L2/20-005

Proposal to make material changes 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 in an extra section, and downplaying the former in the process. This paper focuses on material changes only. Formal edits are suggested alongside but separately in *Proposal suggesting formal edits to UAX #14* (short {Formal}).

Another issue with L2/19-317 was that material suggestions were partly based on assumptions made about Mongolian. For this new proposal by contrast, I do not make any such assumptions.

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

High priority

UAX #14 still maintaining that FIGURE SPACE is preferred as a group separator in numbers may seem to interfere with CLDR, but in the first place it contradicts both the Unicode Standard (see *Background* section in *Proposal to synchronize the Core Specification*) and the SI standard (BIPM) specifying that numbers be grouped using a (no-break) thin space (see L2/19-112). Rather than working, the outdated UAX #14 state of the art is going unnoticed, perhaps because neither standard is taken seriously enough.

The rising importance of NARROW NO-BREAK SPACE on keyboard layouts brings that risky in-limbo status to an end. A failure to correct or update the Unicode Standard would be detrimental to all parties: Locals could be diverted from using better keyboard layouts, the Unicode Standard would be at risk of reputational damages, and locales would lose even more credit for proving unable to persuade Unicode Technical Committee to get the errors fixed. An improbable race with only losers on the finish line.

Disclaimers

This proposal not making any assumptions about Mongolian, should not be misinterpreted as unconcerned, and any suspicion that it aims at depriving Mongolian of a format character would be unfounded.

Mongolian quitting NARROW NO-BREAK SPACE as a format control and keeping it as a general-use whitespace at the user’s convenience is sometimes considered a precondition for other scripts to freely use NARROW NO-BREAK SPACE. The Unicode Code Chart of the block General Punctuation is not very clear about NARROW

NO-BREAK SPACE being a general-use space (please see *Proposal to make focused changes to the Code Charts text* about fixing that), but Unicode's intention at encoding time was unambiguous (see p. 3 of L2/19-112). — Item #20 in L2/19-286 *Recommendations to UTC #160 July 2019 on Script Proposals*, from Deborah Anderson, et al. [1] showcases that this is still valid today.

The importance of the issues at stake makes me think that there is a need of a huge collaborative effort in an international working group, probably distinct from the differently focused ISO/IEC/JTC1/SC2/WG2. Such a process would be unrealistic; it was also pointless in the environmental context of 2019, and would be a nuisance in 2020 as the climate crisis is unfolding. On the other hand I've been directed to not share or submit drafts to Unicode officials, but to rather submit final versions directly to Unicode. This is because proposals from the outside are better endorsed downstream than upstream, with an upside of more streamlined coordination and better transparency. — Please see also section *Background*.

Apologizing, I've thankfully come into the benefit of adding a skipped notation of a deleted sentence partly reused otherwise ("When NARROW NO-BREAK SPACE occurs in French text, it should be interpreted as an 'espace fine insécable'.") and correcting a remaining typo in "theses spaces." In these proposals I'm almost continuously spotting and correcting all sorts of errors and mistakes, so I'm quite unsure whether that would ever come to an end, even if I had plenty of time. I beg the readers' indulgence for the remainder.

Scope

This proposal has focus on NARROW NO-BREAK SPACE and its place in the *The Unicode Standard* and in UAX #14, and on other spaces. Correcting information is vital in documenting existing and upcoming input methods, especially keyboard layouts. These changes are grouped in section *1 Focused changes.*

Additionally, a comprehensive review of UAX #14, conducted for consistency as part of {Formal}, brought up also a number of material suggestions, listed in section *2 Collected changes.* Part of these, related to design, have been moved to the upcoming *Proposal to reengineer spaces and punctuation in UAX #14.*

In this proposal and the other one, some sections were added so as to keep in synch other parts of the Standard and to fix the issue about visibility if not de-facto effectiveness of the suggested changes. Noticing that such aggregation of suggestions contradicts Unicode proposal design rules, I've made them separate proposals for simultaneous submission. Please see *Proposal to synchronize the Core Specification,* related to subsection 1.1, and *Proposal to make focused changes to the Code Charts text,* where section 7 is much about NNBSP. — Please see also *Proposal to extend support for abbreviations, Proposal to synchronize two glyphs in the Core Specification, Proposal to synchronize seven glyphs in the Code Charts,* and *Proposal to ensure maximum visibility of changes to UAXes*.

Technical

The reference version of UAX #14 is current latest version 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.

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.

Some proposed changes are non-trivial. The rationales of those changes may span over several paragraphs, trying to cater for high demands. However, rationales are expected to be brief and to-the-point. In those cases, the "Rationale" is a mere abstract, and the full rationale is appended as "Details". If convinced sooner, the reader is welcome to skip the remainder.

1. Focused changes

1.1 GL: Non-breaking ("Glue")

Change from:

NO-BREAK SPACE is the preferred character to use where two words are to be visually separated but kept on the same line, as in the case of a title and a name "Dr.<NBSP>Joseph Becker". When SPACE follows NO-BREAK SPACE, there is no break, because there never is a break in front of SPACE.

NARROW NO-BREAK SPACE has exactly the same line breaking behavior as NO-BREAK SPACE, but with a narrow display width. The MONGOLIAN VOWEL SEPARATOR acts like a NARROW NO-BREAK SPACE in its line breaking behavior. Both of these characters are regularly used in Mongolian text, where they participate in special shaping behavior, as described in *Section 13.5, Mongolian* of [Unicode].

When NARROW NO-BREAK SPACE occurs in French text, it should be interpreted as an "espace fine insécable".

Change to:

NO-BREAK SPACE has exactly the same behavior as SPACE in horizontal justification, but without providing any line break opportunity. It is ~~the preferred character to~~ used where two words are to be visually separated but kept on the same line, as in the case of a title and a name: "Dr.<NBSP>Joseph Becker", provided that NBSP is not tailored as fixed-width. Otherwise, regular NBSP is emulated by <SP, WJ>. When SPACE follows NO-BREAK SPACE, there is no break, because there never is a break in front of SPACE.

NARROW NO-BREAK SPACE has ~~exactly~~ only the same line breaking behavior as NO-BREAK SPACE, while it does not change in width when horizontal justification is enabled. This is the preferred space to use where a non-breaking THIN SPACE is required. Examples include grouping digits in locales using space as a group separator, and setting off certain punctuation characters in French text, where it is currently called "*espace fine* (*insécable*)" [literally "(no-break) thin space" (supposed to be always non-breaking)]. Consistently with the most current fonts, NNBSP is best thought of as a non-breaking version of THIN SPACE.

The MONGOLIAN VOWEL SEPARATOR acts like a NARROW NO-BREAK SPACE in its line breaking behavior. Both of these characters are regularly used in Mongolian text, where they participate in special shaping behavior, as described in *Section 13.5, Mongolian* of [Unicode].

~~When NARROW NO-BREAK SPACE occurs in French text, it should be interpreted as an "espace fine insécable".~~

Rationale:

Unlike the *nature* of NNBSP, that UAX #14 has recently (version Unicode 12.0.0) stopped misrepresenting (ahead of TUS), the *usage* of NNBSP is still misrepresented here, as opposed to what is found in TUS since Unicode 7.0.0 (see *Narrow No-Break Space*), and in the first place since Unicode 11.0.0 (see *Space Characters*). Spacing off certain punctuation characters in French text using NNBSP seals the failure of UAX #14 to usefully replace non-breaking fixed-width spaces. Please see also *Proposal to synchronize the Core Specification* and the upcoming *Proposal to reengineer spaces and punctuation in UAX #14*.

Details:

One proposed change here is to not present the "espace fine insécable" as kind of a French exception, and to not use the term "interpreted", since NARROW NO-BREAK SPACE is the *regular* no-break thin space for use in any locale, and actually to be used in all locales needing a no-break thin space. Since a space is used as a group seoparator, numerous other locales using various scripts are expected to use NARROW NO-BREAK SPACE (see below). The International System of Units recommends a no-break thin space. In the Unicode Standard there is no other way of representing an interoperable one-character no-break thin space than using NARROW NO-BREAK SPACE. See L2/98-268R cited in L2/19-112: "The concept of the Mongolian space (a non-breaking space, narrower than a normal non-breaking space, and contrasting with it in usage) could be of use in other scripts as well; therefore it is better to make this a general use punctuation character, rather than limiting it to the Mongolian script."

By contrast, whether a title and a name—or a Polish one-letter preposition and its noun [3]—are kept on the same line using NBSP depends on the environment. It may also depend on the locale. In Word (except in Word 2013) [3], where NO-BREAK SPACE is tailored as fixed-width, that is not the preferred space when writing Polish [3]. Instead, the preferred space is an ordinary space followed by a break preventer: ZWNBSP in Word 2016 [3], WJ today and in the Standard.

One reason why the tailoring of NO-BREAK SPACE was re-enabled in Word 2016 at the expense of Polish and other locales [3] is that after 25 years of Unicode, users still couldn't rely on any (not too large like FIGURE SPACE) fixed-width no-break space to replace the non-standard usage of NO-BREAK SPACE for the purpose of French punctuation spacing and in all those locales using a space as a group separator. As a consequence, I'd suggest that the guidance as of the preferred space to keep two words on the same line be completed with a conditional clause and an alternative option: ", provided that NBSP is not tailored as fixed-width. Otherwise, regular NBSP is emulated by <SP, WJ>."

In these circumstances, moving "is the preferred space to use" one paragraph down is not merely an editorial suggestion, but a fully content-centered fix required to put the stress in the right place. In turn, the use of the word "exactly" in the first sentence of the second paragraph is inflating the equality of the line breaking behavior to such an extent that it obfuscates the dissimilarity of the two spaces' behavior in line justification, which is actually what matters even in this UAX, and especially here, to reflect what is laid out in *Section 3 Introduction.* The right place for this adverb is in my opinion in the extra information that should be added prominently, underscoring that SPACE and NO-BREAK SPACE have exactly the same behavior in horizontal justification.

The second paragraph starts talking about NARROW NO-BREAK SPACE before switching to MONGOLIAN VOWEL SEPARATOR, and then to both, while the next brief paragraph is again about NARROW NO-BREAK SPACE only. This is for historical reasons, since the third paragraph has been added later, in 2007. Rearranging these two paragraphs so as to group information about NARROW NO-BREAK SPACE in one paragraph, while giving an extra paragraph to characters used in Mongolian, seems appropriate now.

The next step is then to explain how dissimilar NO-BREAK SPACE and NARROW NO-BREAK SPACE are from the justification viewpoint, despite they have misleadingly close names. In fact (but that does not need to be spoken out in this UAX), “NARROW NO-BREAK SPACE” is a misnomer given instead of the accurate name *NO-BREAK THIN SPACE. It is a THIN SPACE that is non-breaking, rather than a NO-BREAK SPACE that is narrow. The way up to the then-new space is not “SPACE ➔ NO-BREAK SPACE ➔ NARROW NO-BREAK SPACE”; it is “SPACE ➔ THIN SPACE ➔ *NO-BREAK THIN SPACE”, because the display width precedes the line-breaking behavior. Among the three parameters ①line breaking behavior, ②line justification behavior and ③display width, NARROW NO-BREAK SPACE has two in common with THIN SPACE (narrow display width and fixed width), while it has only line breaking behavior in common with NO-BREAK SPACE. Hence, again, it is not a “narrow NO-BREAK SPACE” but a “no-break THIN SPACE,” regardless whether in many fonts it is actually either a no-break FOUR-PER-EM SPACE or anything in that range, probably due to initial underspecification in the Standard.

The problem becomes even clearer when looking at the French translations. There is also another French translation, hence “currently” added as a caveat before the first one, cited in the Standard for being widely in use: “*espace fine insécable*” (italicized as non-English text) translates literally to “no-break thin space”. Parentheses around “*insécable*” are in fact necessary, because “*la fine*” was ever thought of as non-breaking. Only in the Unicode era need we to distinguish between the non-breaking one and the breaking one. Sometimes “*espace fine insécable*” is called a pleonasm, hence the parentheses. The other translation is found in the French Code Charts: “*ESPACE INSÉCABLE ÉTROITE*”, faithfully translating the misnomer NARROW NO-BREAK SPACE. There is of course a reason why Unicode named it the other way around: In front of a THIN SPACE correctly tailored as non-breaking, another “no-break thin space” is confusing. But that confusion is actually salvatory: Tailored THIN SPACE should be definitely dismissed as not interoperable, since Unicode is—or was intended to be—all about interoperability.

Adding an English translation of the French term is good practice, and in this particular case it is extremely useful as it shows how NNBSP is called in the locale that in Latin script makes the most extensive use of NNBSP. Inside the English-translation brackets, the explanation of the parentheses surrounding “*insécable*” is consistently parenthesized.

I suggest replacing “~~is a narrow version of NO-BREAK SPACE~~” deleted for version Unicode 12.0.0. The best place to do so is after the information provided so far, in the form of a conclusion: “Consistently with the most current fonts, NNBSP is best thought of as a non-breaking version of THIN SPACE.” That is based on the state of the art found in Arial and Times New Roman (see test on page 12 of *Proposal to make focused changes to the Code Charts text*) and conveys two recommendations:

1. The best pick for the width is to give NNBSP exactly the width of THIN SPACE.
2. NNBSP is not justifying, as opposed to the standard behavior of NBSP [3][5].

There is this important difference in line justification behavior to carve out, because it is too often forgot when people loosely recommend NO-BREAK SPACE as a group separator, not noticing that NO-BREAK SPACE puts numbers at risk of being torn apart as spaces expand (see examples on page 4 of L2/19-112). As a matter of consequence, legacy environments unable to handle NARROW NO-BREAK SPACE due to this character's late encoding, and websites designed for compatibility with legacy user agents, consistently disable line justification when delivering HTML source code without styling tricks. Simply turning off horizontal justification is surely not, however, what good craftmanship is expected to rely on.

For all these reasons I think that this subsection of UAX #14 is in need of yet more corrections, as well as of the suggested additional information, so as to enable better understanding of the facts, but also in order to provide useful recommendations for all foundries to align on the single best—and definitely the only useful—design choice as found in Arial and Times New Roman:

U+202F → <noBreak> U+2009

1.2 GL: Non-breaking ("Glue") [continued]

Change from:

2007	FIGURE SPACE

This is the preferred space to use in numbers. It has the same width as a digit and keeps the number together for the purpose of line breaking.

Change to:

2007	FIGURE SPACE

This ~~is the preferred~~ space ~~to~~ is used ~~in~~ to indent numbers as a way of vertically aligning decimal separators. It has the same width as a digit and ~~keeps the number together for the purpose of line breaking~~ is thus too wide as a group separator. See NARROW NO-BREAK SPACE.

Rationale:

The survival in UAX #14 of that wrong recommendation is due to a failure to keep this annex in synch with the Core Specification, that had stopped since TUS 3.0 "provid[ing]" FIGURE SPACE "as a thousands separator" or informing that it "is intended to be used" as such. Early, TUS itself started mentioning its width—which makes it unfit for use as a numeric group separator—and made thus the concomitant statement problematic. How could Unicode "provide" a digit-wide space "to use in numbers" while there are plenty of alternatives one or two scalar value increments or decrements away? Picking THIN SPACE instead (2007_{16} + 2 = 2009_{16}) was so easy and straightforward it could scarcely remain unconsidered.

Details:

What UAX #14 recommends persistently—across all of its versions—as of the use of FIGURE SPACE is exactly what the Core Specification told users of Unicode 1.0 and Unicode 2.0. "The figure space is provided for use in some languages as a thousands separator." That was the clear and lapidar statement on page 75 of TUS in 1991 (version 1.0, subsection 3.2 Symbols, General Punctuation). Five years later the information was basically the same, in a more elaborate form with some additions: "U+2007 FIGURE SPACE is intended to be used as a thousands separator in cases where countries use space to separate groups of digits. Typically it has a fixed width the same size as a digit in a particular font. U+2007 FIGURE SPACE behaves like a numeric separator for the purposes of bidirectional layout.” (TUS version 2.0, subsection 6.2 Symbols Area, General Punctuation, Space Characters, Typographical Space Characters, page 6-68). Consistently it was the only non-breaking space in that range as far as we can tell based on reconstructed 1.0 UnicodeData.txt. It still is.

To sum it up: The false recommendation originates from an era when the line breaking behavior of a number of spaces was not yet determined. A quotation from TUS 2.0 is discussed below. In that era, THIN SPACE was perhaps not yet defined as breaking. Typographers were reportedly part of the early Unicode team. Nevertheless, THIN SPACE was disregarded as a group separator, and the widely unrelated FIGURE SPACE was promoted instead. Making sense of that discovery is fairly hard, but it is part of understanding why the Unicode Standard fell short of providing useful guidance and property value support as of fixed-width no-break spaces, whose usefulness Asmus Freytag questioned incidentally (at the end of this contribution to a recent thread [5] on the Unicode Public Mailing List), and how that very early bias managed to stay alive here and there in the Standard.

UAX #14 was drafted while Unicode 2.0 was still valid (it was so until 1999), but any influence of TUS on UAX #14 is highly improbable for two reasons:

1. Unlike TUS making statements about Unicode’s intention (“provided for use”, “intended to be used”), UAX #14 states about (user?) preference, already in the first accessible draft from 1998: “This is the preferred space to use in numbers. It has the same width as a digit […].”
2. Unlike TUS changing its text as soon as NNBSP came in, by deleting any mention of a thousands separator (Unicode 3.0, chapter 6 Punctuation, pages 149–150, quoted in *Proposal to synchronize the Core Specification*), UAX #14 never updated its stance.

Unicode 1.0 had full liberty to pick the right space as a group separator, since TUS 1.0 itself warns:

> ➥ It is important to note that not all space characters have word- or line-breaking properties.

As of decomposition mappings containing information about line breaking behavior, Reconstructed UnicodeData.txt 1.0 admittedly reflects the state of the data from a time when the warning about undefined line breaking property values was going to disappear from TUS. It has been deleted for Unicode 2.0, the version that mapped the aborted quads to the respective spaces (first in UnicodeData 2.0.14; see the XCCS table rows quoted on page 2 of L2/19-115 about why there was no duplicate encoding). When looking at the facts, I cannot help thinking that if TUS alleged that FIGURE SPACE was the group separator, it did so only until the bias altering the line breaking property values of the space range U+2000..U+200A became irreversible, then suddenly both stopped supporting any group separator *and* refrained from disclosing what FIGURE SPACE is actually used for, letting the reader guess.

As a reminder: FIGURE SPACE got its name not because it should be used *in* figures, but because it has the width of a figure character (this information is in TUS) and is intended to be used *before* figures, for the purpose of indenting a number until its decimal separator aligns with those above. This information is neither in UAX #14 nor in TUS, that however mentions the “*tabular width.*” It was used that way in old-fashioned typesetting, hence its non-Unicode alias “tabular space” (currently ESPACE TABULAIRE in French), which is in another way confusing, given that there are actually *two* tabular spaces (PUNCTUATION SPACE is the other one). Both may still be used so in plain text; they were designed for proportional fonts, where they make actual sense (example on page 4 of L2/19-115).

A full quote of the relevant part of TUS is in the *Background* section of *Proposal to synchronize the Core Specification.* It shows that in TUS 3.0, Unicode stopped supporting the locale preference for any space as a group separator. It was well understood that (quoting again L2/98-268R cited in L2/19-112) “[t]he concept of the Mongolian space (a non-breaking space, narrower than a normal non-breaking space, and contrasting with it in usage) could be of use in other scripts as well; therefore it is better to make this a general use punctuation character, rather than limiting it to the Mongolian script.”

Prior to being introduced in section 6.2 of TUS 6.1 (2012), NNBSP probably needed to be supported in fonts outside Mongolian. Conversely, by the means of sane recommendations, perhaps TUS could have participated in making font support for NNBSP skyrocket. That certainly holds true for UAX #14 as well. Now that NNBSP is supported virtually everywhere it is used, I suggest that the Unicode Standard—from an overall point of view—be thoroughly updated.

Still, UAX #14 alone keeps carrying the obsolete recommendation along. About not using FIGURE SPACE in numbers but a non-breaking THIN SPACE, please see L2/19-112 *Proposal to define a space character as a group separator*. I think that the Unicode Standard is the correct place for sorting out which space is preferred in virtually any locale using a space rather than a punctuation mark for grouping digits. There is actually as little choice as where the group separator is FULL STOP, especially because NO-BREAK SPACE as a group separator is an inappropriate pre-Unicode fallback, that cannot be used this way unless line justification is turned off. NBSP is a typographically non-standard, outdated legacy representation of the group separator space. FIGURE SPACE is even worse in this regard, as not only it is typographically non-standard, but it has not even the status of a legacy fallback.

1.3 IS: Infix Numeric Separator

Change from:

Note: FIGURE SPACE, not being a punctuation mark, has been given the line break class **GL.**

Change to:

Note: FIGURE SPACE~~,~~ and PUNCTUATION SPACE are used in front of numbers as a way of vertically aligning decimal separators. Not being ~~a punctuation mark, has~~ intended for use as infix numeric separators, they have been given ~~the~~ other line breaking classes ~~**GL**~~**.**

Rationale:

In its actual state, this note assumes that FIGURE SPACE is the group separator space, and has thus a functionality of an infix numeric separator. This note could be simply deleted, but since it is here, this is another opportunity to spread the word about the true nature of these spaces and their actual usage. I see a need to put FIGURE SPACE and PUNCTUATION SPACE into perspective. They are used alongside each other and therefore should never have got dissimilar line breaking property values. The suggested rewording palliates this error by not citing the individual classes.

1.4 BA: Break After

Change from:

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. See also the **SP** property**.**

Change to:

These characters behave as breaking spaces, but some of them such as THIN SPACE, EN SPACE, EM SPACE (not EN QUAD, EM QUAD) are currently tailored to meet user expectations by non-breaking behavior in some environments.

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. See also the **SP** ~~property~~ class.

Rationale:

Although tailoring is dealt with in *Section 8: Customization*, it is key information here with respect to the use of THIN SPACE on the internet under the assumption that user agent rendering engines would keep its line breaking behavior as tailored in publishing, or after inappropriately exporting text from these environments, not noticing that the non-breaking behavior THIN SPACE is merely due to tailoring and won't subsist in a standard environment. Tailoring THIN SPACE as non-breaking, like recommended in L2/19-286, as a way of getting a non-breaking thin space has grown so important people are fooled into using &thinsp; on the internet, expecting it to be non-breaking by default. Reportedly, THIN SPACE was known for its non-breaking behavior until Unicode came on it.

I think that due to its importance, tailoring needs to be mentioned where appropriate, also outside its dedicated section in UAX #14. Here seems to be a good place for mentioning it amidst the core matter.

Hence the suggested fix is to add an extra paragraph about line breaking, replacing the lone phrase ("but otherwise behave as breaking spaces") buried among unrelated content.

When enumerating examples, I'd suggest adding EN SPACE, EM SPACE, EN QUAD and EM QUAD to THIN SPACE in an attempt to draw public attention on the fact that tailoring is agnostic of canonical equivalence here, given that EN QUAD and EM QUAD are kept breaking even where the other spaces of the range are non-breaking, as in TeX.

Conclusion: Given that the line breaking behavior of these spaces is non-obvious and raises much concern, as exposed in L2/19-115 *Proposal to ensure usability of fixed-width spaces*, some additional information about line breaking behavior seems to me mandatory here.

2. Collected changes

Some of the edit suggestions collected in section 3 of {Formal} exceeded the scope of purely formal edits. As a consequence, they needed to be moved here.

In turn, some of these exceeded the scope of basic material fixes as originally defined for L2/19-317 ***Proposal to update some statements about space characters in*** **Unicode Standard Annex #14: Unicode Line Breaking Algorithm.** That scope is still valid for this proposal. Therefore I needed to add yet another proposal—please see *Proposal to reengineer spaces and punctuation in UAX #14*—that shall contain suggestions pertaining to design decisions made for the Unicode Line Breaking Algorithm.

2.1 2 Definitions

LD6. Line Breaking Class: A class of characters with the same line breaking property value.

➔ Since the term “line breaking class” is also used as a synonym of “line breaking property value”, adding this information would probably be helpful here:

[…] Also used as a synonym of “line breaking property value.”

2.2 2 Definitions

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

➔ This is the first definition where UAX #14 stops defining terms, and starts creating confusion instead.

➔ That is a rule rather than a definition. The missing definition would be something like: “A position in the text where a line is broken following a character that has the mandatory break property value.” As it stands, this pseudo-definition is using the defined term without formally defining it in the first place, and creates confusion by:

1. insinuating that unlike what is defined in LD2, the “[line] break” here does not explicitly refer to a “position in the text”—supposing that this is implied, I’d suggest making it clear—
2. improperly using the word “property”—please see {Formal}, item 3.11.

➔ Providing some information about the line breaking class BK, that “Mandatory Break” is the descriptive name of, would probably be useful here.

➔ The word “break” in the defined term is lowercase according to {Formal}, item 3.5:

LD7. Mandatory ~~B~~break: A ~~line must break~~ position in the text where a line is broken following a character that has the mandatory break property value. Also the descriptive name of the line breaking class BK, Mandatory Break.

2.3 2 Definitions

LD8. Direct Break: A line break opportunity exists between two adjacent characters of the given line breaking classes.

➔ Here again, UAX #14 drops the distinction between a *line break* and a *line break opportunity,* carved out a few lines above in LD2 and LD3. That distinction is irrelevant in LD7 where the break must always happen, but LD8 through LD10 need not only to mention the word “opportunity”, but to address the difference for clarity. I’d suggest collapsing the pairs of definitions so as to not change the definition identifiers.

➔ The word “break” in the defined term is lowercase according to {Formal}, item 3.5:

LD8. Direct ~~B~~break or ***Direct break opportunity:*** A line break or its opportunity that exists between two adjacent characters of the given line breaking classes.

2.4 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.

➔ Same material change; “SPACE” after {Formal}, item 3.14; “break” after {Formal}, item 3.5:

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

2.5 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.

➔ The rule should in my opinion be turned into a definition, here too:

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

2.6 3 Introduction

In languages, such as German, where intercharacter space is commonly used to mark e m p h a s i s (like this), allowing variable intercharacter spacing would have the unintended effect of adding random emphasis, and is therefore best avoided.

➔ For readability, the German rule of doubling spaces in spaced-out text may be applied around the example, by replacing the spaces after and especially before, with <NBSP, SPACE>.

➔ Actually, when text is spaced out, each character, even space, is followed by a space. Where usually one space is typed, three spaces are typed in a row.

➔ Additionally, character-based intercharacter spacing like in this emulation is preferably fixed-width (NNBSP; *for layout reasons, applied in the original quotation*), not justifying (NBSP; however, UAXes do not use line justification. *UAX quotations in these proposals have justification turned off accordingly*).

➔ Applying these suggestions will prevent this example from seeming like German text may have readability issues.

2.7 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.

➔ OGHAM SPACE MARK has the exact behavior of SPACE, Michael Everson explained in L2/07-392.

➔ The continuation of the stemline—if there is any in the Ogham font used—does not make this space overly stand out, as far as SPACE and all space characters in the table are “rendered visibly between words” even if without any pixels turned on.

➔ That sounds like a weak recommendation, while it is actually mandatory:

The OGHAM SPACE MARK ~~may be rendered visibly~~ continues the stemline between words or is blank in stemless fonts, ~~but~~ and it ~~is recommended~~ behaves like SPACE in that it ~~be~~ is elided at the end of a line.

2.8 BA: Break After (A)

Breaking Spaces

For a list of all space characters in the Unicode Standard, see *Section 6.2, General Punctuation*, in [Unicode].

➔ More courteous would be directing the reader to the precise location, rather than pointing to the subsection only. The URL of the table with its anchor may also be provided for easy reference:

[…] see *Table 6-2. Unicode Space Characters*, of *Section 6.2*, *General Punctuation*, in [Unicode].

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

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.

➔ This sentence has multiple issues. For moving it while making it an extra paragraph, and for some minor edits, please see the editorial suggestions in {Formal}, item 3.49.

➔ I fail to see the point in using a pair of EM DASHes instead of “a *single* quotation dash,” since HORIZONTAL BAR is already a single quotation dash, and a single instance of EM DASH can be used instead. What this is actually about is probably in getting a *long* quotation dash.

➔ This sentence is talking about making an overlong *quotation* dash out of two dashes that must not be broken at *line end*, whereas quotation dashes per definition occur at *line start;* so to talk about a quotation dash is probably to miss the point anyway.

➔ The use “in pairs” doesn’t catch it all, as shown by the THREE EM DASH used as a repetition mark in bibliographies per the *Chicago Manual of Style*, Karl Pentzlin reported in L2/10-037. “In pairs” is influenced by the algorithm checking for pairs; in the documentation, “in a row” seems more comprehensive.

“Because EM DASHes are sometimes used in ~~pairs~~ a row ~~instead of~~ to emulate a ~~single quotation~~ long dash, […]”

2.10 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).

➔ I don’t think that this recommendation actually reaches its goal, as users both expect a certain line breaking behavior *and* being able to use the base character of their choice. Rather than prompting users to restrict their choice, this section may inform about the hazard of unexpected line breaking behavior involved by choosing certain base characters, and advise to prefer precomposed symbols over sequences involving enclosing combining marks.

➔ Please see also {Formal}, item 3.52, for a minor edit to the parenthetical.

[…] ~~To avoid s~~Surprising behavior~~, always use a~~ may occur depending on the base character. Safest are ~~that is a~~ symbols ~~or~~ and letters (~~L~~line ~~B~~breaking class **AL**) when using enclosing combining marks (General_Category Me). Using a precomposed symbol instead may be the most straightforward option.

2.11 IN: Inseparable Characters (XP)

Leaders

These characters are intended to be used consecutively. There is never a line break between two characters of this class.

➔ About modifying this rule, please see also suggestion 2.29.

➔ The heading of this line breaking class should indicate the break allowed after, since the only rule about this class, LB22, only prevents a set of break opportunities before, beside prohibiting breaks between pairs as specified:

IN: Inseparable Characters (A/XP)

2.12 5.4 Use of Soft Hyphen

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

➔ German is only involved so far as pre-reform orthography is applied, exactly like in the first item of this bullet list. Swedish, being fully involved, should be cited in the first place.

➔ Rather than “doubling” a consonant, the change consists in *restoring* an elided consonant where composing a word would result in a triple consonant: German “Brenn- + Nessel” (stinging nettle) composes to “Brennessel” (pre-reform) or to “Brennnessel” (post-reform). Hyphenation always results in “Brenn- / nessel”. By contrast, Swedish “bränn- + nässla” *always* composes to “brännässla”, and hyphenates into “bränn- / nässla”.

➔ As reported in item 3.73 of {Formal}, the semicolon after the example is superfluous.

➔ Picking *tuggummi* as an example when there are plenty of other words available is inappropriate (for various reasons), at least in the Unicode Standard:

In ~~German and~~ Swedish and pre-reform German, a consonant that has been elided to avoid a triple consonant at word composition is ~~sometimes doubled~~ restored: Swedish “~~tuggummi~~brännässla”~~;~~ hyphenates into “~~tugg- / gummi~~bränn- / nässla”.

2.13 5.4 Use of Soft Hyphen

There are three types of hyphens: explicit hyphens, conditional hyphens, and dictionary-inserted hyphens resulting from a hyphenation process. There is no character code for the third kind of hyphen. If a distinction is desired, the fact that a hyphen is dictionary-inserted and not user-supplied can only be represented out of band or by using another control code instead of SHY.

➔ The construct “dictionary-inserted” does not appear to exist except in this paragraph of UAX #14. On the Unicode website, all 90 instances found by Google Search are in mostly old versions of UAX #14 and their drafts partly stored in the UTC Document Register. The good reason is that dictionaries do not insert anything. Renderers do, after dictionary lookup. Just “hyphens resulting from a hyphenation process” is fine. If that is too long, one may get away with using the antonym of “explicit”.

➔ Since SHY is a format character, not a control code, the control code used instead would not be “another” one.

➔ Elsewhere in the document, actually as soon as in the next paragraph, a conditional hyphen is called an "explicit SHY", crossing the attribute "explicit" applied to always visible hyphens. That should be taken into account when enumerating the three types of hyphens:

There are three types of hyphens: explicit hyphens, conditional hyphens—also called explicit SOFT HYPHENs—, and ~~dictionary-inserted~~ implicit hyphens resulting from a hyphenation process. There is no character code for the third kind of hyphen. If a distinction is desired, the fact that a hyphen is ~~dictionary-inserted and~~ not user-supplied but results from hyphenation can only be represented out of band or by using ~~another~~ a control code instead of SHY.

2.14 5.5 Use of Double Hyphen

In this example, if the shape of the special hyphen matches an existing character, such as U+2E17 DOUBLE OBLIQUE HYPHEN, that character can be substituted temporarily for display purposes by the line formatter.

➔ Since the "shape" is related to *glyphs*, it can only "match" the average *glyph* of a character, not the "character" itself.

➔ The idea that a character is substituted to another character "for display purposes", rather than a glyph to another glyph, is current in specifications if extrapolating from this and another example, quoted in L2/18-026 *Proposals to ensure legibility of bidirectional mathematical notation*.

➔ When getting the terminology right, the phrase "temporarily for display purposes" becomes superfluous:

In this example, if the shape of the special hyphen matches the glyph of an existing character, such as U+2E17 DOUBLE OBLIQUE HYPHEN, the glyph of that character can be substituted ~~temporarily for display purposes~~ by the line formatter.

2.15 5.5 Use of Double Hyphen

Certain linguistic notations make use of a double-stroke, oblique hyphen to indicate specific features. The U+2E17 DOUBLE OBLIQUE HYPHEN character used in this case is not a hyphen and does not represent a line break opportunity. Automatic hyphenation or SHY would result in the display of an ordinary hyphen.

➔ The U+2E17 DOUBLE OBLIQUE HYPHEN character was "not a hyphen" only from version 4.1.0 (2005) to version 5.0.0 (2006), but since then it has the Hyphen property. As of providing a line break opportunity, it was assigned the line breaking class BA (Break After) since it was encoded for Unicode 4.1.0, and did never change line breaking class since then.

➔ The display that implicit or explicit SHY would result in, depends on the language and the notation, per section 5.4 reading: "The inserted hyphen glyph can take a wide variety of shapes, as appropriate for the situation. Examples include shapes like U+2010 HYPHEN, U+058A ARMENIAN HYPHEN, U+180A MONGOLIAN NIRUGU, or U+1806 MONGOLIAN TODO SOFT HYPHEN."

➔ If this paragraph does not become pointless when each statement turns into its opposite, it must be changed. For clarity that is the most desirable option. Suggested:

Certain linguistic notations make use of a double-stroke, oblique hyphen to indicate specific features. The U+2E17 DOUBLE OBLIQUE HYPHEN character used in this case is ~~not~~ a hyphen and ~~does not~~ represents a line break opportunity. Automatic hyphenation or SHY ~~would~~ should result in the display of ~~an ordinary hyphen~~ whatever glyph is appropriate for the language, the notational system or the situation.

2.16 5.7 Word Separator Characters

For case (1), the line break opportunity is positioned after the word separator character, as in case (3), but the visual display of the character is suppressed. The means by which a line layout and display process inhibits the visible display of the separator character are outside of the scope of the Line Break algorithm. U+1680 OGHAM SPACE MARK is an example of a character which may exhibit this behavior.

➔ Despite this subsection was added for version Unicode 5.1 (2008, see in revision 21), the last sentence tends to hint that Michael Everson’s feedback L2/07-392 was not fully implemented:

> “However, an Ogham font may also have **no** stemline. […] U+1680 would behave exactly as U+0020 in searching Latin-script text, and indeed in UnicodeData.txt they have identical properties.”

➔ The OGHAM SPACE MARK does act like SPACE, not like a word separator character with a non-blank graphic glyph. As a consequence, it should be assigned class SP. The last quoted sentence would then become redundant with the SP class description, that should be cited as an example instead.

➔ I think that “positioned” is improper instead of “located”:

For case (1), the line break opportunity is ~~positioned~~ located after the word separator character, as in case (3), but the visual display of the character is suppressed. The means by which a line layout and display process inhibits the visible display of the separator character are outside of the scope of the Line Break algorithm. ~~U+1680 OGHAM SPACE MARK is an~~ For example, class SP ~~of a character which may~~ exhibits this behavior.

2.17 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

➔ As mentioned in 2.14, this rule (the only one about IN) allows breaks after ellipses, whereas ellipses are also used in leading position (see *Wikipedia*), before the start of a word or a phrase or sentence, without SPACE after, but with SPACE before. Actually, the untailored Unicode Line Breaking Algorithm allows the renderer to break the line after an ellipis even if that ellipsis is preceded by SP and followed by AL. For a leading ellipsis to stay in its place, the user needs to insert WJ after the ellipsis.

➔ This rule would work if the regexes matched the rule title. The title is correct, but the regexes are not. To fix the rule, I suggest making its regexes symmetric except for class EX:

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

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

2.18 6.2 Tailorable Line Breaking Rules

LB30 *Do not break between letters, numbers, or ordinary symbols and opening or closing parentheses.*

(AL | HL | NU) × OP
CP × (AL | HL | NU)

The purpose of this rule is to prevent breaks in common cases where a part of a word appears between delimiters—for example, in “person(s)”.

➔ The preposition “between” does logically not match the regexes, that looking at is required in order to understand that there is an implied “adjoining to the outside of”. However, this information does easily integrate into the rule title.

➔ Given that class OP represents much more punctuation than class CP, the asymmetricity of these regexes should probably be fixed, so that the second one covers also braces and so on. I don’t see the point in restraining the user as of what paired—or at least, closing—punctuation can be used without breaking the layout. Hence, in a next step, I’d suggest completing this rule that way:

LB30 *Do not break between letters, numbers, or ordinary symbols and the outside of adjacent opening or closing ~~parentheses~~ punctuation.*

(AL | HL | NU) × OP
(CL | CP) × (AL | HL | NU)

Background

This proposal encompasses **the problem of the preferred space character to use in numbers as a group separator.** I think of related inconsistencies in CLDR mainly as a by-product of the Unicode Standard in general, and UAX #14 in particular, having a hard time designing and specifying an interoperable encoding scheme of a set of space characters, and that the environment best fit for sorting out which space to use as a group separator is the Standard managing the underlying code space, rather than CLDR, since in my opinion, locale preference is about choosing the group separator on a basic level, whether it is a comma, a period, a space, the Arabic thousands separator, and so on among non-blank graphics. Determining the exact space character to use would be a matter of Unicode education and implementation maturity in a context of time passing, and fonts being updated, after the Unicode Standard had become supportive of many more locales by encoding U+202F NARROW NO-BREAK SPACE.

The trouble with using a non-breaking thin space is reported by Patrick Andries [2]. The use of a no-break thin space is not (and was never intended to be) limited to the graphic industry. As the no-break thin space is a part of the interoperable representation of numerous languages including Armenian, English, French, Georgian, German and Tuareg using Tifinagh script, it is expected to be available out-of-the-box. One example are French keyboard layouts generating input alternatively without or with automated or manual punctuation spacing as exposed in CLDR ticket #10904, and with a facility to type numbers with locale-conformant group separators (NARROW NO-BREAK SPACE, FULL STOP or COMMA). Resulting data is interoperable, that is, usable unchanged in plain text, word processing, DTP, email, and on the web.

The lack of *NO-BREAK THIN SPACE, or rather, the wrong line breaking class assigned to THIN SPACE in Unicode was a main disturbance impacting keyboard layout development and deployment due to discrepancies between resources and requirements on one side, encoding and support on the other side.

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

Special thanks to Deborah Anderson, Ken Whistler, Roozbeh Pournader, Lisa Moore, and Liang Hai from the Script Ad Hoc group, to the Mongolian Working Group, and to the Unicode Technical Committee, for helping fix the non-breaking thin space issue.

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.