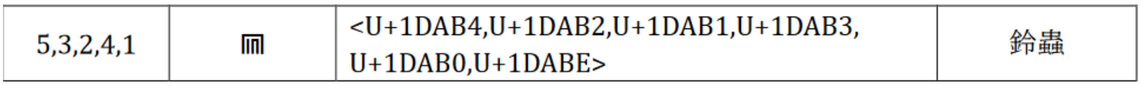
L2/22-156

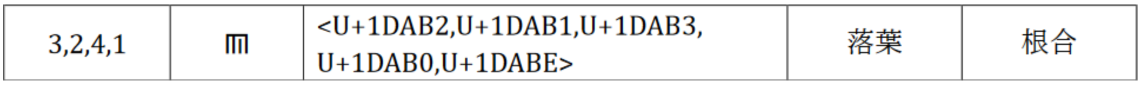
Title: Feedback on L2/22-117 (Proposal to encode Kōdō Incense Linear Patterns)
Author: Wang Yifan
Date: 2022-07-08

Substring match possibility

While the proposed algorithm of injective mapping from partitions of set to digit sequences whose length is same as the set's size is concise and well thought out, the encoding design has an undesired substring match problem in its original form. We can easily pick up a pair from the Section 1.4 such as:



5,3,2,4,1		<U+1DAB4,U+1DAB2,U+1DAB1,U+1DAB3, U+1DAB0,U+1DABE>	鈴蟲



3,2,4,1		<U+1DAB2,U+1DAB1,U+1DAB3, U+1DAB0,U+1DABE>	落葉	根合

where the latter would match the former with a simple code point-based search:

```
<U+1DAB4 U+1DAB2 U+1DAB1 U+1DAB3 U+1DAB0 U+1DABE>
         <U+1DAB2 U+1DAB1 U+1DAB3 U+1DAB0 U+1DABE>
```

although the corresponding patterns are not similar.

Due to the positional nature of the core algorithm, the semantics of each digit in the representation is determined by its position from the starting digit (leftmost in the numerical notation). It means forward matching using this notation system certainly has some naturalness (since it entails the same graphic subpattern), but backward matching is pointless. The current scheme with a terminator, which prevents forward substring match and condones backward, is thus less optimal.

Moreover, given that each digit always represents a context-dependent, relative function in a compositional semantics, partial match of different-length sequences itself is conceptually nonideal. It can also complicate the font implementation (e.g., glyph for a shorter partial sequence can show up unexpectedly when a longer sequence is unsupported). Thus with above said, I suggest to introduce a leading character in tandem with the existing END CONTROL to demarcate both the start and the end, similar to the design of emoji tag sequence in UTS #51.

Repertoire

The original proposal assumes that all notable incense patterns are encoded in the form of named sequences. I however see the usefulness of assigning code points to all of now-practiced 3, 4, 5-bar patterns. A prominent feature of incense patterns is that, unlike most other CJK symbols, they are traditionally sorted by a fixed order with non-simple rules[1] (as described in Section 1.4.1 through

[1] There has been some research on the principle of *genjikō* ordering, and its nature is believed to be an

1.4.3). If patterns are completely defined by underlying sequences, it would be difficult to reproduce pattern-oriented ordering independent of the arrangement of code points constituting the sequence within the mechanism the Unicode Standard currently supports out of box, i.e., without tailoring. Taking up code points allows for the possibility that either of code chart sorting or collation sorting can align with the traditional ordering, which contributes to ease of use in the real world.

Allocating a code point for each named pattern fortunately does not need much space, as their amount is that of smaller elements of Bell numbers. Registration of 3, 4, 5-bar patterns only costs 5 + 15 + 52 = 72 code points. We can establish canonical equivalence between the standalone patterns and sequence representations for normalization.

Pattern names

The original proposal names known patterns with chain of digits generated by the algorithm explained in the Section 1.3. This practice would however hinder the recognition of each combinatory symbols by users, who are accustomed to associate each gestalt with traditional names in its primary use case. Each pattern should be instead named after its traditional label (those which have multiple names need further consideration).

(End of document)

amalgamation of visual symmetricity and arbitrariness. See e.g.: Moriguchi, Shigeichi (1998). “Genjikō-zu no nazo” 源氏香図の謎. *UP* 27 (7), 12–18.