 6.2 Tailorable Line Breaking Rules

LB13 *Do not break before 'J' or '!' or ';' or '/', even after spaces.*

× CL
× CP
× EX
× IS
× SY

➔ This could be likewise streamlined (unless the test file refers to lines, that would be numbered).

× (CL | CP | EX | IS | SY)

~~× CP~~
~~× EX~~
~~× IS~~
~~× SY~~

3.92 6.2 Tailorable Line Breaking Rules

LB18 *Break after spaces.*

SP ÷

➔ In my opinion this regex doesn't catch runs of multiple spaces providing only one break opportunity (after the last one, and hanging into the margin). If there should be a plus sign here as in a comment to LD9, and it's simply missing, adding it is a minor edit:

SP~~+~~ ÷

3.93 6.2 Tailorable Line Breaking Rules

LB19 *Do not break before or after quotation marks, such as ' " '.*

➔ I think that the example is not needed. I'd suggest deleting it the more as the spaced-off single apostrophe quotation marks look weird in my opinion.

3.94 6.2 Tailorable Line Breaking Rules

LB20 […] Conditional breaks should be resolved external to the line breaking rules.

➔ Make “external” an adverb, and eventually change “should” to “shall”, or turn the whole into:

The resolution of ~~C~~conditional breaks ~~should be resolved~~ is external to the line breaking rules.

3.95 6.2 Tailorable Line Breaking Rules

LB21a *Don't break after Hebrew + Hyphen.*

➔ In the actual version, this is the first rule using colloquial “Don’t”. It has been added in revision 27 after all “Don’t” had been changed to “Do not” in revision 16 without being highlighted as changed. The only two instances of “don’t” are now in LB21a and LB21b. I suggest synching language level accordingly.

3.96 6.2 Tailorable Line Breaking Rules

LB22 *Do not break between two ellipses, or between letters, numbers or exclamations and ellipsis.*

(AL | HL) × IN
EX × IN
(ID | EB | EM) × IN
IN × IN
NU × IN

➔ These regexes either should be sorted in rule order, or would be better collapsed.

Sorted in the order they occur in the rule title:

IN × IN
(AL | HL) × IN
~~EX × IN~~
(ID | EB | EM) × IN
~~IN × IN~~
NU × IN
EX × IN

Collapsed into a single regex, with classes sorted in rule order:

(IN | AL | HL | ID | EB | EM | NU | EX) × IN
~~EX × IN~~
~~(ID | EB | EM) × IN~~
~~IN × IN~~
~~NU × IN~~

Collapsed, with classes sorted alphabetically:

(AL | EB | EM | EX | HL | ID | IN | NU) × IN
~~EX × IN~~
~~(ID | EB | EM) × IN~~
~~IN × IN~~
~~NU × IN~~

3.97 6.2 Tailorable Line Breaking Rules

LB24 *Do not break between numeric prefix/postfix and letters, or between letters and prefix/postfix.*

(PR | PO) × (AL | HL)
(AL | HL) × (PR | PO)

In general, it is recommended to not break lines inside numbers of the form described by the following regular expression:

(~~-~~PR | PO)~~-~~? (~~-~~OP | HY~~-~~)~~-~~? NU (NU | SY | IS)~~-~~* (CL | CP)~~-~~? (~~-~~PR | PO)~~-~~?

➔ Regex quantifiers are usually not spaced off. I'd suggest deleting those spaces (*for brevity above*). Another instance of spacing-off is reported in 3.106.

➔ While in UAX #14 the pipe has SPACE on either side for readability, parentheses are not spaced off. Here, some of them are inconsistently spaced off. These extra spaces could also be deleted.

3.98 6.2 Tailorable Line Breaking Rules

LB25 *Do not break between the following pairs of classes relevant to numbers:*

CL × PO
CP × PO
CL × PR
CP × PR
NU × PO
NU × PR
PO × OP
PO × NU
PR × OP
PR × NU
HY × NU
IS × NU
NU × NU
SY × NU

➔ Listing 14 pairs individually is non-obvious when in subsection 8.2, Example 7, and elsewhere regexes are massively streamlined using the pipe character for logical OR. I'd suggest doing the same here:

(CL | CP | NU) × (PR | PO)
~~CP × PO~~
~~CL × PR~~
~~CP × PR~~
~~NU × PO~~
~~NU × PR~~
(PR | PO) × (OP | NU)
~~PO × NU~~
~~PR × OP~~
~~PR × NU~~
(NU | IS | HY | SY) × NU
~~IS × NU~~
~~NU × NU~~
~~SY × NU~~

3.99 6.2 Tailorable Line Breaking Rules

LB26 *Do not break a Korean syllable.*

JL × (JL | JV | H2 | H3)
(JV | H2) × (JV | JT)
(JT | H3) × JT

where the notation (JT | H3) means JT or H3. The effective line breaking class […]

➔ If anywhere, this notation key should be provided at the start of the section, as suggested in 3.81, because interspersing it amidst the first rule for Korean syllable blocks by appending it to the last regex instead seems unwise:
~~where the notation (JT | H3) means JT or H3.~~ The effective line breaking class […]

3.100 6.2 Tailorable Line Breaking Rules
LB31 *Break everywhere else.*

ALL ÷
÷ ALL

➔ Section 10 and the test are using "any" like in section 10 and the test, lowercase like "sot" and "eot", instead of "ALL". As a minimal fix I suggest synching LB31 with UCD, because "any" seems better:

~~ALL~~ any ÷
÷ any ~~ALL~~

➔ "ALL" is usually implied when one side is blank, and it is not a line breaking class anyway. Why not express this rule by:

~~ALL~~ ÷
~~÷ ALL~~

➔ If for any reason that is inconvenient, all blank sides ought to be filled in with the string "any" throughout. After that, the regexes of LB31 will become:

~~ALL~~ any ÷ any
~~÷ ALL~~

3.101 8 Customization
A real-world line breaking algorithm has to be tailorable to some degree to meet user or document requirements.

➔ Given that a significant number of features of the Unicode Line Breaking Algorithm seem to be in disconnect with real world and are typically not implemented as-is, invoking the "real world" here seems to me somewhat ironic. I think that this attribute is too problematic in UAX #14 (and perhaps in any standards document) and can be losslessly deleted:
A ~~real-world~~ line breaking algorithm has to be tailorable […]

3.102 8 Customization
The remainder of this section gives an overview of common types of tailorings.

➔ Singular as in the following heading is correct.

➔ This is not all. Some examples are also given:
[…] gives an overview of common types of tailoring~~s~~ and some examples.

3.103 8.1 Types of Tailoring
This is useful in cases where the line breaking properties of one class of characters are occasionally lumped together with the properties of another class to achieve a less restrictive line breaking behavior.

➔ This is the only instance where “line breaking properties” may not be replaced with “line breaking property values” nor with “line breaking classes” due to colliding terms and telescoped concepts. To fix it, we can note that it looks like sort of colloquial wording and may be streamlined somehow:
This is useful when two line breaking classes are merged to achieve a less restrictive line breaking behavior.

➔ After that, all other remaining instances of “~~line breaking properties~~” need to be replaced with “line breaking classes” following point 6 of item 3.2.

3.104 8.2 Examples of Customization
Example 3. […] Space-based layout is common in magazines and other informal documents with ragged margins, while books, with both margins justified, use the other type, as it affords more line break opportunities and therefore leads to better justification.

➔ Magazines may not be called “informal documents.”

➔ Ragging both margins is less common than having one margin ragged, and the other one aligned.

➔ Korean not being in the habits of expanding spaces for the purpose of justification is implied by the information provided in UAX #14 so far. Feedback from the original contributor about making it explicit would be useful.

➔ Usually when lines are justified, both margins are aligned, and justification is the better the less the spaces are expanded. In the absence of spaces to expand, more other break opportunities are needed:
[…] Space-based layout is common in magazines and ~~other~~ informal documents with at least one margin ragged ~~margins~~, as justification by expanding spaces is uncommon, while books, with both margins ~~justified~~ aligned, use the other type, as it affords more line break opportunities and therefore leads to better justification in the absence of spaces.

3.105 8.2 Examples of Customization
Example 4. In a Far Eastern context […]

➔ As pointed in 3.37, “~~Far Eastern~~” should be replaced with “East Asian”.

3.106 8.2 Examples of Customization
Example 7. Regular expression-based line breaking engines might get better results using a tailoring that directly implements the following regular expression for numeric expressions:

(-PR | PO)-? (-OP | HY-)-? NU (NU | SY | IS)-* (-CL | CP-)-? (-PR | PO)-?

This is equivalent to replacing the rule LB25 by the following tailored rule:
Regex-Number: Do not break numbers.

(PR | PO) × (-OP | HY-)? NU
(-OP | HY-) × NU
NU × (NU | SY | IS)
NU (NU | SY | IS)* × (NU | SY | IS | CL | CP-)
NU (NU | SY | IS)* (CL | CP)? × (PO | PR)

~~In these tailored rules, (PR | PO) means PR or PO, the Symbol “?” means 0 or one occurrence and the symbol “*” means 0 or more occurrences.~~ The last two rules can have a left side of any non-zero length.
When the tailored rule is used, the regular expressions in LB13 need to be tailored as follows:

➔ Regex quantifiers are usually not spaced off, and in example 7 they are only so in the first regex. (This is similar to 3.97.) *For brevity, corrections are made directly in this quotation.*

➔ While in UAX #14 the pipe has SPACE on either side for readability, parentheses are not spaced off. Here, some of them are inconsistently spaced off. These extra spaces should also be deleted.

➔ Keys to regex syntax are now suggested to be placed at the start of section 6, so that the above can be deleted.

➔ In the last sentence, “need” logically refers to the regexes, hence the (grammatically incorrect) plural, so I suggest fixing grammar by completing.

3.107 8 Customization

If this is not done, single digits might be handled by rule **LB13** before being handled in the regular expression. ~~In these tailored rules [^NU] designates any line break class other than NU. The symbol ^ is used, instead of !, to avoid confusion with the use of ! to indicate an explicit break.~~

➔ Repeating the key again is not useful and only helps overcrowding the spec, beside being untrue since it applies not only to “these tailored rules.”

➔ Since “^” is standard in regexes after "[", I think that the explanation can be spared.

3.108 10 Testing

For those who do implement the default breaks as specified in this annex, plus the tailoring of numbers described in Example 7 of *Section 8.2*, *Examples of Customization*, and wish to check that ~~that~~ their implementation matches that specification, a test file has been made available in [Tests14].

➔ Duplicate highlighted for deletion.

References

[1] Deborah Anderson, Ken Whistler, Roozbeh Pournader, Lisa Moore, Liang Hai, *Recommendations to UTC #160 July 2019 on Script Proposals,* p. 14, #20 [L2/19-286].

[2] Patrick Andries, *Unicode 5.0 en pratique : codage des caractères et internationalisation des logiciels et des documents*, Dunod, Paris, 2008 [Read on Google Books].

[3] Jakub Stachu, in “Non-breakable space justification in Word 2016”, page 3, Microsoft Community, 2017.

[4] Jukka “Yucca” Korpela, “*Unicode line breaking rules: explanations and criticism”*, *IT and communication,* 2000-10-11, 2005-08-28/29, 2008-05-21.

[5] Shriramana Sharma, “NBSP supposed to stretch, right?”, Unicode Public Mailing List, December 2019.

Acknowledgments

Thanks to everyone who directly or indirectly helped put this paper together.

Thanks to Microsoft for Word Online, OneDrive, VS Code and MSKLC.

Thanks to Google for Google Chrome, Google Search, Google Books, Google Translate and Gmail.