

**Universal Multiple-Octet Coded Character Set
International Organization for Standardization**

Doc Type: ISO/IEC JTC1/SC2/WG2/IRG

Title: Proposal to encode five new Ideographic Description Characters

Authors: Ken Lunde, John Jenkins & Andrew West

Status: Member Body Contribution

Action: For consideration by the IRG and UTC

Date: 2022-08-24

To follow up on [L2/21-118R](#) (aka [IRG N2492](#)) and UTC #172 Action Item 172-A52, this document is a proposal to encode five (5) new Ideographic Description Characters (IDCs) in order to handle a modest number of edge cases when managing Ideographic Description Sequences (IDSes) and IDS databases. IDCs and IDSes are extensively documented in [Section 18.2, Ideographic Description Characters](#), of the Core Specification of the Unicode Standard.

Five New Ideographic Description Characters


Four new IDCs were most recently proposed in [L2/18-012](#) (aka [IRG N2273](#)) as shown in the first four rows of the table below (the representative glyph of the fourth one was adjusted per UTC feedback), along with a fifth one that was introduced in [L2/21-118R](#):

IDC	Type	Character Name
	Binary	IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM RIGHT
	Binary	IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM LOWER RIGHT
	Unary	IDEOGRAPHIC DESCRIPTION CHARACTER HORIZONTAL REFLECTION
	Unary	IDEOGRAPHIC DESCRIPTION CHARACTER HALF-TURN ROTATION
	Binary	IDEOGRAPHIC DESCRIPTION CHARACTER COMPONENT SUBTRACTION

The first two proposed new IDCs— IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM RIGHT and IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM LOWER RIGHT—follow the pattern of similar IDCs that involve an ideograph component partially surrounding another ideograph component. Other than the possible use cases being relatively low compared to the similar IDCs, these two proposed new IDCs are not expected to be problematic nor controversial.





The third and fourth proposed new IDCs— IDEOGRAPHIC DESCRIPTION CHARACTER HORIZONTAL REFLECTION and IDEOGRAPHIC DESCRIPTION CHARACTER HALF-TURN

ROTATION—are novel in that they would become the very first *unary* IDCs. They indicate the reflection or rotation of the ideograph component that follows.

The fifth proposed new IDC— IDEOGRAPHIC DESCRIPTION CHARACTER COMPONENT SUBTRACTION—is also novel in that it specifies an ideograph component that is removed. It is a binary IDC and is therefore followed by two components:











1. An ideograph component
2. An ideograph component, such as stroke from the [CJK Strokes](#) block, that is omitted from the first ideograph component

Below are examples of this IDC used in IDSes:

- The IDSes for U+2002A 其 and U+2002B 其 are difficult to represent with existing ideograph components, but could be easily represented as 其ノ and 其ノ, respectively.
- The IDS for U+2CEBB 豕 is also difficult to represent with existing ideograph components, but could be represented as 豕ノ.
- The IDS for U+27C27 豕 is also difficult to represent with existing ideograph components, but could be represented as 豕ノ.




A counter-example for the first example above would be to instead encode the common ideograph component of U+5176 其, U+2002A 其, and U+2002B 其 as a new ideograph component, but that accommodates only this particular case. Encoding a new IDC is much more productive.

The following table provides examples of how each of these IDCs would be used to represent existing ideographs in IDSes:

IDC	Ideograph	IDS
	U+355A 叉	 叉ノ
	U+6C37 水	 水ノ
	U+23944 五	 正
	U+20114 予	 予
	U+2002A 其	 其ノ

In terms of existing IDS implementations that use one or more of the proposed new IDCs, the [IDS.TXT](#) IDS database currently uses U+2194 ↔ LEFT RIGHT ARROW, U+21B7 ↻ CLOCKWISE TOP SEMICIRCLE ARROW, and U+2296 ⊖ CIRCLED MINUS as placeholder IDCs for the last three IDCs that are proposed in this document.

Ambiguity & Other Concerns

The two proposed new unary IDCs resolve as no-ops if used in sequence. For example, 正 corresponds to 五, but 正 corresponds to 正 itself, which is a no-op. The same is true of

𠄎𠄎𠄎, which corresponds to 𠄎 itself. In addition, reflected or rotated components can be used as ideograph components as a way to represent their non-reflected or non-rotated counterparts, such as 𠄎𠄎 and 𠄎𠄎 to represent 正 and 𠄎, respectively.

There is also inherit ambiguity in the proposed new IDC, 𠄎 IDEOGRAPHIC DESCRIPTION CHARACTER COMPONENT SUBTRACTION, about which some experts may have concerns for introducing a new dimension of adverse effects on automatic matching algorithms. For example, there are three instances of the 丿 stroke in the ideograph U+27C7 豕, and it is ambiguous as to which instance is removed. The way in which IDCs are currently used, which requires a non-zero amount of human intervention for interpretation, strongly suggests that this will not be issues in practical usage. Besides, an existing IDC, U+2FFB 𠄎 IDEOGRAPHIC DESCRIPTION CHARACTER OVERLAID, is already ambiguous in that human intervention is required to determine the nature of the overlaid ideograph components.

In other words, one or more of the new proposed IDCs, in particular 𠄎 IDEOGRAPHIC DESCRIPTION CHARACTER HORIZONTAL REFLECTION, 𠄎 IDEOGRAPHIC DESCRIPTION CHARACTER HALF-TURN ROTATION, and 𠄎 IDEOGRAPHIC DESCRIPTION CHARACTER COMPONENT SUBTRACTION, are likely to be considered problematic by some experts, but like other characters in the Unicode Standard, they can be ignored by those who find them to be problematic. For example, if one or more of these new IDCs pose problems for the IRG (*Ideographic Research Group*), such as when performing IDS matching against IRG submission data, the IRG could simply mandate in its P&P (*Principles & Procedures*) that particular IDCs cannot be used in IDSes for IRG submissions. IDS database maintainers do not necessarily have such constraints.

Proposed Code Points, Character Names & Properties

The [Ideographic Description Characters](#) block, which is the most appropriate block for encoding these five new IDCs, has exactly four available code points: **U+2FFC through U+2FFF**. We recommend encoding the first four of these new IDCs using these particular code points. It was suggested during the UTC #172 meeting that **U+31EF**, which is at the very end of the [CJK Strokes](#) block, be recommended as the code point for the fifth IDS.

Therefore, the following are the proposed code points, character names, and property values for the five proposed new IDCs:

```
2FFC;IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM RIGHT;So;0;ON;;;;;N;;;;;
2FFD;IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM LOWER RIGHT;So;0;ON;;;;;N;;;;;
2FFE;IDEOGRAPHIC DESCRIPTION CHARACTER HORIZONTAL REFLECTION;So;0;ON;;;;;N;;;;;
2FFF;IDEOGRAPHIC DESCRIPTION CHARACTER HALF-TURN ROTATION;So;0;ON;;;;;N;;;;;
31EF;IDEOGRAPHIC DESCRIPTION CHARACTER COMPONENT SUBTRACTION;So;0;ON;;;;;N;;;;;
```

New Character Property

The two new unary IDCs will require that a new character property, **IDS_Unary_Operator**, be defined. This new property needs to be reflected in the “CJK” section of Table 7, *Property Index by Scope of Use*, in [Section 5.1, Property Index](#), of UAX #44 as a link to a new entry in the “PropList.txt” section of Table 9, *Property Table*, in [Section 5.3, Property Definitions](#), of the same UAX with Property Type, Property Status, and Property Description fields being identical to those of *IDS_Binary_Operator* and *IDS_Tertiary_Operator*:

Property Type: **B**
Property Status: **N**
Property Description: **Used in Ideographic Description Sequences.**

The following are the proposed changes to the IDC-related lines in the UCD’s *PropList.txt* file, showing changes and new lines in **red**:

```
2FFE..2FFF ; IDS_Unary_Operator # So [2] IDEOGRAPHIC DESCRIPTION CHARACTER
HORIZONTAL REFLECTION..IDEOGRAPHIC DESCRIPTION CHARACTER HALF-TURN ROTATION

2FF0..2FF1 ; IDS_Binary_Operator # So [2] IDEOGRAPHIC DESCRIPTION CHARACTER LEFT
TO RIGHT..IDEOGRAPHIC DESCRIPTION CHARACTER ABOVE TO BELOW
2FF4..2FFD ; IDS_Binary_Operator # So [10] IDEOGRAPHIC DESCRIPTION CHARACTER FULL
SURROUND..IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM LOWER RIGHT
31EF ; IDS_Binary_Operator # So IDEOGRAPHIC DESCRIPTION CHARACTER
COMPONENT SUBTRACTION

2FF2..2FF3 ; IDS_Tertiary_Operator # So [2] IDEOGRAPHIC DESCRIPTION CHARACTER LEFT
TO MIDDLE AND RIGHT..IDEOGRAPHIC DESCRIPTION CHARACTER ABOVE TO MIDDLE AND BELOW
```

The proposed short name for the *IDS_Unary_Operator* property is **IDSU**, and following is the proposed change to the IDC-related lines in the “Binary Properties” section of the UCD’s *PropertyAliases.txt* file, showing new lines in **red**:

```
IDSU ; IDS_Unary_Operator
IDSB ; IDS_Binary_Operator
IDST ; IDS_Tertiary_Operator
```

IDS Grammar

The grammar in Section 18.2, *Ideographic Description Characters*, of the Core Specification should be updated to accommodate unary IDCs and the three new binary IDCs, as follows (additions are shown in **red**):

```
IDS := Ideographic | Radical | CJK_Stroke | Private Use | U+FF1F
| IDS_UnaryOperator IDS
| IDS_BinaryOperator IDS IDS
| IDS_TertiaryOperator IDS IDS IDS
CJK_Stroke := U+31C0 | ... | U+31E3
IDS_UnaryOperator := U+2FFE | U+2FFF
IDS_BinaryOperator := U+2FF0 | U+2FF1 | U+2FF4 | ... | U+2FFD | U+31EF
IDS_TertiaryOperator:= U+2FF2 | U+2FF3
```

TrueType Font

A TrueType font with an open source (OFL) license that provides representative glyphs for all 17 IDCs—12 existing plus five proposed—that map from code points in the Ideographic Description Characters (U+2FF0 through U+2FFF) and CJK Strokes (U+31EF) blocks is attached to this PDF.

That is all.

**ISO/IEC JTC 1/SC 2/WG 2
PROPOSAL SUMMARY FORM TO ACCOMPANY SUBMISSIONS
FOR ADDITIONS TO THE REPERTOIRE OF ISO/IEC 10646¹**

Please fill all the sections A, B and C below.

Please read Principles and Procedures Document (P & P) from <http://std.dkuug.dk/JTC1/SC2/WG2/docs/principles.html> for guidelines and details before filling this form.

Please ensure you are using the latest Form from <http://std.dkuug.dk/JTC1/SC2/WG2/docs/summaryform.html>.

See also <http://std.dkuug.dk/JTC1/SC2/WG2/docs/roadmaps.html> for latest **Roadmaps**.

A. Administrative

1. **Title:**
2. Requester's name:
3. Requester type (Member body/Liaison/Individual contribution):
4. Submission date:
5. Requester's reference (if applicable):
6. Choose one of the following:
- This is a complete proposal:
- (or) More information will be provided later:

B. Technical – General

1. Choose one of the following:
- a. This proposal is for a new script (set of characters):
- Proposed name of script:
- b. The proposal is for addition of character(s) to an existing block:
- Name of the existing block:
2. Number of characters in proposal:
3. Proposed category (select one from below - see section 2.2 of P&P document):
- | | | |
|--|--|---|
| A-Contemporary <input checked="" type="checkbox"/> | B.1-Specialized (small collection) <input type="checkbox"/> | B.2-Specialized (large collection) <input type="checkbox"/> |
| C-Major extinct <input type="checkbox"/> | D-Attested extinct <input type="checkbox"/> | E-Minor extinct <input type="checkbox"/> |
| F-Archaic Hieroglyphic or Ideographic <input type="checkbox"/> | G-Obscure or questionable usage symbols <input type="checkbox"/> | |
4. Is a repertoire including character names provided?
- a. If YES, are the names in accordance with the "character naming guidelines" in Annex L of P&P document?
- b. Are the character shapes attached in a legible form suitable for review?
5. Fonts related:
- a. Who will provide the appropriate computerized font to the Project Editor of 10646 for publishing the standard?
- b. Identify the party granting a license for use of the font by the editors (include address, e-mail, ftp-site, etc.):
6. References:
- a. Are references (to other character sets, dictionaries, descriptive texts etc.) provided?
- b. Are published examples of use (such as samples from newspapers, magazines, or other sources) of proposed characters attached?
7. Special encoding issues:
- Does the proposal address other aspects of character data processing (if applicable) such as input, presentation, sorting, searching, indexing, transliteration etc. (if yes please enclose information)?

8. Additional Information:

Submitters are invited to provide any additional information about Properties of the proposed Character(s) or Script that will assist in correct understanding of and correct linguistic processing of the proposed character(s) or script. Examples of such properties are: Casing information, Numeric information, Currency information, Display behaviour information such as line breaks, widths etc., Combining behaviour, Spacing behaviour, Directional behaviour, Default Collation behaviour, relevance in Mark Up contexts, Compatibility equivalence and other Unicode normalization related information. See the Unicode standard at <http://www.unicode.org> for such information on other scripts. Also see Unicode Character Database (<http://www.unicode.org/reports/tr44/>) and associated Unicode Technical Reports for information needed for consideration by the Unicode Technical Committee for inclusion in the Unicode Standard.

¹ Form number: N4502-F (Original 1994-10-14; Revised 1995-01, 1995-04, 1996-04, 1996-08, 1999-03, 2001-05, 2001-09, 2003-11, 2005-01, 2005-09, 2005-10, 2007-03, 2008-05, 2009-11, 2011-03, 2012-01)

C. Technical - Justification

1. Has this proposal for addition of character(s) been submitted before? If YES explain	Yes See proposal
2. Has contact been made to members of the user community (for example: National Body, user groups of the script or characters, other experts, etc.)? If YES, with whom? If YES, available relevant documents:	Yes See proposal See proposal
3. Information on the user community for the proposed characters (for example: size, demographics, information technology use, or publishing use) is included? Reference:	N/A
4. The context of use for the proposed characters (type of use; common or rare) Reference:	Common See proposal
5. Are the proposed characters in current use by the user community? If YES, where? Reference:	N/A
6. After giving due considerations to the principles in the P&P document must the proposed characters be entirely in the BMP? If YES, is a rationale provided? If YES, reference:	Yes Yes See proposal
7. Should the proposed characters be kept together in a contiguous range (rather than being scattered)?	Yes
8. Can any of the proposed characters be considered a presentation form of an existing character or character sequence? If YES, is a rationale for its inclusion provided? If YES, reference:	No
9. Can any of the proposed characters be encoded using a composed character sequence of either existing characters or other proposed characters? If YES, is a rationale for its inclusion provided? If YES, reference:	No
10. Can any of the proposed character(s) be considered to be similar (in appearance or function) to, or could be confused with, an existing character? If YES, is a rationale for its inclusion provided? If YES, reference:	No
11. Does the proposal include use of combining characters and/or use of composite sequences? If YES, is a rationale for such use provided? If YES, reference: Is a list of composite sequences and their corresponding glyph images (graphic symbols) provided? If YES, reference:	No
12. Does the proposal contain characters with any special properties such as control function or similar semantics? If YES, describe in detail (include attachment if necessary)	No
13. Does the proposal contain any Ideographic compatibility characters? If YES, are the equivalent corresponding unified ideographic characters identified? If YES, reference:	No

SAT Feedback to “Preliminary proposal to add a new provisional kIDS property (Unihan)” (IRGN2492) and “Proposal to encode five new Ideographic Description Characters” (IRGN2572)

Date: 2022-08-29

1. Usage of IDEOGRAPHIC DESCRIPTION CHARACTER STROKE SUBTRACTION

We generally acknowledge the usefulness of the newly proposed binary operator IDEOGRAPHIC DESCRIPTION CHARACTER STROKE SUBTRACTION (hereinafter SS) for human users, but at the same time, we are concerned, as Dr. Qin LU in a comment about the document at IRG #57, that it would introduce to the IDS system a new dimension of ambiguity which is hostile to machine checking algorithms (such as search and generating canonical forms/decompositions). We hence suggest that **reasonable constraints should be imposed on its usage in the IDS data of new characters to be submitted to future IRG working sets.**

In the system using traditional IDCs (including SURROUND FROM RIGHT and SURROUND FROM LOWER RIGHT in this discussion), a character is described with a combination of one or more two-dimensionally separable components in the way designated by each IDC. Provided every existing CJK ideograph is associated with an IDS, we can recursively decompose a character until it ultimately reduces to a combination of a limited number of graphemes that are, for most practical purposes, atomic. Although in reality an ideograph does not always resolve to only one canonical sequence, there are only a finite number of paths that can in principle be collated¹. This is the basic principle that (we assume) most IDS machine checkers presuppose, as well as the rationale why we have been freely choosing a short, intuitive IDS from among a vast number of options to describe an ideograph. (That is, the precise choice of IDS ought not matter because IDSeS are assumed to be confluent in a system of rewrite rules.) As regards the subsequent discussion, however, we note a limitation of the traditional IDS system: namely that it does not have a notation or other mechanism to indicate a relationship between two components that are not connected in the IDS forest. One purpose of UCVs is to bridge this gap, by instructing machines on the perceived equivalence of glyphs that are not technically linked in the network, as well as kStrange, to help human users look up hard-to-reach orphaned (often atomic) components for better machine checking coverage.

SS, by its nature, can be regarded as an inverse operator for any suitable traditional IDC (ignoring the difference between CJKUI and CJK Strokes for the purpose of this discussion):

$$\text{氏} = \ominus \text{氏} \setminus \leftarrow \text{氏} = \square \text{氏} \setminus$$

¹ Of course, a number of reservations should be made, such as possible incompatible/incommensurable decompositions allowed by the operator OVERLAID.

$$\begin{aligned} \text{自} &= \ominus \text{百} - \leftarrow \text{百} = \boxminus \text{一} \text{自} \\ \text{其} &= \ominus \text{其} \setminus \leftarrow \text{其} = \boxminus \text{其} \setminus \end{aligned}$$

With the support of additive (compositional) sequences defined elsewhere, it can theoretically be incorporated into the interdependent network of IDSes. However, as stated in the proposal, the motivation for introducing the new operator is to describe an ideograph otherwise difficult to compose with additive operations, such as the third example (from the original document IRGN2492), U+2CEBB (豕 without the last two strokes). Suppose that we do not have a character 大 in the Han repertoire, but only 太 and 犬. Now two submitters try to encode this new character, one using $\ominus \text{太} \setminus$ and the other using $\ominus \text{犬} \setminus$. These two IDSes are not conflatable by an algorithm if 太 and 犬 are atomic (not decomposable), or share no already describable component (which would link them using "traditional IDC" semantics). Thus, if we assume the current IDS data, it is impossible to detect whether two different SS sequences represent an effectively identical shape, or even whether such a sequence is identical to an existing character. This potential for false negatives makes SS qualitatively more dangerous than OVERLAID, which is often termed "ambiguous" and for which a fuzzy search is likely to return false positives if a decomposition strategy is sufficiently reasonable. (False positives are relatively harmless; false negatives are not.)

We therefore believe that, although using SS has a clear advantage for human recognition, the descriptiveness of a subsequence led by it is basically equivalent to ? in automatic duplication checking. We suggest that some safety measures should be taken for the use of SS in IRG WS submissions, such as (but not restricted to) one or more of the following:

- The submitter must also provide an additive IDS of a character for which SS has been used.
- The submitter must also declare the shape which the SS sequence represents in their supplementary components list (or elsewhere).
- The submitter (or another authority in the pipeline) must confirm that the intended component described with SS has not been encoded, as a part of quality assurance.
- Restrict the choice of subtrahend to a small set of minor strokes to avoid arbitrary variation for a shape associable with multiple characters.²
- Do not use sequence as the subtrahend component of SS sequence, or find a mechanism to reduce ambiguity if multiple strokes are subtracted.³
- Do not use SS inside an SS sequence.

In addition, shapes previously described with SS should be recorded in the proposed components block or some IRG documents, so that they can be found. Here, the kStrange property comes in handy.

² E.g.: 宀 = $\ominus \text{家} \text{豕} = \ominus \text{安} \text{女} = \ominus \text{完} \text{元} = \ominus \text{客} \text{各} = \ominus \text{容} \text{谷} = \ominus \text{守} \text{寸}$

³ In the original document's example of U+2CEBB (豕 without the last two strokes) as $\ominus \text{豕} \boxminus$ \setminus , people might also consider the IDSes $\ominus \ominus \text{豕} \setminus \setminus$ and $\ominus \ominus \text{豕} \setminus \setminus$.

Finally, **one idea for handling subtractive sequences from an algorithmic perspective would be to rewrite them as additive ones** (possibly using OVERLAID as a position-agnostic addition operator) in a database-internal preprocessing step.

2. Usage of IDEOGRAPHIC DESCRIPTION CHARACTER HORIZONTAL REFLECTION and IDEOGRAPHIC DESCRIPTION CHARACTER HALF-TURN ROTATION (IDEOGRAPHIC DESCRIPTION CHARACTER ONE HUNDRED EIGHTY DEGREE ROTATION)

The potential problems of unrestricted usage of those two IDCs in the IRG work have been already covered in [L2/18-012](#) (by Taichi KAWABATA) and [Kushim JIANG's Feedback](#) to IRGN2273R, which can be summarized as:

- The possibility of multiple interpretations regarding which component is transformed in a complex component (e.g., 壯 = $\overleftrightarrow{\square} \overleftrightarrow{\square} \overleftrightarrow{\square} \overleftrightarrow{\square} \overleftrightarrow{\square} \overleftrightarrow{\square}$ 干片 (?))
- The ability to create idempotent and/or redundant notations, which can cause infinite loops in the algorithm (e.g., 字 = $\overleftrightarrow{\square} \overleftrightarrow{\square}$ 字 = $\overleftrightarrow{\square} \overleftrightarrow{\square} \overleftrightarrow{\square} \overleftrightarrow{\square}$ 字 = ...)
- (We would like to add: the ambiguity that $\overleftrightarrow{\square} \overleftrightarrow{\square} = \overleftrightarrow{\square} \overleftrightarrow{\square}$)

Thus, we suggest the following for the safe handling of those unary operators in IRG WS submissions:

- Prohibit sequences as the argument of these new unary operators; that is, allow only single characters and strokes.
- The algorithm should strip away all of the unary operators from IDSeS before matching.
- The submitter (or another authority in the pipeline) must confirm that the component intended to be described by the unary operators is not encoded, as a part of quality assurance.

Also note that Kushim JIANG questions whether those unary operators should be named with *IDEOGRAPHIC DESCRIPTION CHARACTER* where their function is closer to that of $\overleftrightarrow{\square}$ U+303E IDEOGRAPHIC VARIATION INDICATOR.

3. Name and mapping of the components block

We agree with Eiso CHAN's Feedback to IRGN2492, that:

- (a) The new ideograph components block seems better placed somewhere near the end of the SIP, where a considerable number of code points remain unassigned. The range U+2EBF0 through U+2F7FF has 3,088 code points, and the range U+2FA20 through U+2FFFD has 1,502 components available, which will be more than sufficient for prospective maximal number of components. We especially note that the range after the CJK Compatibility Ideographs Supplement is the least expected place to take in any further extension of CJKUI and might be good to accommodate a smaller block, unless other factors are considered.
- (b) The block name (with the proposed name *CJK Unified Ideographs Components*) could be more explicit about being for technically plain CJKUIs, to minimize misunderstandings of potential

users. The best name will depend on the intended purpose of the block, but could be e.g., *CJK Unified Auxiliary Ideographs* or *CJK Unified Accessory Ideographs* (besides Eiso's suggestion).

4. Other

- As for the name IDEOGRAPHIC DESCRIPTION CHARACTER HALF-TURN ROTATION, we note that the phrase "half-turn rotation" is unusual. Rotations are rarely full-turn (360°) rotations. If numerals are permitted, explicitly calling it a 180°-rotation might be preferable. Otherwise, we would prefer IDEOGRAPHIC DESCRIPTION CHARACTER ONE HUNDRED EIGHTY DEGREE ROTATION (as in IRGN2492) or simply IDEOGRAPHIC DESCRIPTION CHARACTER HUNDRED EIGHTY DEGREE ROTATION (*i.e.*, without the "ONE").
- On the side, we note that in the IDS syntax, the term `IDS_Ternary_Operator` does not match standard usage in programming language semantics (e.g., C and C++ have a *ternary* operator with the following intended/abstract semantics:

$$\lambda(b:\text{bool}, x:\tau, y:\tau).(b ? x : y):\tau$$

We propose to consider using `IDS_Ternary_Operator` if possible.

Acknowledgments

We thank Stephan Hyeonjun STILLER for comprehensive suggestions and proofreading.

(End of document)

Feedback on IRGN2572 “Proposal to encode 5 new ideograph description characters”

Author: FAN Ming

Date: 2022.09.02 (Revised)

Status: Individual Contribution

Many new IDCs were proposed in IRGN2572 for encoding, and some of them, especially unary operators (\leftrightarrow , \curvearrowright) and the subtraction operator (\ominus) were concerned by many that they might added ambiguity of IDS representation and are hostile to computer algorithms. However, I argue that most problems regarding ambiguity can at least be resolved theoretically by algorithms, and this article is the technical details of these solutions.

1. Rotation and Reflection (\leftrightarrow and \curvearrowright)

In [IRGN2572SATFeedback](#), there are mainly two issues raised by SAT regarding ambiguity of the rotation and the reflection operator, that is: 1) Issue of using sequences as argument of these operator, 2) Consecutive usage of these operators. Here I will show that all these two issues can be solved at least theoretically.

1) Consecutive usage of rotation and reflection

The fact is that any finite combination of rotations and reflections is equal to one of these 4 basic operations:

i. no-op (identity transformation, i.e. no actual change happened)

ii. horizontal reflection \leftrightarrow

iii. 180° rotation \curvearrowright

iv. vertical reflection (\leftrightarrow \curvearrowright , also \curvearrowright \leftrightarrow). So this can be treated as one single operator, and unless otherwise specified, the \updownarrow appeared in this article refers to this basic single operation.

The point is that the rotation and the reflection operator can be seen as *linear transformation*, and can be denoted as matrix:

$$\curvearrowright: \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \leftrightarrow: \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

And their combination follows the rules of *matrix multiplication*. So the result must be one of the four:

$$\text{i. no op: } \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{ii. horizontal reflection: } \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{iii. 180° rotation: } \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad \text{iv. vertical reflection: } \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

And this can be dealt with by algorithm easily. For example:

$$\leftrightarrow \curvearrowright \leftrightarrow \curvearrowright \leftrightarrow \curvearrowright \curvearrowright \leftrightarrow \leftrightarrow \curvearrowright =$$

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$= \updownarrow \text{ (i.e. } \leftrightarrow \curvearrowright \text{)}.$$

As for the “idempotent property” mentioned by SAT, it seems that it’s more accurate to call the rotation/reflection operations “*involutory*” (i.e. $\mathbf{A}^2=\mathbf{I}$ [**identity**]) instead of “*idempotent*” (i.e. $\mathbf{A}^2=\mathbf{A}$).

SAT argued that this will cause ambiguity (i.e. $\leftrightarrow \leftrightarrow$ = no-op) and will cause the infinite loop of

computer programs. In fact I'm little confused about this since the IDS sequences are always finite, and any computer programs that parse the IDS sequence will only need to reduce the possible redundant unary operations instead of expanding them recursively (e.g. $\leftrightarrow = \leftrightarrow \leftrightarrow = \leftrightarrow \leftrightarrow \leftrightarrow \leftrightarrow \dots$). Does SAT mean the program that *generate* (instead of parsing) the IDS automatically? If so, as mentioned above, any combination of reflection/rotation operator in an IDS representation will be reduced to **only one** basic operation (denote as "BO-OP" below). That means when generating IDS sequence, if we need to perform the BO-OP, we only need to perform once. So when called the function with BO-OP flag *on*, the next recursive call that pushed into the call stack must **do not need to be** the BO-OP operation, so the BO-OP flag of this call **must** be *off*. This can be easily controlled when programming. Thus we can avoid computer program calling BO-OP recursively (e.g. $\leftrightarrow \leftrightarrow \leftrightarrow \leftrightarrow \dots$), and avoid the stalemate.

2) Using sequences as argument of rotation/reflection

SAT also argues that using sequences as argument of rotation/reflection operator will cause ambiguous/strange expressions of IDS with an example: $\leftrightarrow \square \curvearrowright$ 干片, and appeal to ban using sequences as arguments of rotation/reflection operator. But in fact, it's *at least theoretically possible* to design an algorithm that, for example, converted the IDS $\leftrightarrow \square \curvearrowright$ 干片 mentioned above to it's canonical form: \square 片士, and below is the analysis:

An IDS sequence can be seen as a *prefix expression* (i.e. *Polish notation*), with the IDCs as operators, and the ideographs/strokes/components as operands, which is easy to be dealt by computer. What's more important, the rotation/reflection operator satisfies *distributive law* or "*quasi-distributive law*" with traditional IDCs, and here is the explanation (For convenience and easy to be understood, below I'll also illustrate the law with infix expression, which is easy to be read and understood by human, and in this kind of expressions, traditional IDC will be denoted as **Cx**, in which x is the last hex digit of the code point of the IDC (i.e. U+2FFx), instead of the IDC itself, to avoid confusing with prefix expression. For example, **C0** denotes \square , **C1** denotes \leftrightarrow , and **CB** denotes \curvearrowright , and the rotation is denoted as **R0**, the reflection is denoted as **R1**):







$$\begin{aligned}
 R1(op1 \mathbf{C1} op2) &= R1(op1) \mathbf{C1} R1(op2), \text{ ① i.e. } \leftrightarrow \square op1 op2 = \square \leftrightarrow op1 \leftrightarrow op2 \\
 R1(op1 \mathbf{C0} op2) &= R1(op2) \mathbf{C0} R1(op1), \text{ ② i.e. } \leftrightarrow \square op1 op2 = \square \leftrightarrow op2 \leftrightarrow op1 \\
 R0(op1 \mathbf{C5} op2) &= R0(op1) \mathbf{C6} R0(op2), \text{ i.e. } \curvearrowright \square op1 op2 = \square \curvearrowright op1 \curvearrowright op2 \\
 R0(op1 \mathbf{C9} op2) &= R0(op1) \mathbf{CA} R0(op2), \text{ i.e. } \curvearrowright \square op1 op2 = \square \curvearrowright op1 \curvearrowright op2
 \end{aligned}$$

Anyway, every **Rx** operator satisfies such laws with every **Cx** operator, the results mentioned above are only some typical examples, and I'll not pinpoint every result here. These laws can be easily implemented in recursive steps of parsing IDSes of computer algorithms, and, what's more important, when these laws was implemented, **every operand of Rx operator will be single character/stroke/component instead of sequence when recursion ended**. And that's just one step away from the canonical form that we want. For example, following the laws mentioned above, $\leftrightarrow \square \curvearrowright$ 干片 can be converted to $\square \leftrightarrow$ 片 $\leftrightarrow \curvearrowright$ 干. Obviously, if we link \leftrightarrow 片 with 片, $\leftrightarrow \curvearrowright$ 干 (i.e. \curvearrowright 干) with 士, it's clear that we will get the final canonical form \square 片士. To achieve this goal, what we need is simply a database that link the possible relations of reflection/rotation of characters, since the number of encoded ideographs is finite. For example, a given ideograph A, is the **i) horizontal reflection of A ii) rotation of A iii) vertical reflection of A A** itself, or another encoded


① That's just like a(b+c)=ab+ac, the so-called distributive law.





② Note in this situation the expression does NOT satisfies the distributive law, as the positions of op1 and op2 were reversed, and the IDC is not commutative operator, so I called the situation "quasi-distributive".









ideograph? Or does some two of them not encoded, but identical due to the symmetry property of this ideograph? Creating such a database may require extensive human work, however, with proper image processing tool and data as aid, it's unlikely that many mistakes will be made. So this database keeps 3 entries for every ideograph as mentioned above, and can be efficiently indexed by computer algorithms (at most ~300000 entries, which is not a large number), which means this process is not too compute-intensive at all.









And by the way, an interesting phenomenon is that when applied with the law mentioned above, we may not need proposed IDC **IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM RIGHT** and **IDEOGRAPHIC DESCRIPTION CHARACTER SURROUND FROM RIGHT (This does NOT mean that the author think these IDCs shouldn't be encoded)**, even already encoded U+2FF6 , U+2FF9  and U+2FFA  at all, since, for example, any op1 op2 can be represented as op1op2. However, this will allow sequences as arguments of reflection/rotation operator without fully converted, and this will possibly cause computer algorithms hard to find the relations, which will cause utter chaos. So this is just a random thought, and can be ignored.


2. Subtraction ()

There are also some concerns regarding the ambiguity of subtraction operator  when the subtrahend was arbitrary ideographs. Since this operator is somewhat arbitrary, this seems reasonable that the subtrahend should be restricted to minor strokes. Based on this, however, I argue that other ambiguity issues are either resolvable or unimportant:

First is the order of subtracted strokes (  豕 丿 乚 and   豕 乚 丿), this can be done by reordering the subtracted strokes when multiple consecutive subtraction operators are used, and can be done easily by computer algorithms.

As for sequences used as subtrahend (e.g.  豕  丿 乚 vs   豕 丿 乚), it seems not matter what the structure of subtrahend is, that means, we may try complete drop the traditional IDCs appeared in subtrahend, and adjust the number of subtraction operators. That can be done after a subtrahend finished converting to its canonical form, we count the total operators (denoted as k) of the subtrahend sequence, drop all traditional binary and ternary IDSs, and added k-1 subtraction symbols. This is also not hard to be implemented by computer algorithms. However, this may cause different characters corresponded to the same IDS(for example, if  A  BC and  A  BC is not the same ideograph), which will return false positives. However, this may be harmless. Also, the converted results by this procedure may be hard to be understood by human, but this seems also harmless since this was mainly done for machine checking.

Another concern is that since subtraction operator has no inverse operator, if there are also subtraction operators in subtrahend, thus will cause problems when using methods above. For example, if we represent 其 as  其  八 丿, there will be no simple way to convert this form to its canonical form. A worse example would be  A  BCD, when applied the method mentioned above, the result would be   A  BCD, which is an incorrect result. However, since the subtrahend is usually single stroke or combination of strokes, it seems no reason that the subtrahend needs to use this operator  to describe itself, so maybe we can simply prohibit the usage of the subtraction operator in the subtrahend.

The last problem is the 大-太-犬 problem [mentioned by SAT](#). However, I argue that actually this situation may be rare. The ideographs that needed to use the subtraction operator typically has strong connection with some encoded character (for example, 共 彳 其 亠 且 and  東 丿), it's easy to identify that what will we get if we add a stroke to them, and is unlike to the situation such as

“人 add a stroke we get 及”. Even the “大-太-犬 situation” occurred, the structure and stroke count of involved ideograph usually quite simple and less, which is easy to be identified by human. So this issue may be not a big problem.

(End of Document)