Latin challenges

Luigi Scarso

Abstract

A history of Latin since 7000 BC until today is presented, along with some code to typeset Old Italic documents. The paper shows how the application of features written on-the-fly can solve some problems of representation and content rendering.

Sommario

L'articolo offre una concisa storia della lingua latina dal 7000 a.C. ai giorni nostri. Viene discusso un esempio di documento in Old Italic che evidenzia la relazione tra presentazione e contenuto e come l'applicazione di caratteristiche appositamente definite possano risolvere alcuni problemi di tale relazione.

1 Introduction

Patterns, a fundamental part of a typesetting engine, are shipped with the ConTEXt distribution and kept in sync with the TEXLive, and each new pattern—even an update of an existing one—is first checked by experts to verify the correctness and the backward compatibility.

It is not surprising then that the new version of the Latin language patterns by C. Beccari created a bit of bustle, especially because there is a new set of patterns for classical Latin—that ended in ConTEXt as a new tag `ala`, while the previous tag `la` is now for medieval and modern Latin—which triggered some questions about Latin, its history and its "vitality". These questions raised interesting connections with (or better, challenges for) the contemporary TEX typography.

The next section outlines the history of Latin, highlighting some aspects that will be developed later.

2 A short history of Latin

Proto-Indo-European (7000–3000 BC)
A hypothesis widely accepted by linguists (the *Kurgan hypothesis*, (WIKIPEDIA, 2015A)) says that a lot of the current and old languagesof a region ranging from Europe to the Indian subcontinent, across Iran and Afghanistan, have a common ancestor: a language (designed as *proto-indo-european* or PIE) spoken around 7000 BC in the Pontic-Caspian steppe. The hypothesis describes 3 waves of expansion from the origin (*Urheimat*). These waves are marked out by 3 stages of evolution of the language itself: early PIE (4500–4000 BC), middle PIE (3500–3000 BC) and late PIE (3000–2500 BC). After the last wave, the spoken language disappeared as a single entity. Due to these waves, the language reached modern Romania, Bulgaria, eastern Hungary and Georgia.

Proto-Italic(?) (3000–1000 BC)
During the III millennium BC, waves from Central Europe propagate to South-West and South-East, very likely reaching modern Italy from North-East Alps around the 2nd millennium BC. Some scholars *reconstructed* a sub-family of late PIE called *Proto-Italic* (BAKKUM, 2009) as the common ancestor of some important subsequent languages of the Peninsula (e.g., Sabellic and Latino-Faliscan) but, since there are no textual sources, Proto-Italic is currently not tied to a specific Italic region and could have originated outside the Peninsula. Archaeological findings also unveiled another language, spoken in a region corresponding to the modern Veneto region, (PETRINI, 2014) which could be a direct descendant of late PIE (some scholars talk of *Italic-Venetic*) and a non-PIE language (*Etruscan*) in modern Tuscany, western Umbria and northern Latium.

Italic (1000–550 BC)
Around 7th century BC there were at least two languages descending from Proto-Italic—Sabellic and Latino-Faliscan (SLOCUM, 2014). Other classifications see Venetic as another Italic descendant, but this hypothesis is not yet widely accepted. Two other languages, Aequian and Vestinian, are also descendants of *Italic*, nevertheless there are too few findings to state this with certainty. This period marks the transition between prehistoric and documented phase, and it is the period of the Rome foundation, supposedly around 754/753 BC. Fig. 1 depicts the situation around 600 BC, when Umbrian and Oscan were part of the Sabellic group: it is evident the relevance of Greek and Etruscan and the presence of another late PIE descendant (*Messapic*) unrelated to all of the other languages.

The origins of Latin are influenced by several languages, and this is evident in the ancient shrine, known as *Lapis Niger* (Black Stone), where there is one of the oldest Latin inscriptions (fig. 2, 575–550 BC).

The lettering is very close to Greek letters and the inscription is written boustrophedon;it is cur-

FIGURE 1: Italic, 600 BC.

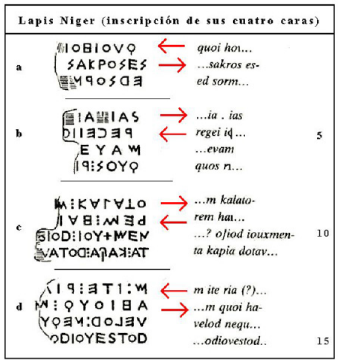

FIGURE 2: Lapis Niger, inscription, 575–550 BC.

rently the oldest Italic inscription that exhibits the word *sakros*, i.e., *sacred*. Very likely the Roman alphabet was a modification of the Etruscan one, which in turn could be derived from the Greek alphabet of the Comae colony (in the modern province of Naples). This is an Etruscan alphabet (*Marsiliana* tablet, 700 BC), with the transliteration in modern Greek:

𐌀 𐌁 𐌂 𐌃 𐌄 𐌅 𐌆 𐌇 𐌉 𐌊 𐌋 𐌌 𐌍 𐌏 𐌐 𐌑 𐌓 𐌔 𐌕 𐌖 𐌗 𐌘 𐌙
Α Β Γ Δ Ε Ϝ Ζ Η Θ Ι Κ Λ Μ Ν * Ο Ρ * Ϙ Ρ Σ Τ Υ Ξ Φ Χ

Asterisks are just placeholders for missing Greek characters; the Etruscan ones correspond to the English sound `sh'. The `H' Etruscan glyph corresponds to the English sound of `h', not to the long Greek vowel `H'.

That period saw the rise of Rome, that ended with the domination of Latin over all of the other languages of the Peninsula. Following (MONDIN, 2012) the periods are the following ones.

Pre-Literary Latin (550–241 BC)

The textual documentation is limited but shows the alphabet gradually moving from Etruscan to Roman (SAMPSON, 1985): this is the alphabet used in Etruria around 400 BC

𐌀 𐌂 𐌃 𐌄 𐌅 𐌆 𐌇 𐌉 𐌊 𐌋 𐌌 𐌍 𐌏 𐌐 𐌑 𐌓 𐌔 𐌕 𐌖 𐌗

and this is the alphabet used by Romans during the same time:

A B C D E F Z H I K L M N O P Q R S T V X

For example, the Ficoroni cista (fig. 3) already shows a clear Roman alphabet (DANIELS AND BRIGHT, 1996), even though the "A" is still Italic. The text says: *Novios Plautios med Romai fecid. Dindia Macolnia fileai dedit* and it's one of the first inscriptions with the word "Rome".

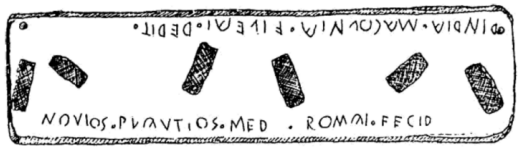

FIGURE 3: Ficoroni cista, latin inscription (350 / 325 BC).

Archaic Latin (241–79 BC)

The textual documents became more relevant and the alphabet started to get stability: in the 3rd century BC it is

A B C D E F G H I K L M N O P Q R S T V X

but already around 1st century BC the letters (called *Square Capitals*) were:

A B C D E F G H I K L M N O P Q R S T V X Y Z

Classical Latin (79 BC -14 AC)

This is the well-known period of Marcus Tullius Cicero, Gaius Julius Caesar, Gaius Sallustius Crispus, Gaius Valerius Catullus (Republican period) and Publius Vergilius Maro (Virgil), Quintus Horatius Flaccus (Horace), Publius Ovidius Naso (Ovid), Titus Livius (Augustan period). The alphabet evolved considerably: in addiction to a formal one used for official documents and inscriptions there is an informal one, the *Rustic Capitals* (fig. 4, (MARCOS, 2014)):

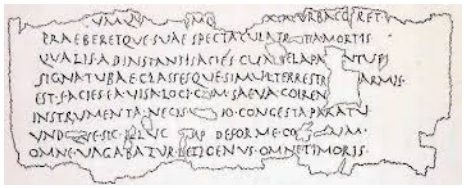

FIGURE 4: Rustic Capitals: De Bello Actico, 31 BC - 79 AD.

In this period is also proved the existence of an Old Roman Cursive alphabet (fig. 6, (MARCOS, 2014)) where the letters were still uppercase but with some glyphs very similar to the lowercase counterpart (e.g. the "D" in fig. 7).

Post-classical Latin (14 AD - 193 AD)

This is the period of the Pax Romana and, thanks to Trajan, of the greatest military expansion. The Square Capital used for his inscription (fig. 5) is considered a canonical form. Latin was the official language of the Roman empire and it was increasingly used for administrative and commercial tasks. For this reasons, it started to spread

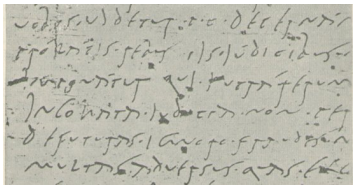

Figure 6: Papyrus
Claudius, 41 - 54 AD.

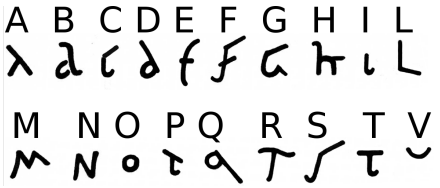

Figure 7: Old Roman Cursive.

outside the élite. As a consequence, the separation between formal/written literary Latin and informal spoken Latin was already observable at the end of this period.

Figure 7: Old Roman Cursive.

Late Latin (193 AC - 476 AD)
The distance between formal and "pure" Latin and *low* Latin increased. A new cursive (*New Roman Cursive* (Marcos, 2014), fig. 8) appeared, and it was consistently used as minuscule also in ordinary documents. The end of the 2nd century saw the rise of the Christian religion, initially spread among the poor but later (edit of Milan, 313 AD) officially admitted and even promoted as a way to keep the unity of the empire; *catholicism* comes from the Greek term καθολικισμός (katholikismos, i.e., "universal") and it has been used for the first time in The Epistles of St. Ignatius (108 AD, in Greek) to assert the role of a unique Church for the *entire* mankind in contrast with several different local Churches (Smyrna in this case), but the emperors pushed instead the use of Catholicism as universal, i.e., mandatory, for all the empire.

This had two opposite consequences: on one side, it enforced the role of the spoken Latin that will later die as a living language originating the Continental Romance; on the other side, the adoption of Classical Latin in its new role of supporting the emperor saved the language from extinction. Greek language had a similar evolution: initially most part of the Christians documents (included the Old and New Testament) were written in Greek, and in the first seven councils (323–787 AD) Greek remained the reference language for the dogmatic definitions.

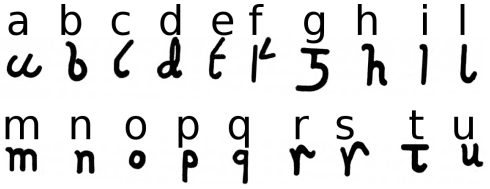

Figure 8: New Roman Cursive.

The importance of the period is attested by another alphabet that started to gain consensus especially among Christian authors: the *Uncial*, probably derived both from New Roman Cursive and Greek (the Codex Vaticanus, 325–350 AD is a Greek Bible) but increasingly used also for Latin, as in the Codex Victor (Steffens, 2015) in fig. 9.

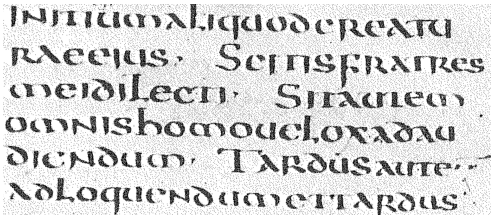

Figure 9: Codex Victor, Uncial, 546 AD.

Uncial (from Latin *uncia*, "one twelfth") is a kind of transitional font: it's a majuscule (i.e. upper case) alphabet with some minuscule letters, such as "p", "q" and "h". It also acts as a "bridge" between the paganism to reject and evangelism to pursue: the New Roman Cursive is too informal for the Christian literature and the Rustic Capital and Square Capital are both too tied to the old world, while the Uncial is more elegant and blends well initially with Greek and with Latin alphabet later. While this period ended with the Fall of Rome, it did not see the fall of Latin as a language because it kept being used as *the* language of literature and religion.

Medieval Latin (476 AD–first half 14th century)
This was the last period of Latin as a living language. The crisis of the Western Empire left the Christian Church as the strongest authority, but its initial vocation to universality was toward all the men and not military based, even though the 7th century saw the birth of the Papal State as a political entity. As a consequence of the absence of a strong political authority, Latin of that time was influenced by spoken Latin and by languages like Germanic and Gallic. The invasion of the Iberian Peninsula in 711 AD by Arabs, as well as their occupation of Sicily between 831 AD and 1072, was another source of influence because it broke

for the first time the Christian Church universal role. It is well known that the terms algorithm (al-Khwārizmī) and algebra (al-ğabr) were caught from the author Muhammad ibn Mūsā al-Khwārizmī and his book *Kitāb al-muḫtaṣar fī ḥisāb al-ğabr wa-l-muqābala* (830 AD).

To set a date, the Council of Tours in 813 AD decided that priests should preach sermons in *rusticam romanam linguam* (rustic romance language) or Theodiscam (German) (WIKIPEDIA, 2016A) and one of the first known official documents, the Oaths of Strasbourg (842 AD), is written in three different languages: Medieval Latin, Old French and Old High German. The Old French passages are generally considered to be the earliest texts in a language that is distinctly French (WIKIPEDIA, 2015B). The 9th century can be hence considered as the beginning of the end of Latin as a living language.

Renaissance Latin (14th–15th century)
During the European Renaissance, a strong cultural movement (*humanism*) enforced the return to the origins (*ad fontes*) of Classical Latin. As a result, from that time on *Renaissance Latin* (or humanist Latin) became the standard. The Papal State is a defined political and geographical entity and Latin is the official language.

Neo-Latin (15th–20th century)
Humanist Latin spread in almost all fields of education (in 1687 Isaac Newton published the first edition of *Philosophiæ Naturalis Principia Mathematica* in humanistic Latin) and also diplomacy made an extensive use of Latin as international language. Catholic Roman Church kept using Latin but Martin Luther, between 1522 and 1534, published the *Luther Bible* in German, marking the beginning of the end of Latin as the only official language in Christianity. From 1700 to 1900, Latin gradually disappeared in Europe from scientific literature, laws and diplomacy (the last international treaty to be written in Latin is the Treaty of Vienna, written in 1738). The Papal State as an entity ceased to exist in October 1870, after a short conflict with the kingdom of Italy.

It's worth noting that up to the end of Middle Ages there are about 600 written texts, but afterwards their number rises to 18 000. The Gutenberg printing revolution played a fundamental role, marking the start of traditional typography.

Contemporary Latin (20th century–today)
By the mid of the 20th century, Latin was gradually dropped out of the core instruction. Humanistic Latin used by Catholic Roman Church slowly evolved in *Ecclesiastical Latin*, practically a minor variation of Classical Latin. The Lateran Treaty with Italy in 1929 created the State of the Vatican City in Rome, with Italian as the official language. The State hosts the Holy See, i.e., the episcopal see of the Pope, representing his authority over the Catholic Church. The official language of the Holy See is Ecclesiastical Latin. Latin is used in theological works, liturgical rites, dogmatic proclamations and documents of Holy See. The Second Vatican Council (1962–1965) made Latin alternative, no longer the exclusive language of liturgies. Despite that, the definition of Latin as a *dead language* appears incorrect, because it's still in use for administrative tasks as shown in fig. 10.

FIGURE 10: ATM in Vatican, 2008.

In the final part of 20th century, the development of computer technologies marked the transitions from traditional to digital typography and, as a consequence, a renewed interest for the Latin documents as a source of best characters (as the Trajan typeface from Carol Twombly for Adobe, 1989). Unicode Consortium in 1991 published the first volume of UNICODE, "a computing industry standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems" (WIKIPEDIA, 2016B). In this book, thanks to the large amount of documents, Latin glyphs are adequately represented. For Old Italic, on the contrary, the situation is quite different.

3 Some challenges for contemporary typography

Contemporary typography is almost completely digital typography and as such it cannot be separated from the type of digital support. In this paper, the support is the PDF file format, subject to two constraints: 1) the font is a "good" (i.e., well designed and with large number of glyphs) OPENTYPE and 2) the content derived from the PDF has to be UNICODE-compliant (i.e., "copy-and-paste" operations must preserve characters). These are quite reasonable requests for modern languages, but what about the Lapis Niger shown in fig. 2?

UNICODE supports Old Italic from 2000, so the first step is to find a good OPENTYPE font; AEGEAN (DOUROS, 2015) meets the requirements. Installing the font in ConTEXt is a matter of co-

pying the files into a suitable location such as, for example, `tex/texmf-fonts` and issue a `mtxrun --generate` to update the cache. Then

```
\definefontsynonym[aegeanhint][file:Aegean.ttf]
\definefontsynonym[ArchaicLatin][aegeanhint]
```

after that, the font is available. Just make ConTEXt use it with

```
{\definedfont[ArchaicLatin at 12pt]  A B }
```

(ConTEXt widely uses layered declarations to decouple the real file name and the abstract font name). The text of the Lapis Niger is written boustrophedon, as the red arrows show. It's then natural to define a similar set of macros:

```
\def\←#1{{\righttoleft #1}}
\def\m←#1{\bgroup% right to left & mirror
\setbox\nextbox\hbox{#1}%
\hbox to\hsize{\hss\vbox{\hsize=\wd\nextbox%
\righttoleft\mirror{\unhbox\nextbox}}}%
\egroup}
\def\→#1{\hbox to 0.5\textwidth{\hss{\lefttoright #1}}}
\def\↶#1{\rotate[rotation=180,location=high]{#1}}
\def\⇆#1{\mirror{#1}}
```

so that typesetting is straight:

```
  \definedfont[ArchaicLatin at 9.8pt]
\←{𐌒 𐌖 𐌏 𐌉 𐌇 𐌏 𐌉}
\→{𐌔 𐌀 𐌊 𐌓 𐌏 𐌔 𐌄 𐌔}
\m←{𐌄 𐌃 \⇆𐌔 𐌏 𐌓 𐌌}
\→{𐌆 𐌀  𐌆 𐌀 𐌔}
\m←{𐌓 𐌄 𐌂 𐌄 𐌉 𐌉 𐌂}
\→{𐌄 𐌖 𐌀 𐌌 }
\m←{𐌒 𐌖 𐌏 \⇆𐌔 ⁝ 𐌓 𐌉}
\→{\⇆{\↶𐌌} ⁝ 𐌊 \↶𐌀 \⇆{\↶𐌋} \↶𐌀 \↶𐌕 𐌏}
\m←{\↶{\⇆{𐌓}} 𐌄 \⇆{\↶𐌌} ⁝ 𐌇 𐌀 𐌉}
\→{𐌏 𐌉 𐌏 𐌃 ⁝ 𐌉 𐌒 𐌖 𐌗 𐌌 𐌄 𐌍}
\←{𐌕 𐌀 : 𐌊 𐌀 \⇆𐌐 𐌉 𐌀 : \⇆𐌃 𐌏 𐌕 𐌀 𐌖}
\m←{𐌌 : 𐌉 𐌕 𐌄 : 𐌓 𐌉 𐌀}
\→{𐌌 : 𐌒 𐌖 𐌏 𐌉 𐌇 𐌀}
\m←{𐌖 𐌄 𐌋 𐌏 𐌃 : 𐌍 𐌄 𐌒 𐌖 }
\→{𐌏 𐌃 𐌉 𐌏 𐌖 𐌄 𐌔 𐌕 𐌏 𐌃}
```

(as a side note, the previous macros implement a simple notation that helps discovering which characters are transformed).

The result is in fig. 13 and it can be verified that the representation matches the source inscription. The next step is to verify if the content matches the representation, but unfortunately this is not always true: for example, in the fifth row from the end (`ta kapia dotav`), the text copied is

𐌖 𐌀 𐌕 𐌏 𐌃 : 𐌀 𐌉 𐌐 𐌀 𐌊 : 𐌀 𐌕

where 𐌐 and 𐌃 are not mirrored. This is a limit of Unicode: there is no modifier for this kind of transformations and the only solution is to add a codepoint to the standard as, for example, (Everson, 2016) proposed for a similar problem.

Following this example, a question arises: what does it happen if the image of a glyph is dubious and there are not alternative fonts? One answer is to create a font ad-hoc, i.e., with few glyphs, but this is not always the right answer. The design of an OpenType font is a difficult and time-consuming task and a low number of glyphs usually means low quality. The alternative METAFONT produces Type 3 fonts, which is excluded because it is not OpenType, and bitmaps in OpenType are not widely used. The solution this paper proposes is a mix: define a new font feature on-the-fly to substitute just the needed glyphs with a bitmap image previously generated with METAFONT (it is basically an extension of the virtual font mechanism already used in TEX).

The code starts registering the features `mfgl` and `mfgln` (names are arbitrary) for bitmap glyphs scaled / not scaled to the current fontsize:

```
\startluacode
fonts.handlers.otf.features.register {
    name        = "mfgl",
    description  = "Add some metafont bitmap glyphs,
scaled to the current size.",
    manipulators = {
        base = addmfgl,
        node = addmfgl,
    }
}
fonts.handlers.otf.features.register {
    name        = "mfgln",
    description  = "add some metafont bitmap glyphs, not scaled",
    manipulators = {
        base = addmfgl,
        node = addmfgl,
    }
}
\stopluacode
```

Both the features inherit from the `default` feature and add their own manipulators:

```
\definefontfeature[mfgl][default][mfgl=true]
\definefontfeature[mfgln][default][mfgln=true]
```

The features can be associated to a font by means of `\definefontsynonym` or directly with `\definedfont`:

```
{\definedfont[Aegean.ttf*mfgl at 12pt] this font has the feature
mfgl}
{\definedfont[Aegean.ttf*mfgln at 12pt]this font has the feature
mfgln}
```

Fig. 14 shows the Lua code implementing the features; it is not hard to explain the code: it takes the bitmap images already stored in the table `MFglyphimage`, which is indexed with Unicode codes, and replaces each glyph of the original font

with the related bitmap image. This is possible because the PDF has a glyph effect that does not draw the character and LuaTEX can inject this PDF sequence as a part of a more complex command (please refer to the section *Virtual fonts* of the LuaTEX manual for a detailed explanation). The net effect is a replacement of the glyphs keeping the ToUnicode mapping untouched, i.e., preserving the content.

The final step is to show how to produce a bitmap with METAFONT. First, it is important to recall that a pk font file can be translated into a hexadecimal bitmap with the program `pk2bm`, obtaining an ASCII file as the following (for the sake of simplicity most of the rows are omitted):

```
character : 65 (A)
   height : 351
    width : 203
     xoff : -15
     yoff : 327

  00000000000000000000000000000ff8000000000000000000
  00000000000000000000000000003fff0000000000000000000
  00000000000000000000000000000ffffc000000000000000000
  00000000000000000000000000003fffffe000000000000000000
  00000000000000000000000000007fffff0000000000000000000
  0000000000000000000000000000ffffff8000000000000000000
  0000000000000000000000000001ffffffc000000000000000000
  0000000000000000000000000001ffffffe000000000000000000
:
  0ffffff0000000000000000000000000000000000000000000000
  07ffffe0000000000000000000000000000000000000000000000
  01fffffc000000000000000000000000000000000000000000000
  00ffff0000000000000000000000000000000000000000000000
  003ffc000000000000000000000000000000000000000000000
```

The first step is hence to store these files into a LUA table with the right UNICODE index. Next, every hexadecimal glyph is translated into a PDF image object that uses the stencil masking, so that black pixels are transferred into the page, while the white ones (the background) are left unchanged. These images are then stored into the `MFglyphimage` global table (fig. 15). The result is shown in fig. 11: The first step is hence to store these files into a LUA table with the right UNICODE index. Next, every hexadecimal glyph.

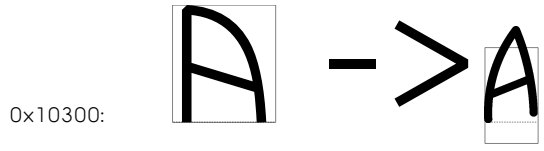

FIGURE 11: Original (left) and bitmap (right) U+10300 glyph.

The first step is hence to store these files into a LUA table with the right UNICODE index. Next, every hexadecimal glyph.

Copy-and-paste is also preserved: 0x10300: 𐌀 ->𐌀
and the fonts are

```
$> pdffonts sample.pdf
name                                 type
------------------------------------ ------------
XNQYLW+Aegean                        CID TrueType
```

so there is no Type 3 font in the PDF.

Instead of bitmap images it is possible to use MetaPost code, but some workflows use METAFONT as an intermediate step in the production of OpenType. In this case, the bitmap glyph eventually can be later replaced with a vectorial version.

The last challenge is the Ecclesiastical Latin character *œ́*, which is absent from the UNICODE repertoire. It's quite a rare character, in use at least from 1894 (CHURCH, 1894) and the correct way to obtain it is by combining U+0153 and U+0301 (which is the combining acute accent). This preserves the content but the representation can be ugly as is in the case of Latin Modern: the accent is not correctly positioned. This is really a font problem, but it is possible to define an on-the-fly feature as seen before:

```
\startluacode
    fonts.handlers.otf.addfeature {
        name = "ptest",
        type = "pair",
        data = {
            [0x0153] = { [0x0301] = { false,  { -75, -40,
0, 0 } } },
        }
    }
\stopluacode
\definefontfeature[default][default][ptest=yes]
```

The word onomatopœ́ia in Linux Libertine at 12pt is shown in fig. 12: of course the patch is only valid for a specific font at a specific size, as viewable in the figure caption for Latin Modern.

onomatopœ́ia

onomatopœ́ia

FIGURE 12: onomatopœ́ia patched for the font Linux Libertine, before (top) and after (bottom).

This can appear a negligible difference, but (PER

l'Ecumenismo e il Dialogo, 1999) explains that, during Middle Ages, *pænitentia* (penance) was transcribed, due an erroneous interpretation of *æ*, as *pœnitentia* (the italic font well shows how it is possible to make such kind of mistakes), where pœna is punishment. This had an enormous influence on the Sacrament of Confession and only at the end of 20th century, also supported by these "typographical" considerations, the Sacrament gradually changed its name in Sacrament of Reconciliation, more aligned to the original meaning.

4 Conclusion

The history of Latin shows that even a very old language reserves surprises when meeting modern technologies. For Old Italic, it is clear that Unicode does a great job, but it is not yet complete for the current needs of a digital document. Such needs require every domain to have its characters completely specified so that a simple search will not fail. For Neo-Latin the situation is good, because even in case of *œ́* it's easy in ConTEXt to adjust the representation if it is suboptimal for a given font.

The author wishes to thank C. Vincoletto for his work with METAFONT on the font LapisNiger, used in section 3.

Bibliography

Links checked on 2016.02.13.

Bakkum, G.-C.-L.-M. (2009). *The Latin Dialect of the Ager Faliscus: 150 Years of Scholarship*. (no. v. 1). Amsterdam University Press.

Church, C. (1894). *Missale romanum: en decreto sacrosancti Concilii Tridentini restitutum*. URL: https://archive.org/details/missaleromanume01churgoog.

Daniels, P.-T. & Bright, W. (1996). *The World's Writing Systems*. Oxford University Press.

Douros, G. (2015). *Unicode Fonts for Ancient Scripts*. URL: http://users.teilar.gr/~g1951d/.

Everson, M. (2016). *Proposal for the addition of three Old Italic characters*. URL: http://www.unicode.org/L2/L2015/15181r-n4669r-olditalic.pdf.

Marcos, J.-J. (2014). *Fonts for Latin paleography*. URL: http://guindo.pntic.mec.es/jmag0042/LATIN_PALEOGRAPHY.pdf.

Mondin, L. (2012). *Introduzione allo studio del latino — dispensa dei corsi 2009-2010*. URL: http://lettere2.unive.it/lmondin/Introduzione allo studio del latino.

Petrini, S. (2014). *Storia, vita, costumi, religiosità dei Veneti antichi*. URL: http://www.venetoimage.com/svc.htm.

per l'Ecumenismo e il Dialogo, C.E.I.S. (1999). *Lettera di collegamento nr. 53*. URL: https://www.chiesacattolica.it/cci_new/documenti_cei/2001-02/13-25/lettera35.pdf.

Sampson, G. (1985). *Writing Systems: A Linguistic Introduction*. Stanford University Press.

Slocum, J. (2014). *Indo-European Languages — Evolution and Locale Maps*. URL: http://www.utexas.edu/cola/centers/lrc/general/IE.html.

Steffens, F. (2015). *Paléographie latine*. URL: http://www.icar.beniculturali.it/biblio/_view_volume.asp?ID_VOLUME=51.

WIKIPEDIA (2015a). *Kurgan_hypothesis*. URL: https://en.wikipedia.org/wiki/Kurgan_hypothesis.

WIKIPEDIA (2015b). *Oaths of Strasbourg*. URL: https://en.wikipedia.org/wiki/Oaths_of_Strasbourg.

WIKIPEDIA (2016a). *Council of Tours*. URL: https://en.wikipedia.org/wiki/Council_of_Tours.

WIKIPEDIA (2016b). *Unicode*. URL: https://en.wikipedia.org/wiki/Unicode.

Source of figures

- fig. 1: *Di Dbachmann, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=333677*
- fig. 2: *By http://imperioromano.com/ - http://imperioromano.com/imagenes/galeria/1174089912.jpg, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=7710498*
- fig. 6: *Original uploader was Avitus at la.wikipedia - Originally from la.wikipedia; description page is/was here., Public Domain, https://commons.wikimedia.org/w/index.php?curid=2504913*
- fig. 7: *By Osado - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=6505958*
- fig. 5: *By Photo of the inscription on Trajan's column, https://en.wikipedia.org/w/index.php?curid=7999444*
- fig. 8: *By Osado - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=6505958*
- fig. 9: *L'utente che ha caricato in origine il file è stato Vincent Ramos di Wikipedia in francese - Trasferito da fr.wikipedia su Commons., CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2237975*
- fig. 10: *Di Seth Schoen - Latin ATMUploaded by h-stt, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=10908243*
- fig. 11: *Pubblico dominio https://it.wikipedia.org/w/index.php?curid=804425*

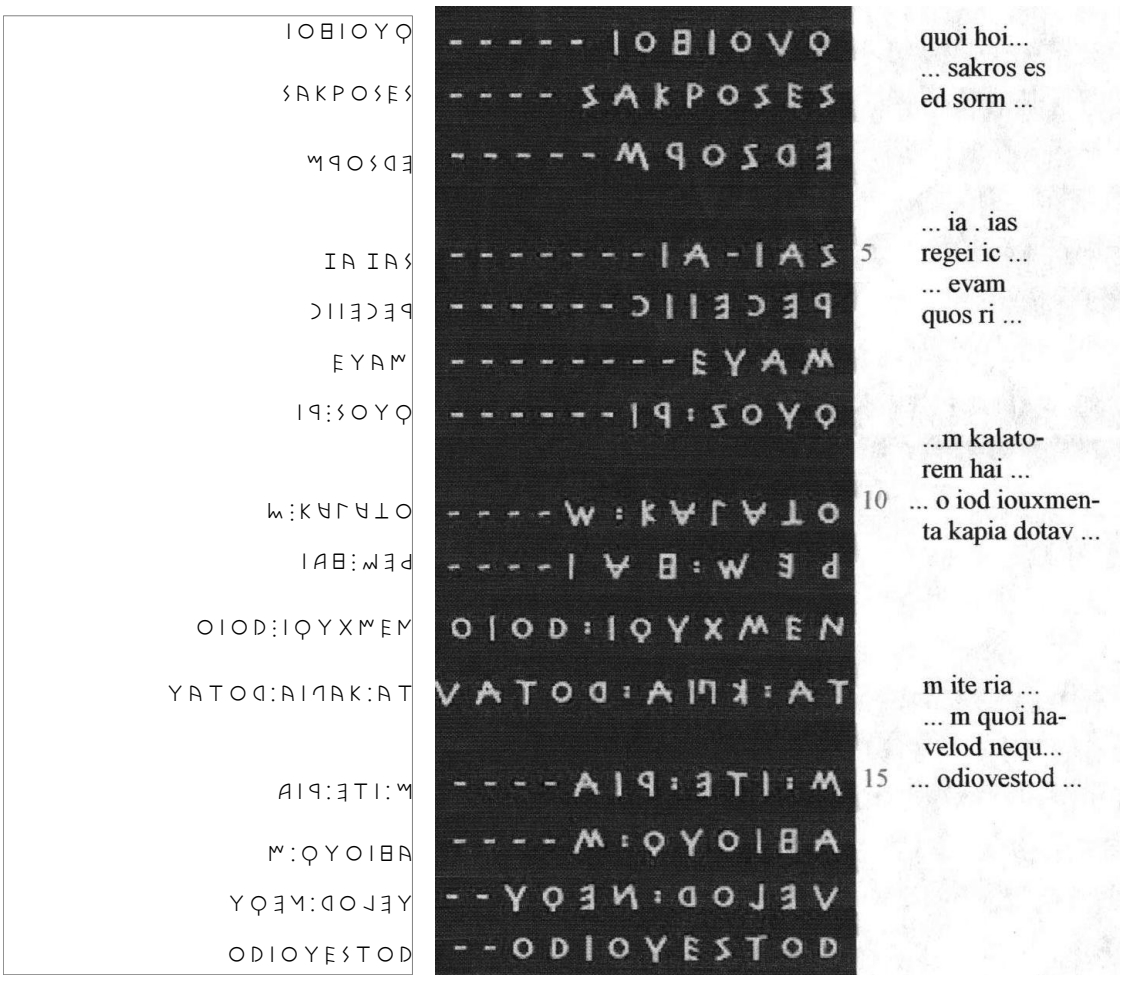

FIGURE 13: Lapis Niger typeset.

```
\startluacode
local function addmfgl(tfmdata,value)
    if value then
        local properties       = tfmdata.properties
        local parameters       = tfmdata.parameters
        local name             = tfmdata.filename
        local size             = parameters.scaledpoints
        local font, id         = fonts.constructors.readanddefine(name, size)
for targetslot,_n in pairs(MFglyphimage) do
        local slot             = targetslot  --[==[ original glyph ]==]
        local chardata         = font.characters[slot]
if not(chardata) then
  goto CONTINUE
end
        properties.virtualized = true
        tfmdata.fonts          = { { id = id } }
        local width            = chardata.width  or 0
        local height           = chardata.height or 0
        local depth            = chardata.depth or 0
        local correction       = number.dimenfactors.bp * width
        local mfglyph          = MFglyphimage[targetslot].img
        local mfglw            = 1
        local mfglh            = 1
        local mfglyph_depth    = MFglyphimage[targetslot].dim[3]
        if  mfglyph_depth == nil then mfglyph_depth = 0 end
        if value == 'mfgl' then
         if mfglyph.width ~=0 and width ~=0 then
           mfglw = width/mfglyph.width
         else
           mfglw = 0
         end
         if (mfglyph.height+mfglyph_depth) ~=0 and (height+depth) ~=0 then
           mfglh = (height+depth)/(mfglyph.height+mfglyph_depth)
         else
          mfglh = 0
         end
        end
if mfglw > mfglh then mfglw=mfglh else mfglh=mfglw end
        tfmdata.characters[targetslot] = {
            width     = MFglyphimage[targetslot].dim[1]*mfglw or 0 ,
            height    = MFglyphimage[targetslot].dim[2]*mfglw or 0 ,
            depth     = MFglyphimage[targetslot].dim[3]*mfglw or 0 ,
            commands  = {
              { "push" },{ 'special', 'pdf: q' },{ "special", 'pdf: 3 Tr' },
               { "slot", 1, targetslot },   --[==[ Original char ]==]
               { "right", -1*width }, { 'special', 'pdf: Q' }, { 'special', 'pdf: q' },
               { "lua", "pdf.print(' ".. mfglw  ..  " 0 0 " .. mfglh  .. " 0 0 cm \\n')" },
               { "down", 0*mfglyph_depth}, { "image", mfglyph}, { "down", -0*mfglyph_depth},
               { 'special', 'pdf: Q' },{ "pop" },
            }
         }
::CONTINUE::
end
end
end
```

Figure 14: The code for the features `mfgl` and `mfgln`.

```
\startluacode
local _hexstr = function(s) return string.format('%x',s) end
local _hexnr  = function(s) return tonumber(table.concat({'0x',s})) end
MFglyphimage = MFglyphimage or {}

local mffont = dofile('lapisniger-mfgl-2540.lua')
local characters = mffont.characters
local tfm  = mffont.tfm
local dpi  = mffont.dpi
for k,ch in pairs(mffont.characters) do
   if not(ch.rows) or not(ch.mfindex) then
      mffont.characters[k] = nil
   end
end
for k,ch in pairs(mffont.characters) do
   local targetslot = _hexnr(k)
   local x          = ch.rows[1]:len()*4    --[==[ 8bits <-> 2 hex chars ]==]
   local y          = ch.height
   local w          = tfm.characters[tonumber(ch.mfindex)].width
   local h          = tfm.characters[tonumber(ch.mfindex)].height
   local d          = tfm.characters[tonumber(ch.mfindex)].depth
   local rows       = table.concat(ch.rows,'')
   local basepoints = number.dimenfactors["bp"]
   local template   = "q BI %s ID %s > EI Q"
   pdfconstant = lpdf.constant
   pdfarray    = lpdf.array
   newimg      =  img.new
   x = tonumber(x)
   y = tonumber(y)
   w = tonumber(w)
   h = tonumber(h)
   d = tonumber(d)
   local dict = lpdf.dictionary {
      W   = x,
      H   = y,
      F   = pdfconstant("AHx"),
      CS  = nil,
      BPC = 1,
      IM  = true,
      D   = pdfarray({1,0}), --[==[ reverse with pdfarray({0,1}) ]==]
    }
   --[==[for some reasons it only works well if we take a 1bp boundingbox (HH)]==]
   local urx, ury = 1/basepoints, 1/basepoints
   local image = newimg {
      stream = string.formatters[template](dict(),rows),
      width  = w,
      height = h+d,
      bbox   = { 0, 0, urx, ury },
   }
   MFglyphimage[targetslot] = {}
   MFglyphimage[targetslot].img =  image
   MFglyphimage[targetslot].dim =  {w,h,d}
end
\stopluacode
```

FIGURE 15: The code for the bitmap images of features `mfgl` and `mfgln`.