WG2 N5282

To: Script Encoding Working Group, UTC, and WG2
From: Debbie Anderson (on behalf of LUCP, Stötzner et al.)
Date: 17 June 2024
Subject: Leibniz project - Supplemental documents

Attached are three documents with supplemental information to WG2 N5277 "Proposal to add historic scientific characters":

L-2403

Leibniz Unicode Characters Project
(LUCP)
Philiumm Project, Université de Paris VII
Leibniz-Archiv Hannover
Andreas Stötzner, Fontentwicklung

Comments to Script-ad-hoc questionaire about the LUCP proposal to add historic scientific characters to the UCS

Related: Script Ad Hoc working group, comments sent by Dr D. Anderson, 23-07.06. / 23-11.30.; doc. L-2402 (proposal)
Contributors: Uwe Mayer, Siegmund Probst, Elisabeth Rinner, Andreas Stötzner, Achim Trunk, Charlotte Wahl (LUCP)
Doc. number: L-2403
Version: AS/ER 24-02.11.
Status: Final

*#1a The "**ambiguity signs**" are a set of generatively built signs. We are going to need justification as to why they shouldn't be dealt with via sequences.*

¶1a All these characters have a) a unique graphical representation and b) a different semantic value. A more or less optical resemblance of some of them is of no importance. We provide additional information about their meaning, systematic and usage in relevant sources, in the document *"Towards the encoding of Leibnizian Ambiguity Signs"*.

These characters are essential for editing of those sources, for text processing, searching and character recognition operations. Just as no one would expect e.g. the character PLUSMINUS ± being handled via a sequence encoding of PLUS and MINUS characters, the same rationale applies for the set of ambiguity signs. By this we follow the same method as used, for example, for Braille patterns (2800–28FF) and the Yijing hexagram symbols (4DC0–4DFF).

¶1b Remark about ambiguity signs *numbering* in the proposal: The set of ambiguity signs is based on a study by A. Trunk and a list of Leibnizian characters by E. Rinner, in which the firsts steps towards an overall systematization of this group have been made. The R- and T-numbers refer technically to those papers and constitute an in-house proto-standard of the Leibniz Edition project group (LE). As a following step towards encoding we have further systematized the complete ambiguity sign's set and applied a new, consistent nomenclature, with A-, B- and C- prefixes, according to the several sub-systems of the whole set.

*#2 All the slashed, backslashed, and **crossed digits** should be dealt with via sequences of combining symbols, since these marked-up digits aren't regular digits, but rather are part of an elaborate calculation scheme worked out in two dimensions.*
¶2 We present a fair amount of printed sources in which these characters are testified. An encoding of these as sequences may make sense in theory but produces endless hazards and headaches in editorial practice. There are two issues with sequence encoding. First, these characters are evident in a lot of printed sources where they clearly appear as discrete letters. This needs to be transported 1:1 into digital text. Second, a use of combining overlay slash marks with the regular digits will in the majority of cases produce visual results which are very badly recognizable, often almost unreadable. For readers and scholars this opens the danger of misinterpretations and errors. Therefore, it is necessary to give each of these glyphs its own codepoint in order to allow appropriate, individually adjusted glyph design which takes care of the specific challenges these glyphs present.

*#3 **Enclosed sun and moon** are at the btm left of this table (a summary table from 1931), and tartar-salt near the top of the 2nd column.*
¶3 The source mentioned strengthens the case for the encoding of these two characters.

*#4 I don't know if the **rotated oil** symbol is contrastive in the supporting doc. In other sources it means 'flint' (lapis silex), so here it might just be a graphic variant for 'oil'?*
¶4 The ALCHEMICAL SYMBOL FOR OIL BOILED is evidently distinguished from the known ALCHEMICAL SYMBOL OIL (1F746) by a different glyph and a different meaning. It is clearly **not** a mere graphic variant of an existing character. In the proposal we show an additional proof from another alchemical reference work (Geßmann).

*#5 I suppose the **retort** might be distinct enough for its own code point. We did plenty of that for the crucible symbol, after all.*
¶5 We prefer a separate codepoint for this character, because it is a scriptive character and not a pictogram like 1F76D.

*#6 The **hora** sign is an allograph of what we already have. … a pictogram of an hourglass.*
¶6 There are certain ambiguities in the documented relations between several encoded 'hourglass' characters. 1F76E we consider the semantically closest candidate for the use cases we have in mind. However, the current glyphic representation of 1F76E in the codepage shows a calligraphic variant of that character, which in many sources (typically and also in the sources we deal with) appears in a straight-lined, X-like shape. See samples of that in the proposal. Therefore we propose a seperate encoding of "hora", alongside the existing 1F76E. This solution seems advisable because a hard-coded distinction of these two

versions (heart-shaped vs. X-shaped) is neccessary in some editorial contexts which deal critically with the discussion of various writing conventions and traditions.

#7 ***Realgar 3*** *appears to be ☋. Maybe some connection to the alchemical use of ☋ for ' purify'? Nitre flowers is probably the ordinary symbol for nitrum, U+1F715.*

¶7 The Realgar symbol has a glyph based on an arc much more open than in the typical case of 260B, so that the different meaning is sufficiently manifest in a distinct shape. In the case of Nitre flowers we have established that 1F715 is the appropriate codepoint.

#8 INFINITY SIGN WITH DOTS This character could potentially already be accurately represented using the sequence ‹∞̣̇›.

¶8 This solution would require a sequence of **three** different existing characters of which two are combing characters. In which succession they are to be arranged properly? This is not at all obvious and hence this model would result in at least two different sequences in practice, which makes the identification of the character difficult to impossible, because one can never be sure (in a search) to reach all instances, when they happen to be encoded differently (although looking the same visually). Moreover, there is precedence of structurally similar cases encoded. For instance, the Mathematical Operators block contains a range of characters whose glyphs are built of other well-known base glyphs and dots, in the range 2238 to 2255; e.g. HOMOTHETIC (223B) and GEOMETRICALLY EQUAL TO (2251) can be seen as analogue characters, which have their own meaning, despite being graphically composed of prevalent glyphs. There are also many characters which are represented by combinations of e.g. = or < with a SLASH or 'solidus overlay'. It is a feature inherent in mathematical notation that new expressions are created by combining established elements, in the one way or the other. Nevertheless, it is justifiable to assign a separate codepoint for such a character, because its use and meaning are testified and this encoding is conformant with established encoding practice and principles.

#9a COMBINING SUPERSCRIPT UNDERSCORE, COMBINING SUPERSCRIPT UNDERSCORE WITH DOT BELOW, COMBINING SUPERSCRIPT BREVE MARK BELOW These should be handled through OpenType substitution. A few fonts have already implemented mark miniaturization when a combining mark such as dieresis, caron, breve and dieresis are attached to miniature characters such as superscript characters, subscript characters and combining mark letters.

#9b ***Certain "superscript" characters***

I'm not convinced that encoding mathematical notation using encoding of superscript forms is appropriate. In mathematical layout, superscripting is a function of the notation, and anything placed in a such a slot, will be scaled and positioned. Suggest we require the use of mathematical layout engine with a proper rich-text (or markdown) format and encode only

full-size symbols.
¶9a+b, COMBINING SUPERSCRIPT UNDERSCORE, COMBINING SUPERSCRIPT UNDERSCORE WITH DOT BELOW: As it is right that the question concerns the setting expressions in mathematical notation, we follow th*e objection that an encoding by means of combining characters as proposed will not give an appropriate result. The characters were meant for encoding an underlined expression of (possibly) several characters with a dot beneath the line which* is situated in its middle, which will be better represented using the mathematical layout engine.
¶9a+b, (was:) COMBINING SUPERSCRIPT BREVE MARK BELOW: It is necessary to identify this character on the encoding level, not only on the level of visual representation. A glyphic workaround via Opentype substitution is no option here. The glyph's vertical position is on the baseline and its typical usage is in combination with superscript digits.
The proposed name of this character is COMBINING BOMBELLI POWER MARK.

#10a LOWERCASE Y WITH DIERESIS BELOW This character should be handled using ‹y̤›. [0079 + 0324]
#10b ***Diaeresis below***
I'm not persuaded that y with diaresis below must be a precomposed letter.
We would be permanently exclude encoding a diaeresis below for any future linguistic purpose, or encur one of our normalization exceptions where things that look normalizable aren't. The argument that the layout will look bad does not take into account that this is not in the context of ordinary text layout, but requires an engine capable of mathematical layout.
¶10 This character is a mathematical symbol on its own and does not stand for a quantity that is derived from y via some mathematical operation and, as such, is denoted by adding diacritica as it is the practice in current mathematics. A hard-coding would be much preferred, not at least because a sequence encoding is unpractical and will produce bad (and dangerous) visual results. We may eventually accept a sequence encoding in this case.

#11 OMICRON-UPSILON SIGN This character, if encoded, should probably be handled as a Greek lowercase letter. An upper and lowercase case pair was proposed by Everson in L2/98-210. In the Greek script, the ȣ letter, is currently meant to be represented through the use of contextual or discretionary ligatures.
¶11 We would advise the encoding this as lowercase *and* capital Greek letters.
On the one hand, Everson (1998) has demonstrated the case for this character in both historic and recent typographical and epigraphical usage for Greek language texts. On the other hand, in mathematical notation it is not used as a mere graphic representation of **ο_υ** but as an peculiar ideogram with a specific meaning. For the use in mathematical context, a handling of this character as an OT ligature is no option, because it would result in an encoding of (Greek) **ο υ** in which case the specific semantic content is lost and the text bit or formulæ in question get spoiled by the ambiguity of other occurrences and meanings of **ο** and **υ**.

#12 REVERSED CAPITAL L A lower and uppercase reversed l can be found in works by Canepari. The uppercase letter is in his works used as phonetic category symbol, while the lowercase is used for a phoneme. This character, if encoded, should be handled as a standard Latin case pair.
¶12 We would follow the idea of encoding this as a lowercase and capital letter pair.

#13 CAPITAL DOUBLE X SIGN This character, if encoded, should be handled as a standard Latin case pair.
¶13 We don't see any real case for a lowercase variant of the capital double X, which appears (as known so far) as a rare mathematical symbol only. We prefer a single encoding of the capital form (as a *symbol*) but would also follow the idea of encoding this as lowercase and capital, if that is regarded more appropriate.

#14 ***Combining equal sign***
We should not be lured into using running-text letter-based typography to do annotations of mathematical symbols. We should require that Leibniz use of annotations of the equal sign to indicate things like numbered equations (placing e.g. a (23) above an =) are to be treated just like annotation of mathematical operators or arrows in chemistry: it's a feature of the layout system, and only the elements, that is, standard sized =, 2, 3, (and) are to be encoded.
How the annotation is scoped is not part of plain text, but perhaps someone will invent a markdown like protocol akin to Murray's plain text math. How the annotation is placed and scaled is likewise a feature of Leibnizian mathematical layout and not part of Unicode plain text.
¶14 As some longer expressions are combined with the equal sign as well as expressions that consist of a single character and as we prefer a uniform encoding of all such cases, we follow the idea to treat them as annotations to the equal sign which should be taken into account otherwise.

#15 ***Combining double slash***
here "double" is ambiguous. It would have to be "double-wide" in distinction to a putative cousin of 0338 that uses a doubled slash. Also, usage of "slash" vs. "solidus" in naming.
¶15 We concur with this argument and propose the naming "COMBINING DOUBLE-WIDE SLASH OVERLAY" for to avoid unnecessary ambiguity. We applied the same nomenclature in the case of COMBINING DOUBLE-WIDE ENCLOSING SPIRAL MARK and COMBINING DOUBLE-WIDE OVERLINE WITH TERMINALS.

– – – –

ISO/IEC JTC1/SC2/WG2 **Nxxxx**
LUCP **L-2404**

Universal Multiple-Octet Coded Character Set
International Organization for Standardization
Internationale Standardisierungs-Organisation
Organisation Internationale de Normalisation
Διεθνής Οργανισμός Τυποποίησης
Международная организация по стандартизации

Doc Type: Working Group Document

Title: Towards the Encoding of Leibnizian Ambiguity Signs

Related: L-2402 Proposal to add historic scientific characters to the UCS
Source: Uwe Mayer, Siegmund Probst, Elisabeth Rinner, Andreas Stötzner, Achim Trunk, Charlotte Wahl
Status: Preliminary
Version: 1.1
Date: February 15, 2024

The proposal requests the encoding of 57 ambiguity signs that are testified in works of Gottfried Wilhelm Leibniz (1646–1716), in editions of his works, and in literature from the field of history of mathematics. This document aims to explicate the systematics and groupings of these signs and to introduce their meaning.

1. Introductory remarks on Leibnizian Ambiguity signs

Early in his career, Leibniz wrote several texts in which he designed and systematically examined systems of symbols for analytical calculations. Complex systems of ambiguity signs, with which more than two cases are distinguished, represent an elementary and novel component of this *Méthode de l'universalité*. As part of the *Ars Characteristica*, the treatment of this method belongs to that branch of philosophy that is "the art of forming and arranging characters so that they agree with thoughts" (Mugnai 2018, abstract).

However, Leibniz's interest is not only theoretical. Rather, the design of higher ambiguity signs is closely linked to his occupation with the mathematics of conic sections. There he has to consider sub-cases of cases, but would like to write only one equation to treat them all together, since often the equations do not differ except for the signs of the terms. The use of double signs, which allows to represent two cases simultaneously, is already part of common practice in mathematics. The characters ± and ∓, which are still in use today, are used for this purpose.

According to current knowledge, Leibniz designed six different systems over the course of time—as long as transitional forms and preliminary considerations are ignored.

One reason why a system of ambiguity signs is abandoned by Leibniz is the consideration that a large number of specific new printing types are required if a system does not rely on the traditional set of printing types. Leibniz's further penetration of the topic also led to improved, simpler or, in some cases, even more complex characters. The draft of the first system, for example, provides for special characters to express the product of the two double signs A-01 ∓ and A-02 ⩱. For these, Leibniz envisages the ligature A-07 [glyph] and A-08 [glyph] of these two symbols with the LEIBNIZIAN PRODUCT SIGN he typically uses. Only later does he take advantage of the fact that the mathematical meaning can also be expressed using existing symbols.

Some systems also take into account the relationship between several ambiguity signs in the same expression. Ambiguity signs can be *homogeneous* or *corresponding* and therefore dependent on one another, as well as *heterogeneous* and therefore independent of one another.

Likewise, it was only in his 5th system that Leibniz gave up structuring ambiguity signs according to the distinction between cases and sub-cases as they arise in the calculation process. Even though, from the perspective of modern mathematics, it makes no difference with regard to calculations whether the ambiguity sign (*mp*)*m* (i. e. a sign which has the sub-cases *mp* in the first case and *m* in the second case, with *p* as abbreviation for plus and *m* for minus) or *m*(*pm*) (i. e. a sign which has *m* in the first case and the sub-cases *pm* in the second case) is used, they do refer to two fundamentally different conceptions of the mathematical situation.

Design questions also play a role when considering the layout of systems of ambiguity signs, which lead Leibniz to the discussion of different positioning of lines and thus to variants that are compared to the systems ultimately favored.

Particularly in Leibniz's drafts, the ambiguity signs that occur can contribute to the dating of the texts, as a sequence of systems can be observed.

2. Overview of systems and character names for the UCS

Some systems of ambiguity signs are designed in such a way that they can be extended to distinguish any number of cases. In systems that use specific new characters and do not use the existing character set of a typesetting box, the surviving texts only contain characters that distinguish a maximum of four different cases. Usually, not all possible combinations of *p* and *m* occur in the texts. However, the systematics described or reconstructed on the basis of the surviving texts often allows to reconstruct the full set of ambiguity signs that belong to a system.

The overview in the appendix therefore contains only systems that use special new characters. For them, a list of all possible cases is provided. A representation of their glyphs is given, provided their use is documented in the texts written by Leibniz. In the overview, the meaning of the ambiguity signs is also stated in an abbreviated form.

For the encoding of Leibniz's ambiguity signs in the Unicode standard, we propose a name consisting of the components "AMBIGUITY SIGN", an identifier for the system ("A" for system 1, "B" for system 2, and "C" for system 5) which is followed by a hyphen, and a sequential number, with a leading zero being added to single-digit numbers. The character-specific parts of the proposed names are also included in the overview.

The characteristics of the systems are briefly described below with references to the overview.

2. 1 System 1

System 1 is based on the signs + for *p* and – for *m* that are still in use today, with A-01 ∓ being understood as a combination of these signs. The additional bar in A-02 ⩱ represents negation, so the sign stands for *mp*.

When looking at the layout of the triple signs A-03 +∓, A-04 +⩱, A-05 −∓ and A-06 −⩱, it can be seen that Leibniz takes the structure of possible distinctions of cases and sub-cases into account. In the texts, only signs for which a distinction between two sub-cases arises in the second case are described and documented. This second case with both sub-cases is represented in the right part of the sign in analogy to the associated double signs A-01 ǂ and A-02 丰: the two upper crossbars suspended from the vertical bar again refer to the combination of *p* and *m*, while the third, lower bar appearing in A-04 +丰 and A-06 −丰 represents the negation of this part of the ambiguity sign. The value of the first case is on the left side (*p* or *m*). This part is connected to the second crossbar from the top in the right part of the ambiguity sign.

A special feature of the 1st system are the ambiguity signs A-07 ǂ† and A-08 丰† that stand for products of the double signs of the system.

2. 2 System 2

System 2 develops from the same combination of + and − that forms the sign A-01 ǂ. Unlike in the first system, the negation of the sign is not expressed by a third crossbar with the same width, but by a longer, horizontal bar that is placed at the bottom of the vertical bar (e. g. in B-01 ±). Such negations of the complete ambiguity sign can be applied to all of them, with B-05 +±, B-06 +ǂ, B-07 −±, and B-10 ±+ being examples.

As before, triple signs are composed of signs and double signs, with the first case on the left (subdivided or not) and the second case (not subdivided or subdivided) on the right. During this transition from double sign to triple sign, the negation bar of B-01 ± slides upwards, so to speak, so that the vertical bar in the partial sign that is given in one of Leibniz's texts, which rather coincidentally has the same design as A-02 丰, now protrudes at the bottom, and the width of the crossbar is adjusted to that of others in the sign. In contrast to the first system, both parts of the triple sign are composed by connecting the horizontal bar of the single sign part to the top bar of the double sign part. The ambiguity signs B-02 +†, B-03 +ǂ, B-04 −ǂ, B-05 +±, B-06 +ǂ, B-07 −±, B-08 †+, B-09 †− and B-10 ±+ are of this kind.

The principle of composition is meant to be continued for distinguishing further ambiguities (i. e. the type of combination of cases that are distinguished). B-11 +丰 and B-12 −丰 which represent the negation of *p*(*mp*) and *m*(*mp*) make it clear that the negation bar is "moved up" again in these partial signs, and that its size corresponds to the size of all other crossbars.

As Leibniz discussed questions about the suitability of different positions of the crossbars when designing this system, the six ambiguity signs B-13 +†, B-14 +ǂ, B-15 +±, B-16 ±+, B-17 +ǂ, and B-18 +丰 have come down to us. They represent variants of ambiguity signs of the standard form.

2. 3 System 3

Leibniz builds system 3 from the ambiguity signs A-01 ǂ and B-01 ±, which are used for *pm* and *mp* in the 2nd system. This means a reduction of the number of characters required, while still any complex ambiguity as well as all dependencies between ambiguity signs (i. e. whether they are *homogeneous*, *corresponding* or *heterogeneous*) can be expressed. To do this, numbers are added to the left and right of A-01 ǂ and B-01 ± according to certain given rules. Levels of case distinctions can also be expressed by building nested expressions according to rules. The entire expression is marked by a bracket with *vinculum* (i. e. they are connected by an overline).

As a ligature of the brackets "(" and ")" with the *vinculum* is needed, LEFT VIRGULA PARANTHESIS and RIGHT VIRGULA PARANTHESIS are included in the proposal to encode these expressions. The overview at the end of the document does not

3

contain any characters that are specifically assigned to this system since the glyphs of the ambiguity signs used in this system match with those of signs from the first two systems.

2. 4 System 4

System 4 has no new characters at all and instead uses lowercase letters of the Greek alphabet. Ambiguity is expressed by strings of certain pairs of letters such as α and ω, β and ψ, or γ and χ, where the two letters are equidistant from the beginning or end of the alphabet, and the letter from the beginning stands for *p* and the other for *m*. As in the 3rd system, the letters are written one after the other, following the order of the cases, and marked by brackets and a *vinculum*.

By using several pairs of letters it is possible to express the relationship between the ambiguity signs that occur in an expression, because the same pair of letters is used for interdependent ambiguity signs and different pairs of letters for independent ones. The level of cases can be represented by structuring these sequences with commas.

2. 5 System 5

The principles of the standard form of system 5 and all its predecessors were reconstructed on the basis of ambiguity signs that can be found in Leibniz's manuscripts.

In its final version (see 3. 5 subgroup "Standard" in the overview), the 5th system probably has the simplest structure in the design of the glyphs. The set of *n*-fold ambiguity signs consists of almost all possible *n*-combinations of *p* and *m*. All the signs with the meaning *pp…p* (string with *n* characters) and *mm…m* (string with *n* characters) are omitted as they have the same meaning as + and –. The level of cases is not represented in this system. Likewise, dependencies that occur between several ambiguity signs of the same expression are not represented.

The double signs C-16 ∓ and C-17 [sign], the triple signs C-18 [sign], C-19 [sign], C-20 [sign], C-21 [sign], C-22 [sign], and C-23 [sign] as well as the quadruple signs C-24 [sign], C-25 [sign], C-26 [sign], C-27 [sign], C-28 [sign], C-29 [sign], C-30 [sign], and C-31 [sign] belong to this group.

Their design follows a uniform principle. On a vertical bar, horizontal bars of equal length are positioned at equal distances depending on the cases distinguished in the sign. If a bar represents *p*, it is bisected by the vertical bar. If it stands for *m*, it starts on the left at the same distance from the vertical bar as the *p*-bars, but already ends at the vertical bar.

This approach was derived from the 2nd system, and there is a total of four stages in the development of the standard version of system 5 of which only few examples of ambiguity signs have been preserved. The representation of the level of case distinction is a common feature of all these systems.

- The "Transition Form 2 → 5" (subgroup 3. 1) preserves the division into left and right part from the 2nd system as a means to represent two cases on the top level of case distinction. What is new, however, is the reduction of the representation of the cases on the second level, where, among other things, a crossbar shortened to the half width appears for the first time. The ambiguity sign C-01 [sign] has been handed down in this group.
- In the ambiguity signs of the group "prae-pro-proto-5" (subgroup 3. 2), instead of being divided into left and right halves to distinguish the two cases, there is a subdivision of the sign into an upper and lower section which is arranged along a vertical bar. In the case of the two known ambiguity signs C-02 [sign] and C-03 [sign], a subordinate case distinction occurs in the first (upper) case, where *p* and *m* are expressed by a long and a short horizontal bar, respectively, which are positioned in the middle of the vertical bar and to its left, respectively. A connection of these two

4

horizontal bars by a short vertical bar at their left which ends above the upper horizontal bar illustrates that they belong to the same group of sub-cases.

- From the group “pro-proto-5” (subgroup 3. 3), only the ambiguity signs C-04 ‡ and C-05 ‡ have come down to us. For both, signs that represent the same type of ambiguity in the previous group “prae-pro-proto-5” (subgroup 3. 2) are also known. Compared to them, the upper halves of the short vertical bars which illustrate in the previous variant that the horizontal bars that are connected by them belong to the same group of sub-cases are omitted, so that the resulting glyphs are further reduced compared to their predecessors.
- The group “proto-5” (subgroup 3. 4) already shows very close similarities to the standard form. However, Leibniz continues to distinguish the levels of case distinction in the triple signs, with the first case or its two sub-cases being shown in the upper part of the signs, the second case or its sub-cases in the lower section. C-09 ‡ and C-10 ‡, which stand for *p*(*mp*) and (*pm*)*p* respectively, differ only in the positions of the short crossbar that represents *m*. They have the same meaning as long as the levels of case distinction are ignored. The triple signs C-08 ‡ and C-11 ‡ also belong to this group, as well as the double signs C-06 † and C-07 ‡.

  In this system there are also composed forms: C-12 ± represents a composition of *mp* and *p*(*mp*) according to a rule, the negation is C-13 ∓. There are also combinations based on this: in C-14 ‡, C-13 ∓ occurs as the second case of an ambiguity sign whose other case is *p*, while in C-15 ‡ it is the first case, again in combination with *p*.

Thus, it is giving up the representation of the level of case distinctions and the choice of equal distances that ultimately constitute the final step towards the standard form. At the same time, this reduces the number of ambiguity signs to be taken into account. While in its standard form the continuation of the system for distinguishing more cases is known, it is not clear from the surviving texts for the transitional and preliminary forms.

2. 6 System 6

System 6 which can be derived from several manuscripts shares its basic idea with system 4. In all examples known from these texts, *p* is expressed by 1. For *m*, Leibniz uses 3 in one text and 2 in all others.

By relying entirely on character types which are included in the usual typesetting box, this system does not have to be taken into account in the proposal.

References and additional literature

Mugnai 2018 Massimo Mugnai, *Ars Characteristica, Logical Calculus,and Natural Languages*, in: Maria Rosa Antognazza (ed.): *The Oxford Handbook of Leibniz*, Oxford University Press. Oxford 2018, p. 177-207 (published online as https://doi.org/10.1093/oxfordhb/9780199744725.013.20).

Probst & Trunk 2019 Siegmund Probst and Achim Trunk: *Einleitung*, in: Gottfried Wilhelm Leibniz: *Sämtliche Schriften und Briefe*, Vol. VII, 7.

Trunk 2016-2017 Achim Trunk, *Sechs Systeme: Leibniz und seine signa ambigua*, in: Wenchao Li (ed.): *Für unser Glück oder das Glück anderer, Vorträge des X. Internationalen Leibniz-Kongresses*. Hildesheim 2016-2017, vol. 4.

Overview of Leibniz's systems of ambiguity signs

The columns in this overview contain the following information:

1 number within the system
2 mathematical meaning
3 specific part of proposed character name (if it is part of the proposal)
4 representative glyph
5 ID in the proto standard as defined by the Leibniz-Edition (LE)
6 number and glyph in the current font of the Leibniz-Edition (if existing). The current glyphs can deviate from their actual layout.

The following signs are used to express the meaning of the signs:

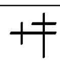
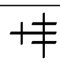
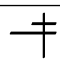
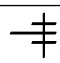
p plus
m minus
(…) group of sub-cases
non[…] negation
· multiplication of signs
∘ composition of signs

1 System 1

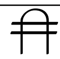
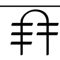
1. 1 Double Signs

nr.	*meaning*	*character name ID*	*repr. glyph*	*ID in proto standard of LE*	*font of LE*
1	Sys. 1 pm (= sys. 2 pm)	A-01		T-01 = T-19	12: 
2	Sys. 1 mp (= sys. 2 (parts) mp)	A-02		T-20 = T-02	13: 

1. 2 Triple Signs

1	Sys. 1 p(pm)	A-03		T-03	—
2	Sys. 1 p(mp)	A-04		T-04	—
3	Sys. 1 m(pm)	A-05		T-05	—
4	Sys. 1 m(mp)	A-06		T-06	—

1. 3 Multiplication Forms

1	Sys. 1 pm · pm	A-07		T-07	—
2	Sys. 1 mp · pm	A-08		T-08	—

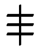
2 System 2

2. 1 Double Signs

2. 1. 1 Standard Forms

1	Sys. 2 pm (= sys. 1 pm)	(A-01)		T-01 = T-19	6: 
2	Sys. 2 mp	B-01		T-12	7: 

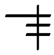
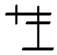
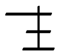
2. 1. 2 Standard Forms (Parts)

1	Sys. 2 (parts) pm	—	—	—	—
2	Sys. 2 (parts) mp (= sys. 1 mp)	(A-02)		T-20 = T-02	230: 

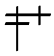
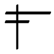
2. 2 Triple Signs

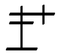
2. 2. 1 Standard Forms

2. 2. 1 a) Type _ (_ _)

1	Sys. 2 p(pm)	B-02		R-118	118: 
2	Sys. 2 p(mp)	B-03		T-10	120: 
3	Sys. 2 m(pm)	—	—	—	—
4	Sys. 2 m(mp)	B-04		T-09	—
5	Sys. 2 *non*[p(pm)]	B-05		T-15	222: 
6	Sys. 2 *non*[p(mp)]	B-06		R-119	119: 
7	Sys. 2 *non*[m(pm)]	B-07		R-84	84: 
8	Sys. 2 *non*[m(mp)]	—	—	—	—

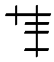
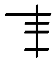
2. 2. 1 b) Type (_ _) _

9	Sys. 2 (pm)p	B-08		T-17	231: 
10	Sys. 2 (mp)p	—	—	—	—
11	Sys. 2 (pm)m	B-09		R-233	233: 

12	Sys. 2 (mp)m	—	—	—	—
13	Sys. 2 *non*[(pm)p]	B-10		R-234	234: 
14	Sys. 2 *non*[(mp)p]	—	—	—	—
15	Sys. 2 *non*[(pm)m]	—	—	—	—
16	Sys. 2 *non*[(mp)m]	—	—	—	—

2. 2. 2 Standard Forms (Parts)

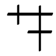
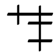
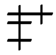
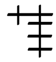
2. 2. 2 a) Type _ (_ _)

1	Sys. 2 (parts) p(pm)	—	—	—	—
2	Sys. 2 (parts) p(mp)	—	—	—	—
3	Sys. 2 (parts) m(pm)	—	—	—	—
4	Sys. 2 (parts) m(mp)	—	—	—	—
5	Sys. 2 (parts) *non*[p(pm)]	—	—	—	—
6	Sys. 2 (parts) *non*[p(mp)]	B-11		T-25	224: 
7	Sys. 2 (parts) *non*[m(pm)]	—	—	—	—
8	Sys. 2 (parts) *non*[m(mp)]	B-12		R-226	226: 

2. 2. 2 b) Type (_ _) _

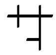
9	Sys. 2 (parts) (pm)p	—	—	—	—
10	Sys. 2 (parts) (mp)p	—	—	—	—
11	Sys. 2 (parts) (pm)m	—	—	—	—
12	Sys. 2 (parts) (mp)m	—	—	—	—
13	Sys. 2 (parts) *non*[(pm)p]	—	—	—	—
14	Sys. 2 (parts) *non*[(mp)p]	—	—	—	—
15	Sys. 2 (parts) *non*[(pm)m]	—	—	—	—
16	Sys. 2 (parts) *non*[(mp)m]	—	—	—	—

2. 2. 3 Variants

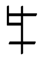
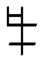
1	Sys. 2 p(pm)	B-13		T-13	228: 
2	Sys. 2 p(mp)	B-14		T-14	223: 
6	Sys. 2 *non*[p(mp)]	B-15		T-16	225: 
10	Sys. 2 (mp)p	B-16		T-18	229: 
6	Sys. 2 (part) *non*[p(mp)]	B-17		R-232	232: 
6	Sys. 2 (part) *non*[p(mp)]	B-18		R-227	227: 

3 System 5

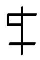
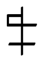
3. 1 Subgroup “Transition Form 2 → 5”

—	Sys. Transition Form 2 to 5 p(mp)	C-01		T-75	—

3. 2 Subgroup “prae-pro-proto-5”

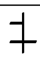
—	Sys. prae-pro-proto-5 (pm)p	C-02		T-72	—
—	Sys. prae-pro-proto-5(mp)p	C-03		T-73	—

3. 3 Subgroup “pro-proto-5”

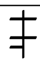
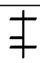
—	Sys. pro-proto-5 (pm)p	C-04		R-220	220: 
—	Sys. pro-proto-5 (mp)p	C-05		T-74	221: 

3. 4 Subgroup “proto-5”

3. 4. 1 Double Signs

1	Sys. proto-5 pm	C-06		T-46	—
2	Sys. proto-5 mp	C-07		T-71	—

3. 4. 2 Triple Signs

1	Sys. proto-5 p(pm)	C-08		T-69	—
2	Sys. proto-5 p(mp)	C-09		T-47	—
3	Sys. proto-5 m(pm)	—	—	—	—

4	Sys. proto-5 m(mp)	—	—	—	—
5	Sys. proto-5 (pm)p	C-10		T-42	—
6	Sys. proto-5 (mp)p	C-11		T-70	—
7	Sys. proto-5 (pm)m	—	—	—	—
8	Sys. proto-5 (mp)m	—	—	—	—

3. 4. 3 Composed Forms

—	Sys. proto-5 mp ∘ p(mp)	C-12		T-41	—
—	Sys. proto-5 *non*[mp ∘ p(mp)]	C-13		T-48	—
—	Sys. proto-5 p*non*[mp ∘ p(mp)]	C-14		T-44	—
—	Sys. proto-5 *non*[mp ∘ p(mp)]p	C-15		T-43	—

3. 5 Subgroup “Standard”

3. 5. 1 Double Signs

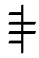
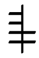
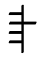
1	Sys. 5 pm	C-16		T-55	8:
2	Sys. 5 mp	C-17		T-56	38:

3. 5. 2 Triple Signs

1	Sys. 5 ppm	C-18		T-57	99:
2	Sys. 5 pmp	C-19		T-58	39:
3	Sys. 5 mpp	C-20		T-45	85:
4	Sys. 5 pmm	C-21		T-59	40:
5	Sys. 5 mpm	C-22		T-60	42:
6	Sys. 5 mmp	C-23		T-61	41:

3. 5. 3 Quadruple Signs

1	Sys. 5 pppm	C-24		T-66	46:
2	Sys. 5 ppmp	C-25		T-65	45:
3	Sys. 5 pmpp	C-26		T-64	44:
4	Sys. 5 mppp	C-27		T-63	43:
5	Sys. 5 ppmm	C-28		T-67	47:

6	Sys. 5 pmpm	—	—	—	—
7	Sys. 5 mppm	C-29		R-49	—
8	Sys. 5 pmmp	—	—	—	—
9	Sys. 5 mpmp	—	—	—	—
10	Sys. 5 mmpp	C-30		T-68	—
11	Sys. 5 pmmm	—	—	—	—
12	Sys. 5 mpmm	C-31		T-62	48: 
13	Sys. 5 mmpm	—	—	—	—
14	Sys. 5 mmmp	—	—	—	—

Slashed digits as encoded characters

Doc. type: individual contribution
Source: Uwe Mayer, Siegmund Probst, Elisabeth Rinner,
Andreas Stötzner, Achim Trunk, Charlotte Wahl
Related: L-2402 – Proposal to add historic scientific characters to the UCS
Doc.-no.: L-2405
Date: 2024-02.14.

Contents

1. Background

In the history of mathematical notation slashed digits play an important role. They emerged as a means of marking in the course of a calculation process. Unlike slashed or barred letters (e.g. the Scandinavian ø) the base character and the addendum are not written at once, but in two separate actions in the course of the calculation process. The custom of this widely used kind of notation was transferred to printing, when books upon algebra and calculus were in demand. We find digits with a single slash in many treatises – and in specimen books of printing shops. On the other hand, markings other than the single slash are rare. They occur so far mainly in manuscripts of Leibniz. These sources are due for editing and therefore 5 sets of slashed digits have been included in the Character Encoding Proposal (L-2402).

2. The semantics of slashed digits and objectives of their usage in writing, printing, and digital texts

Slashed digits are a common feature of several procedure for calculation by hand. Historically, they appear in the western world in the so-called *galley division* whose name is derived from the shape of their configuration of digits that resemble ships:

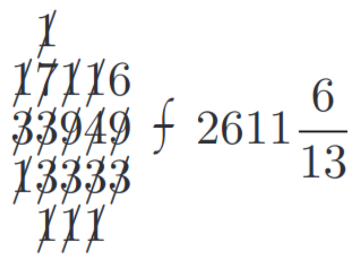


Fig. 1: Typical configuration of the output from galley division (VE VII, 8 p. 54).

Today, slashed digits are used among others when subtracting numbers by hand according to the “american method”. Again, a typical configuration of horizontal lines, signs, digits, slashed digits, and smaller digits occurs.

In many cases, slashed digits are the result of crossing out digits in specific steps of the calculation procedure. Even though it would not lead to a different result of the calculation if digits would be crossed out differently, the way of crossing out is usually shared by members of larger communities. Thus, different directions of crossing out can indicate an origin of texts within different traditions.[11]

The written output of such procedures usually has a semantic ambiguity, which is expressed, among other things, in different verbalizations of the procedure, with the differences reflecting the use of sometimes completely different ideas about mathematical operations and procedures (Radatz et al. 1999 p. 132-133).

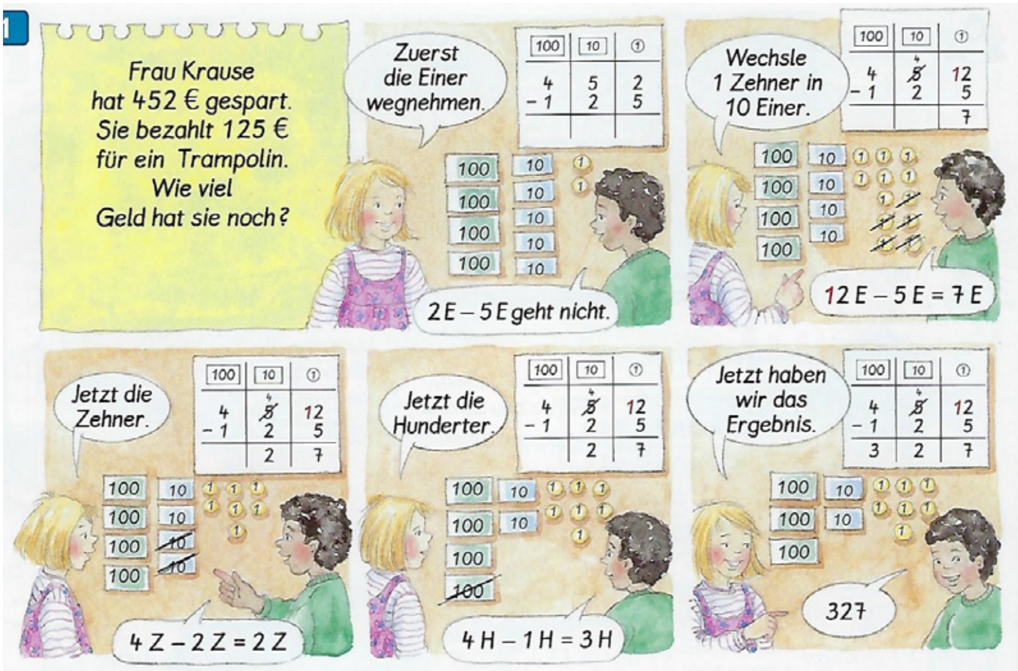


Fig. 2: Introduction of subtracting numbers using the american method in a comic strip from a maths book for the 3rd grade (Rinkens & Hönisch 2008, p. 98)

Due to the possibility of having several meaningful readings, different meaning-bearing structures can be identified within the output. For example, the encoding of subtraction in MathML presentation markup follows the structure of the spatial arrangement of the elements and thus the typesetting view of the calculation scheme (from https://www.w3.org/TR/MathML/chapter3.html#presm.addsub, sec. 3.6.8.1):

```
<mstack>
  <mscarries crossout='updiagonalstrike'>
    <mn>2</mn>  <mn>12</mn>  <mscarry crossout='none'> <none/> </mscarry>
  </mscarries>
  <mn>2,327</mn>
  <msrow> <mo>-</mo> <mn> 1,156</mn> </msrow>
  <msline/>
  <mn>1,171</mn>
</mstack>
```

However, if macros are defined at all and if mathematical typesetting programs and markup languages do not use such generic markup for the encoding, they usually are not able to represent all relevant structures. Against this background, a focus on the generation process appears to be a certainly relevant choice, but ultimately an arbitrary one.

1 In MathML, different directions for crossing out digits can be specified (https://www.w3.org/TR/MathML/chapter3.html#presm.addsub, sec. 3.6.6.2 Attributes). In the example below, a slash ("updiagonalstrike") is used.

Typically, uniform procedures are used within specific groups of actors. The advantages resulting from such standardization are pointed out, for example, when school curricula are established. The selection does not always fall on procedures that are considered useful due to didactic considerations, but they can also be chosen as a continuation of existing traditions. This makes communication across generations easier. However, the counterargument to the advantages of cultural anchoring of such procedures is the perception of international isolation, which is particularly relevant in view of global migration (Radatz et al. 1999 p. 129–132).

Both lines of argument in this discussion point to a second way of "acting" with the output of such calculation methods: it should be read. By using uniform procedures, it is possible for everyone to understand the output of such procedures and thus check the arguments of others. A typical (historical) scenario is, for example, checking a business partner's invoice, as well as understanding calculations when solving mathematical problems—be it checking calculation problems at primary school level or justifying more complex calculations at higher level education when the use of calculators is avoided due to exam regulations.

If it were just about doing calculations which has certainly become less relevant with the development of calculators, one could forego standardization and use individual procedures and formatting. Instead, reading plays a major role when learning arithmetic in primary school (and when checking the work done by students), not least in order to be able to more precisely identify the cause of systematic errors in individual students. This is also reflected in the literature on mathematics didactics, in which correct procedures as well as typical causes for errors are discussed (e. g. Radatz et al. 1999 p. 137–140). Both topics require the representation of the typical calculation schemes in print.

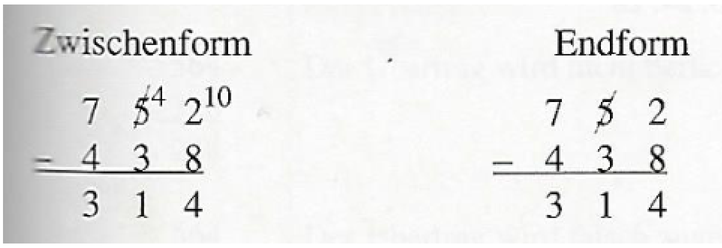


Fig. 3: Example of the use of slashed digits in a book on mathematics didactic with two different formalizations that should be used by students when they learn how to use the calculation procedure (Radatz et al. 1999 p. 137).

The history of mathematics is also interested in reading the output of calculation methods, and in particular making them readable and accessible is a central objective of scholarly editions. A typical example are texts from so-called "Rechenbücher", or notes from mathematicians which make it possible to explore the process of developing mathematical methods and ideas.

The latter in particular can have a greater variety of crossing out digits. Leibniz, for example, uses double slashes and crossing double slashes in variants of common calculation methods to mark specific intermediate steps in this modified calculation procedures.

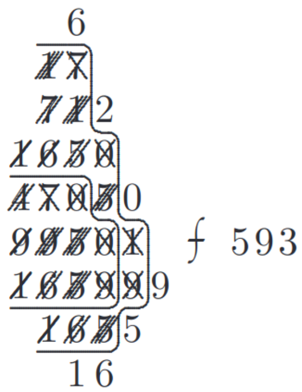


Fig. 4: Edition of Leibniz's manuscript GWLB LH 35 XII 1 f. 250 v^o^ with slashed, double slashed, triple slashed, and crosed out digits (VE VII, 8 p. 356).

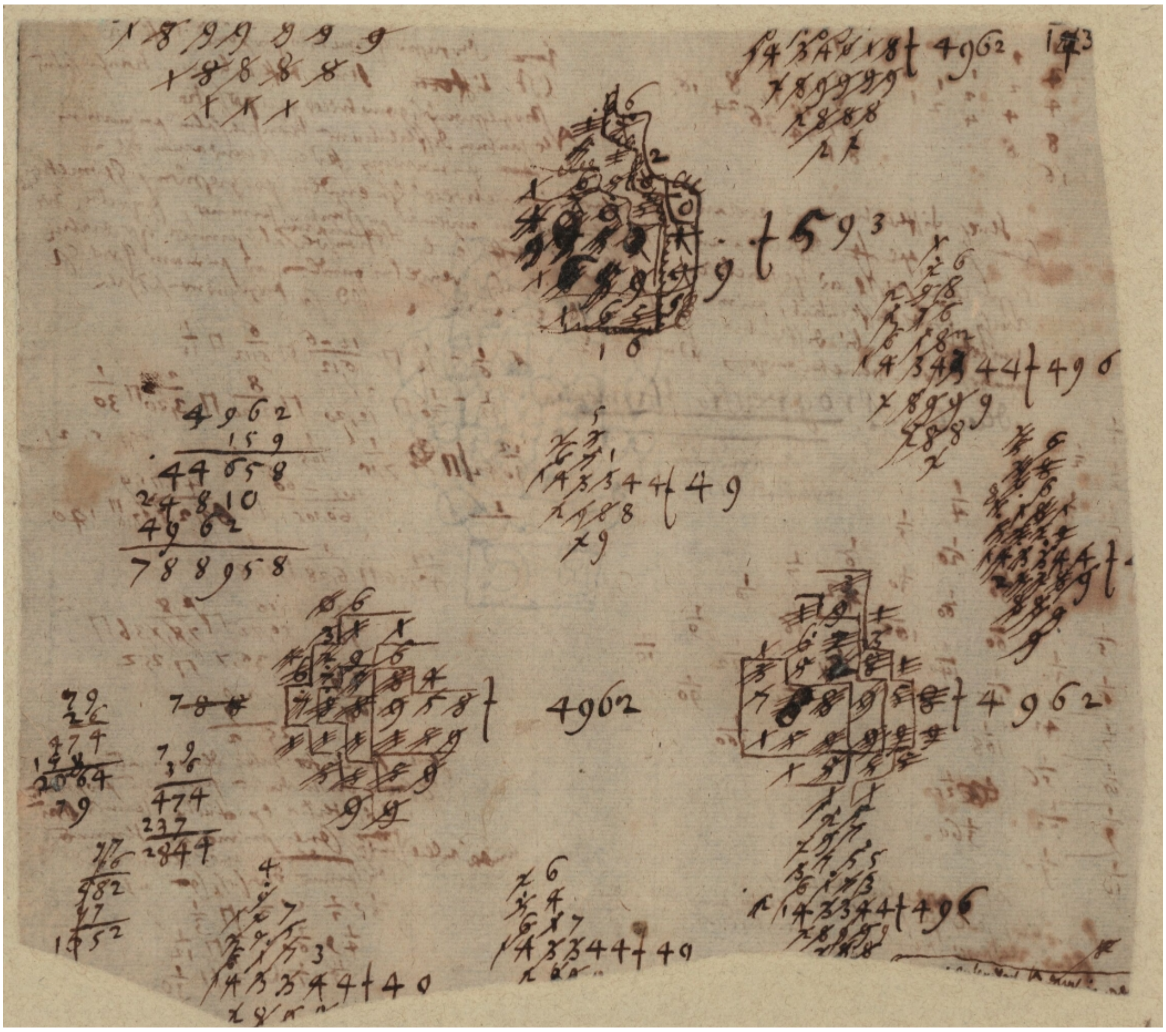


Fig. 5: Manuscript page with several examples of calculation schemes with different types of crossing out digits. GWLB LH 35 XII 1 f. 250 v^o^.

Different ways of crossing out are also used by Leibniz within methods for checking calculations such as the "Neunerprobe" ("rule of nines").

$$
\begin{array}{llllllllllll}
\text{fiet} \left\{ \begin{array}{l} \\ \\ \end{array} \right. & 1z^6 & - & 10z^5 & + & 35z^4 & - & 50z^3 & + & 24z^2 & & \\
 & & - & 5\ldots\ldots & + & 50\ldots\ldots & - & 175\ldots & + & 250 & - & 120z \\
\hline
\text{seu} & 1z^6 & - & 15z^5 & + & 85z^4 & - & 225z^3 & + & 274z^2 & - & 120z \\
 & \not 1 & & \not 3 & & \not 4 & & 0 & & \not 4 & & \not 6
\end{array}
$$

Fig. 6: Edition of Leibniz's manuscript LH 35 III A 34 f. 1 v°; with slashed digits (VE VII, 8 p. 214).

If the coefficients in the example do not sum up to a number which is divisible by 9, it is proven that an error has occurred during calculation. For this, Leibniz first writes down the rest when dividing each coefficient by nine. In the second step, he marks two different groups of rests, each summing up to nine. By using different directions for crossing out those digits, it becomes easier to understand the procedure afterwards. That means that the specific way of crossing is a relevant aspect of his writings.

All in all, even though crossing out is a typical result of calculation procedures, the output of such procedures are meant to being read. Moreover, modern prints or other representations in digital form usually only aim on reading as their only use case. Due to this, it seems highly justified to pay attention to a good legibility of slashed digits.

3. Hard-coding of slashed digits vs. sequence encoding

As an alternative to hard-coding five sets of slashed digits a possible encoding scheme as combining sequences shall be discussed.

First, the latter way would require much less codespace. It would make 5 new characters neccessary (single digit-slash, double digit-slash, triple digit-slash, reverse digit-slash, digit-crossing). Whereas a full encoding scheme of five sets of digits 0–9 would require 50 code points.

A utilization of the existing character *combining long solidus overlay* (0338) could not be advised, because even the single digit-slash needs to be shaped and measured as to exactly fit the dimensions of the figures it is supposed to be combined with. Moreover, the set of these five markings need to be designed *as a set* consistent in style and dimensions. Technically, the strategy of sequence encoding would be lean and in line with Unicode's policy to encode component-based characters as sequences.

On the other hand, the proposed hard-coding of these characters has also practical advantages. These advantages include easy finding and identification of characters with unique semantic value, a much more safe and convenient handling in auto-recognition and search processings, and, as far as the human user is concerned, a much better possibilty to reproduce these characters in a well-legible way and to avoid misreadings and wrong interpretations.
This is what the next chapter is about.

4. Visual recognition aspects

A character encoding has not only to fulfill theoretical concepts and technical requirements, it has also to enable a legible representation of text for the human eye, on any sort of output device. This aspect may be of lesser interest in many cases, but with the set of dashed digits it poses a range of peculiar issues, as we will subsequently demonstrate.

We are going to compare:

a) individually designed glyphs based on seperate codepoints for each character;

b) combined sequences of the ordinary digits (0030 to 0039) and non-spacing combining overlay characters.

First we look at a specimen in which the digits 0 to 9 (tabular figures, equal width) have been used for slashed digits by combining them with combining mark glyphs made specifically for that purpose. In order to establish the result of this approach, a test font has been created, with ordinary digits and the special slash marks as non-spacing characters.

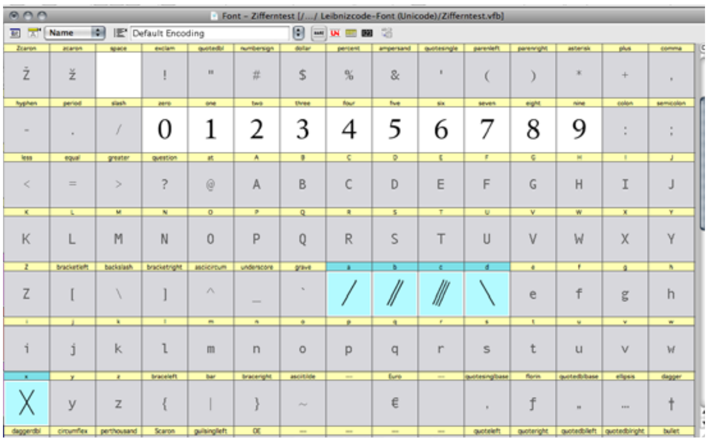
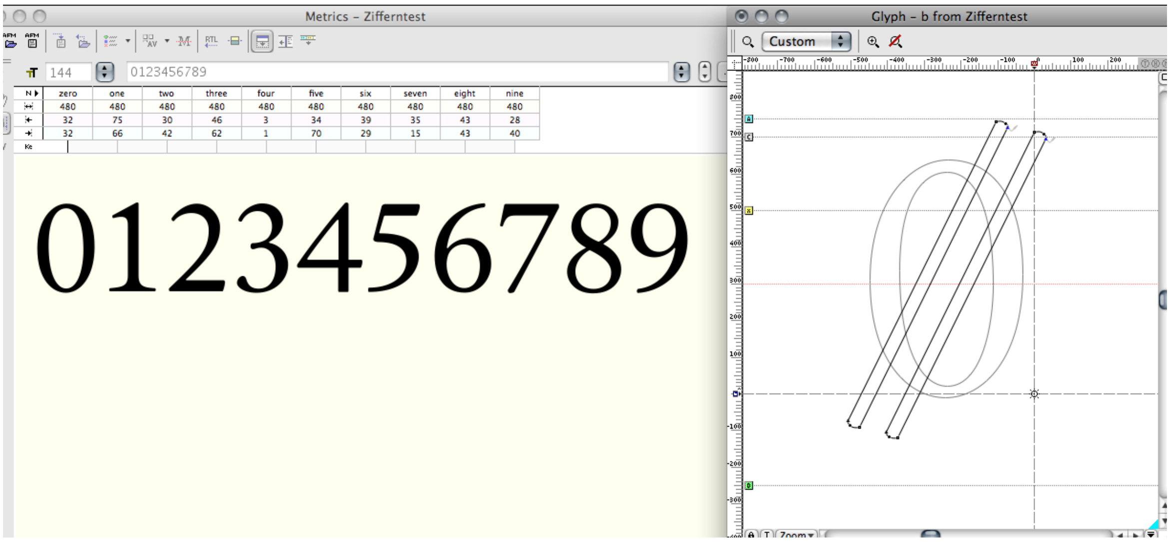


Fig.7:
A simple test font has been created for testing combined sequences. The five digit-slash mark glyphs are highlighted.

Fig 8:
The metrics of the digits follow the tabular-figures convention: all 10 characters have the same width (left). The combining glyphs have beed dimensioned and adjusted to fit the glyph of zero in the first place (right).



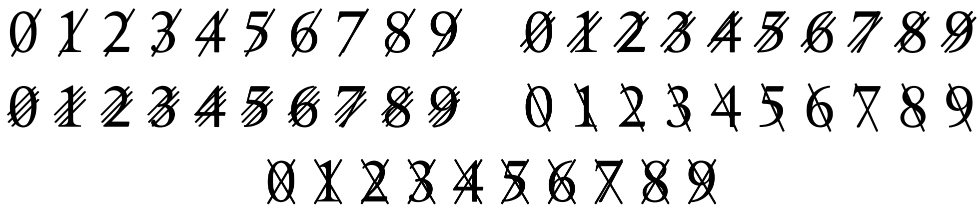


Fig. 9: Slashed digits, set via combining sequences, using the aforementioned test font.

In this test we see that the visual result is more or less satisfactory in many instances, but we also see a few problematic cases, and a few grave issues. The most difficult results occur with the 7, but also the 2, the 4 and the 8 produce results whith very bad or even impossible legibility. Especially the oblique parts of 2, 4 and 7 happen to coincide with the geometry of the oblique dash lines in such a way that the distinction mark gets neutralized and the depiction of the intended semantic fails. – This approach will lead to misreadings and interpretation errors.

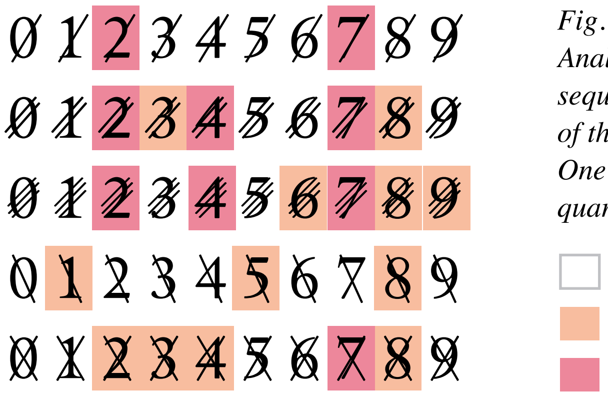


Fig. 10:
Analysis of the visual performance of the sequence combinations. Only about the half of the resulting glyphs is sufficiently legible. One quarter is poor or problematic, the 4th quarter is practically unreadable.

good or satisfactory legibility

problematic legibility

very bad or no legibility

The danger of reading errors increases dramatically when we consider typical font sizes for texts and mediocre visual conditions, such as low resolution display, diminished light or a recipient person with impared eye-sight.

We now take a look at the combined sequence digits at 12 points text size:

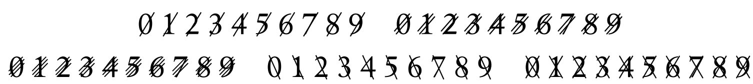


Fig. 11: Slashed digits, set via combining sequences, set at 12p

Depending on the quality of printing, screen display or eye-sight, a more or less large amount of those characters become unrecognisable. This shall be demonstratedby the result of a blurr-test which simulates a viewing under less-than-optimal visual conditions:

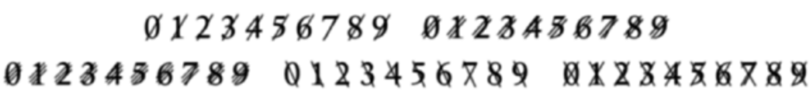


Fig. 12: Combining sequences, Blurr test

The conclusion of this test is: the rendering of these characters is insufficient for a reliable visual representation, it will lead to reading errors and therefore is unsuitable.

The assignment of seperate single codepoints allows each of the characters getting treated in a font design individually and with attendance to the neccessary adjustments in specific situations.

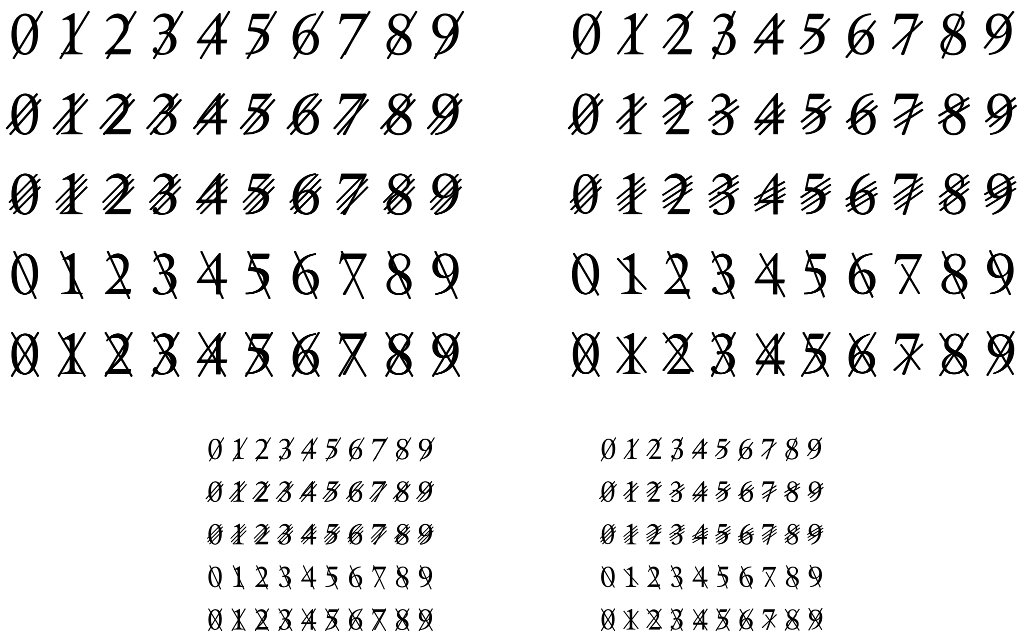


Fig. 13: Comparison between combined sequences (left) and individual glyphs/characters (right).

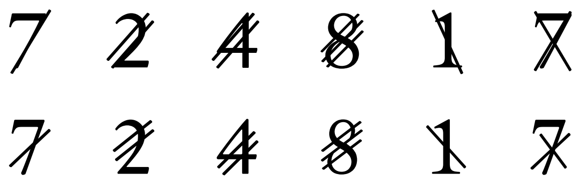


Fig. 14: A comparison of selected characters; combined (top), individual glyphs (bottom).

References

Radatz et al. 1999 — Hendrik Radatz, Wilhelm Schipper, Rotraut Dröge, and Astrid Ebeling: *Handbuch für den Mathematikunterricht. 3. Schuljahr.* Schroedel Verlag. Hanover 1999.

Rinkens & Hönisch 2008 — Hans-Dieter Rinkens and Kurt Hönisch (ed.): *Welt der Zahl 3 (Bayern).* Schroedel Verlag. Hanover 2008.

VE VII, 8 — *Vorausedition zur Leibniz-Akademie-Ausgabe, Band VII, 8: Varia mathematica, Nachträge 1670–1676. Version 4.* Prepared by Alexandra Lewendoski, Siegmund Probst, Elisabeth Rinner, Regina Stuber, and Achim Trunk, ed. by Leibniz-Forschungsstelle Hannover der Niedersächsischen Akademie der Wissenschaften zu Göttingen beim Leibniz-Archiv der Gottfried Wilhelm Leibniz Bibliothek – Niedersächsische Landesbibliothek. Hanover, 4 October 2023.