Greek and Latin hyphenation – Recent advances

Claudio Beccari

Abstract

A recent package of mine revealed some unacceptable inconveniences in the hyphenation of the Greek language: *pdflatex*, that can manage only 8-bit encoded fonts, could not differentiate between the actual three varieties of Greek (the modern monotonic, the modern polytonic, and the ancient polytonic ones) and could hyphenate more or less correctly provided that the source text was input with the special Latin transliteration connected with LGR encoded fonts.

The hyphenation patterns for the three varieties had been part of any complete format file for years, but the Greek language description file used only the modern polytonic Greek patterns.

The problem was detected; the solution was found; work is in progress in order to perform a fine tuning of the new facilities.

At the same time the hyphenation patterns for standard Latin did not work well for classical Latin; the professional latinists did not work well with LaTeX and the existing hyphenation patterns. A new set of patterns was developed an a suitable set-up was created in order to switch pattern set according to the type of Latin that is being typeset.

The solutions that have been found for Greek and Latin are similar, and for this reason we deal with them in a single article.

Sommario

Mediante un mio recente pacchetto è venuto alla luce un problema relativo alla sillabazione del greco, nelle sue tre varietà, quella del greco moderno monotonico, di quello moderno politonico e di quello antico politonico, che si manifesta quando si compone il testo con *pdflatex*, il quale riconosce solo i font con codifiche di otto bit. Di fatto da diversi anni i file di formato contengono i pattern di sillabazione delle tre varietà di greco, ma il file di descrizione di questa lingua usa solo quelli per il greco moderno politonico; inoltre questi pattern funzionano correttamente solo se il testo sorgente è scritto mediate la traslitterazione latina e i font con la codifica speciale LGR.

Il problema è stato individuato; è stata trovata una soluzione; il lavoro è ancora in corso d'opera per mettere a punto i nuovi file.

Nello stesso tempo i pattern di sillabazione per il latino corrente non erano adatti per comporre testi in latino classico; i latinisti non potevano lavorare bene con LaTeX e i pattern esistenti. È stato creato un nuovo set di pattern e si è creato un tipo nuovo di impostazione per cambiare i pattern da usare a seconda del tipo di testo latino che si sta componendo.

Le soluzioni trovate per le due lingue sono simili e vengono raccolte in quest'unico articolo.

1 Introduction

1.1 Greek

The TeX system is continuously enriched with new facilities for typesetting in other languages, also those that are being typeset with glyph sets that are not part of the so called *Latin alphabet*.

Greek has been supported for some 15 years now; at the beginning there existed only one font created by Silvio Levy (Levy and Murphy, 2010) but no hyphenation patterns nor any language description file did exist.

I created an almost full collection of Computer Modern and Latin Modern compatible Greek bitmapped fonts suited for typesetting with *latex* and, later on, vector fonts for use with *pdflatex*; since both programs are only 8-bit-encoding aware typesetting ones, they could manage only fonts containing just 256 glyphs. Actually for typesetting Greek about 300 glyphs should be necessary. I was aware that a complete solution was described in Mylonas and Witney (1992), but my fonts became the *de facto* default Greek fonts since the beginning of the nineties.

Apostolos Syropoulos wrote the first version of the Greek language description file, in order to access the babel package facilities (Braams and Bezos, 2014; Syropoulos and Milde, 2014). I wrote an initial pattern file for typesetting in polytonic (ancient) Greek; this file was supposed to be adequate for the western European scholars who need to write ancient Greek, without having available a Greek polytonic keyboard. The Greek language description file was updated several times and a new maintainer, Günter Milde, is taking care of updating/upgrading it. Version 1.9 of this language description file (Syropoulos and Milde, 2014) has just been uploaded to ctan; it contains not only the possibility of selecting the specific variant of the Greek language, but it allows also the direct input of literal Greek.

My Greek hyphenation pattern file was taken care by a succession of Greek maintainers and eventually Dimitrios Filippou became the actual maintainer (Filippou, 2004). My file was eventu-

ally substituted with three other different pattern files, each one suited for the corresponding Greek language variant. But all these files still referred the patterns to the Latin transliteration; they would not work with direct input of literal Greek text.

Meanwhile *xetex* and *xelatex* were introduced in the TEX system; after a short time also *luatex* and *lualatex* were added. These typesetting engines can work with OpenType UNICODE encoded fonts; at the same time they have a more performant system for managing languages, realized through polyglossia (Charette and Reutenauer, 2011) and its commands for selecting languages and their variants. Apostolos Syropoulos decided to support the OpenType fonts and polyglossia, and of course there are no problems as those exhibited by 8-bit encoded fonts and by the babel package with its collection of language description files.

Actually babel can be used also with *xelatex* and *lualatex*; and the `babel-greek` file bundle has a partial support also for being used with the latter typesetting engines. But I suppose that when these two typesetting engines are used, most if not all users, prefer using polyglossia and its special selecting commands that accept the variant specifications by means of *key* = *value* options.

This was the situation by the end of June 2014. Then the problem was spotted; a dense e-mail exchange between the official actors (Günter Milde, Dimitrios Filippou, Mojca Miklavec) and myself was started; we decided a plan to correct this situation, and things are under way to be fine tuned and released to CTAN.

1.2 Latin

The modern and medieval varieties of Latin have received babel and polyglossia support in the past twenty years or so. Classic Latin was not supported. As the author of the existing files, I actually did not know that classic Latin had to receive such a different treatment from the medieval variety. My only excuse is that I am neither a latinist nor a linguist.

Recently I gave a seminar on LATEX to a group of international PhD students in humanities who complained about the fact that they could not use the Latin support for typesetting classic Latin, because spelling and hyphenations were not suited for that Latin variety.

I sought support from a real latinist: my old-time friend Raffaella Tabacco, professor at the university of Vercelli, Italy, and director of the local Department of Humanities that, besides other important activities, is the seat of digilibLT (Biblioteca digitale di testi latini tardo-antichi, *Digital library of late-ancient Latin texts*). My friend, whom I warmly thank for her kind help, gave me access to a book, that is now difficult to find, that contains the hyphenation rules for classic Latin (Farina and Marinone, 1979).

This book, actually, is not very deep in the question of hyphenation, but it gives the essential grammar rules and good advice for the composition of Latin texts and theses on subjects that deal with classic Latin.

In the past I became sort of expert in transforming grammar hyphenation rules into TEX code. I am not a guru nor a wizard in this field, but I have already done this work for more than a dozen languages; some hyphenation pattern files have not been uploaded to CTAN; some of them have been uploaded to CTAN but have been superseded by newer and more correct ones created by other experts; some have been prepared for LATEX 2.09, and of course they became obsolete as soon as LATEX 2_ε became available; the advent of program *xetex*, with its special packages, among which polyglossia, required some upgrading, but in general the existing hyphenation patterns remained available, although with minor but essential adaptations.

I thought I should not encounter any problem in preparing the classic Latin hyphenation patterns, but I did; actually classic Latin requires etymological hyphenation; this kind of hyphenation style requires huge pattern files that have nothing or little to do with grammar rules, in the sense that there is no grammar rule that says what is a prefix or a suffix, or a compound word, or where a certain word comes from. A complete dictionary should be examined and the set of words should be completed with all the flexed forms of verbs, nouns, adjectives, pronouns, and the like.

In this paper I discuss first the problems concerning Greek and the solutions I found; I show some results that I temporarily obtained from the provisional files I used for my tests. Then I discuss what concerns classical Latin and the solutions I found to switch hyphenation patterns when classic Latin is being typeset.

By the time when this paper is available Greek with babel should be almost fixed or shall be completely fixed in a short time with the new patterns files tested and verified by Dimitrios Filippou and his team. With polyglossia there should not be any problem.

At the same time the Latin set up with the three varieties of Latin, modern, medieval, and classical, are fixed when using *pdflatex* and babel, while with polyglossia the upgrading of the `gloss-latin.ldf` language definition file and the new patterns will take e little more time.

2 The LGR encoded Greek fonts

Before going into the details it is necessary to recall the properties of the 8-bit LGR encoded Greek fonts[1]. A table is shown in figure 1.

1. The encoding names starting with L are considered "Local", but for the GReek fonts the LGR encoding has become the *de facto* standard for Greek fonts; any TEX

	’00	’01	’02	’03	’04	’05	’06	’07	’10	’11	’12	’13	’14	’15	’16	’17	
’000	— 0	⌢ 1	𐅄 2	𐅅 3	𐅆 4	𐅇 5	ϟ 6	ϡ 7	ɪ 8	ᾼ 9	ῌ 10	ῼ 11	Α 12	ϔ 13	α 14	ϋ 15	"00
’020	͵ 16	ˎ 17	ϛ 18	ϙ 19	˘ 20	Ϙ 21	Ϡ 22	ƛ 23	€ 24	‰ 25	ə 26	ͳ 27	ʽ 28	ʼ 29	˘ 30	¯ 31	"10
’040	῁ 32	! 33	¨ 34	΅ 35	῭ 36	% 37	· 38	ʹ 39	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47	"20
’060	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	· 59	῾ 60	= 61	᾿ 62	; 63	"30
’100	῏ 64	Α 65	Β 66	῍ 67	Δ 68	Ε 69	Φ 70	Γ 71	Η 72	Ι 73	Θ 74	Κ 75	Λ 76	Μ 77	Ν 78	Ο 79	"40
’120	Π 80	Χ 81	Ρ 82	Σ 83	Τ 84	ϒ 85	῎ 86	Ω 87	Ξ 88	Ψ 89	Ζ 90	[91	῟ 92	] 93	῞ 94	῝ 95	"50
’140	` 96	α 97	β 98	ς 99	δ 100	ε 101	φ 102	γ 103	η 104	ι 105	ϑ 106	ϰ 107	λ 108	μ 109	ν 110	ο 111	"60
’160	π 112	χ 113	ρ 114	σ 115	τ 116	υ 117	118	ω 119	ξ 120	ψ 121	ζ 122	« 123	. 124	» 125	˜ 126	— 127	"70
’200	ὰ 128	ἁ 129	ἀ 130	ἃ 131	ᾲ 132	ᾁ 133	ᾀ 134	ᾃ 135	ά 136	ἅ 137	ἄ 138	ἂ 139	ᾴ 140	ᾅ 141	ᾄ 142	ᾂ 143	"80
’220	ᾶ 144	ἇ 145	ἆ 146	ϝ 147	ᾷ 148	ᾇ 149	ᾆ 150	‿ 151	ὴ 152	ἡ 153	ἠ 154	155	ῂ 156	ᾑ 157	ᾐ 158	159	"90
’240	ή 160	ἥ 161	ἤ 162	ἣ 163	ῄ 164	ᾕ 165	ᾔ 166	ᾓ 167	ῆ 168	ἧ 169	ἦ 170	ἢ 171	ῇ 172	ᾗ 173	ᾖ 174	ᾒ 175	"A0
’260	ὼ 176	ὡ 177	ὠ 178	ὣ 179	ῲ 180	ᾡ 181	ᾠ 182	ᾣ 183	ώ 184	ὥ 185	ὤ 186	ὢ 187	ῴ 188	ᾥ 189	ᾤ 190	ᾢ 191	"B0
’300	ῶ 192	ὧ 193	ὦ 194	Ϝ 195	ῷ 196	ᾧ 197	ᾦ 198	199	ὶ 200	ἱ 201	ἰ 202	ἳ 203	ὺ 204	ὑ 205	ὐ 206	ὓ 207	"C0
’320	ί 208	ἵ 209	ἴ 210	ἲ 211	ύ 212	ὕ 213	ὔ 214	ὒ 215	ῖ 216	ἷ 217	ἶ 218	Ϊ 219	ῦ 220	ὗ 221	ὖ 222	Ϋ 223	"D0
’340	ὲ 224	ἑ 225	ἐ 226	ἓ 227	ὸ 228	ὁ 229	ὀ 230	ὃ 231	έ 232	ἕ 233	ἔ 234	ἒ 235	ό 236	ὅ 237	ὄ 238	ὂ 239	"E0
’360	ϊ 240	ῒ 241	ΐ 242	ῗ 243	ϋ 244	ῢ 245	ΰ 246	ῧ 247	ᾳ 248	ῃ 249	ῳ 250	ῥ 251	ῤ 252	253	ʹ 254	͵ 255	"F0
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

Font: grmn1000.

ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ

αβγδεζηθικλμνξοπρςστυφχψω

Font parameters

slant per pt	0.0pt
interword space	3.33252pt
interword stretch	1.66626pt
interword shrink	1.11084pt
x-height	4.3045pt
quad width	9.99756pt
extra space	1.11084pt

FIGURE 1: The layout of the LGR encoded Greek fonts.

Almost all the 256 positions in the table contain a glyph; some of these glyphs are very unusual in any font, even in the OpenType ones, for example the four glyphs at code points 2–5, that represent the Attic numerals; for example the capital Alpha, Eta and Omega with adscript capital iota at code points 9–11.[2] Some cells appear to be empty, but they are not; for example code point 118 appears to be empty, but it contains an invisible character with vanishing width but as high as a lower case letter without ascenders. It has the usual specification of normal lower case characters, it takes place in the hyphenation process and forbids the lower case closed sigma to see the end of a word; in facts the code point 115 (corresponding to the Latin letter ‘s’) is such that if this letter is used at the end of a word, instead of the open sigma at code point 99 (corresponding to the Latin letter ‘c’), it detects the end of the word and changes itself to an open sigma by means of a ligature process; in fact, in order to typeset an isolated closed sigma it is necessary to input `sv` so as to get σ, because if you enter just `s` you get ς.

The ligature process is used very often with these LGR encoded fonts; the actual ligature commands are contained in the `.tfm` files as well as the kerning commands and the individual glyph dimensions. By this ligature process it is possible to use only the Latin keyboard keys corresponding to the ASCII characters in order to input suitable character sequences that produce “complicated” Greek glyphs; for example by setting `>'a|` in the source file, the output will result in ᾄ.

This process is very handy, but it has a drawback: since the ligature mechanism can process just two characters at a time, the ligated glyph is not kerned

system distribution contains also other packages that allow using the so called BETA code.

2. The decimal code points are written within each table cell; the octal code points are obtained by adding the left row index with the top column index of each cell; the hexadecimal code points are obtained by adding the right row index and the bottom column index.

any more; in other words kerning commands and ligature commands interfere with each other so that each command of one kind forbids the action of the other kind.

For this reason it would be better to have a direct access to pre-composed characters at typesetting level, not at the font level. This is what I tried to do in my teubner package (Beccari, 2013b,a), but Günter Milde introduced a much more versatile method that heavily relies on advanced LaTeX core font commands such as `\DeclareTextCommand`, `\DeclareTextComposite` and the like.

In version 1.8 of the Greek language description file, Milde introduced a new idea: the direct access to actual pre-composed glyphs could be extended by making it independent from the input encoding of the source file. This implied the definition of intermediate macros, the so called LICR macros (LICR stands for "LaTeX Internal Character Representation") so that input characters could be mapped to the LICR macros, which in turn select the pre-composed characters. In this way the literal input of Greek text becomes possible. For example with the previous method, in order to get

> Τῃ πάντα διδούσῃ καὶ ἀπολαμβανούσῃ φύσει ὁ πεπαιδευμένος καὶ αἰδήμων λέγει· «δὸς, ὃ θέλεις, ἀπόλαβε, ὃ θέλεις». Λέγει δὲ τοῦτο οὐ καταθρασυνόμενος, ἀλλὰ πειθαρχῶν μόνον καὶ εὐνοῶν αὐτῃ.

you had to typeset[3]:

```
Th| p\'anta dido\'usih| ka\`i
\>apolambano\'ush| f\'usei
\<o pepaodeum\'enos ka\`i
a\>id\'hmwn l\'egei; ((\<`o
j\'eleis, \>ap\'olabe, \<`o
j\'eleis)). L\'egei d\`e
to\~uto o\>ukatajrasumn\'omenos,
\>all\`a peijarq\~wn m\'onon ka\`i
e\>uno\~wn a\>uth|.
```

With the new LICR method you simply enter the Greek text in the source file and you need not use any diacritic macro to select the pre-composed characters. Of course you need to specify the `utf8` input encoding, but the rest is done directly by the LICR macros.

Obviously in order to do any literal Greek input you need a suitable real or virtual keyboard, unless you copy and paste some text from other electronic documents. Virtual keyboards today are not uncommon, so the first method is actually usable. The copy and paste method is dangerous, because even if the source file to copy some text from appears to contain pre-composed characters, they might not be pre-composed but they might be formed with *self combining diacritics*, which superimpose themselves on the base characters; a sort of mechanism similar to the process used also for Latin diacritics when using *pdflatex* with the obsolete OT1 encoding.

Here I would not enter into the details of the Greek-Latin transliteration, but the example above gives a general insight into this method. Notice that the diacritic macros, as well as the LICR macros at the moment do not access directly the glyphs with iota subscript but resort to the ligature process. Among the fine tuning actions, this is one to be dealt with. The postfixed position of the ASCII symbol for iota subscript does not harm the ligature-kerning process and works pretty well; nevertheless it would be a more general approach if the LICR macros could access also the glyphs with iota subscript.

3 The Greek language description file

As it was previously remarked Milde has already published version 1.9 of `greek.ldf` file. He already added the definitions for the attribute `ancient`, but he also declared the monotonic Greek variant as the default one; with version 3.9 of babel the attributes can be used also as modifiers, therefore the user selects one of the three Greek variants with one of these three self commenting specifications:

```
\usepackage[greek,english]{babel}
\usepackage[greek.polutoniko,english]{babel}
\usepackage[greek.ancient,english]{babel}
```

in the preamble of a document where the main language is English, and the secondary language is one of the three Greek variants. Needless to say that each variant is exclusive and it would be an error to specify both `polutoniko` and `ancient`.

It may be worth noting that the obsolete "pseudo" language `polutonikogreek` might not be supported any more, not even for backwards compatibility with legacy documents. In any case changing the option to babel from `polutonikogreek`[4] to `greek.polutoniko` is not a difficult task for the user who needs to compile again a legacy document.

4 The hyphenation pattern files

The three existing hyphenation pattern files for the three Greek variants to be used by the 8-bit engines are named `grahyph5.tex`, `grmhyph5.tex`,

3. With the ligature mechanism you would save all the backslashes but, with certain fonts and certain glyphs, you would notice some kerning glitches.

4. Günter Milde informed me that there exist several programs that can produce LaTeX files and that still use the `polutonikogreek` "pseudo" language, in spite of the fact that this name is being kept in the Greek language description file only for legacy documents, and its use is deprecated; among these programs there is the well known *LyX* word processor.

Table 1: Single and combined diacritics.

diacritics	◌̓	◌̔	◌̈
accents	◌́	◌̀	◌͂
combined diacritics	◌̓́	◌̔̀	◌̈͂
	◌̓̀	◌̔́	◌̈́
	◌̓͂	◌̔͂	◌̈̀
iota subscript	◌ι		

`grphyph5.tex`: the third letter in their names recalls the initial letter of the specific variant.

Since pattern files must contain only lowercase letters[5] and no expandable macros, the construction of pattern files is pretty complicated if some non ASCII glyphs have to take part in the process.

Th Greek LGR encoded fonts cannot work if only the ASCII code points are used; possibly monotonic Greek can get along with such limitations when using the ISO-8859-7 encoding[6] that is being used by a normal Greek keyboard. So for both polytonic varieties it is necessary to address the upper half plane of the 8-bit font encoding.

This requires to assign a lower case code to all diacritics, the simple ones and the combined ones; see table 1.

And since these simple and combined diacritics must appear in the pattern files as ASCII glyphs, they must be set with their lower case codes.

For monotonic greek it is necessary to set the lower case code also to the seven accented letters with the single acute accent and the iota and upsilon with just the diaeresis or with the diaeresis and acute accent. With both polytonic varieties it is necessary to set also the lower case code of approximately 120 upper plane glyphs that include all the available base letters with all combinations of allowed diacritics.

This second part was missing from the `gr*hyph5.tex` files. The new `gr*hyph6.tex` files had to be completed with the lower case codes of all the available letters. But this was not sufficient.

The actual patterns included in the `gr*hyph5.tex` files contained the diacritics only as simple ASCII characters, not as diacritic macros (forbidden in the pattern files), nor with the actual code points of the pre-composed glyphs. The upper plane ones are nor representable with ASCII codes, therefore it is necessary to use their numerical code points. The most cryptic representation of the numerical code point is the double caret lower case hexadecimal representation, where, for example, `^^ba` addresses the Greek glyph ὦ.

5. For use within the pattern files a *letter* is any glyph with a positive lower case code. Several languages assign a positive lower case code to analphabetic characters, such as the apostrophe, for example; Italian is one of these languages.

6. The ISO standard for monotonic Greek.

This double character representation, in spite of being so cryptic, is useful because it does not require spaces after control sequence names, and its two hexadecimal digits are well defined as belonging to the glyph address, and cannot be confused with the numerical digits used in the patterns to identify possible allowed or prohibited hyphen points.

Therefore the pattern `gr*hyph6.tex` files had to be enriched, compared to the existing `gr*hyph5.tex`, with the new patterns containing the double caret hexadecimal code points of the pre-composed glyphs. The `grmhyph6.tex` required very little work, because the patterns to add were relatively few. The `grphyph6.tex` file required a very attentive long editing, because the new patterns to add were several hundred ones. The `grahyph6.tex` file required several days of work, because there were about 2000 patterns to add.

5 Testing

I run some tests on monotonic, modern and ancient polytonic texts and found that several patterns were missing; not only the new ones with the pre-composed glyph double caret code points, but also the previous ones based on the ligature mechanism. The problem was very delicate with the ancient Greek variant. The difficulty arises from the fact that prefixes and suffixes are hyphenated on an etymological base; patterns become increasingly more complicated, because they create several variants for the same prefix or suffix and the same word stems where accents change position according to the word declination or conjugation.

Moreover Greek is perhaps the only language where typographical hyphenation can take place after the initial vowel and before the ending vowel. This feature implies more detailed patterns and certainly it does not simplify the whole process. Notice that when I created the hyphenation patters for Italian I worked as if it were possible to hyphenate after the initial vowel and before the ending vowel, but I set the minimum length of the first and last syllable as *two* letters; if the user needs more break points, s/he can locally set the `\lefthyphenmin` and `\righthyphenmin` counters to one, but by default they equal two. This said, the hyphenation trie for Italian contains about 340 patterns and 37 ops; for ancient Greek more than 5000 patterns are necessary end they require 122 ops. In spite of a very similar sonority, the two polytonic Greek variants require much more attention due to the multitude of diacritics and the etymological hyphenation.

I introduced the modifications I considered necessary; since I am not a Greek speaker, but I am just interested in this language that I happened to study in high school (hyphenation was not part of the syllabus...), I think I did the best I could,

but what I did is certainly very far from perfect; of course now the work in progress consists in a full control of the new Greek pattern files performance, and Dimitrios Filippou is working on that.

At the same time it seems that the UNICODE compliant pattern files created for the three Greek variants to be typeset with UNI-aware typesetting engines could be converted to LGR encoding with a suitable conversion program, similar to the ones used with the other existing UNICODE to ⟨*encoding*⟩ conversion programs. Mojca Miklavec volunteered to create such program. If this conversion program can do a good job, may be that what I did becomes automatically obsolete. In facts with a good conversion program it is possible to maintain only the UNICODE compliant pattern files, and the LGR compliant ones can be derived directly from the updated/upgraded UNICODE files.

But my work represents a feasibility proof that the now existing pattern files can be edited to become suitable for typesetting with 8-bit aware engines, *pdflatex* in particular, while, at the moment, they are not suited for this purpose.

6 Examples

Here some hyphenated words are shown with words taken from the comment lines of the `grahyph6.tex` pattern file. Supposedly they will be hyphenated correctly if the ancient Greek language is selected; but we use the same set of words to show the hyphen points also with the other two variants; of course the words for monotonic are input with a single tonic accent and/or dialitika. Here is the list.

ἀδυσώπητος ἀναίδεια ἀναμφισβήτητος
Ἀναξάνδρος ἄνοικος ἀπρόσκοπος
ἀπρόσκοπτος εἰσίν ἐκλείσαμεν ἐκληροῦτον
ἐκρυστάλλισα Ἑλλήσποντος ἐνίδρωσα
εὐσύνετος καλωσόρισα μελανόρισα
μελανόμματος μαγισαψεδάφα ξυναγείρατο
παλινάγρετος πανισδόμην προσεισπτάσσω
προϋπεξορμάω ὑοσκύαμος ὑπεκλήψομαι
χαρίσανδρος δισχίλιοι

The testing results are shown in figures 2, 3, and 4.

Comments

It may be noticed that words with an initial upper case precomposed vowel require patterns that need an hybrid approach between ASCII patterns and LGR literal Greek transformation into double caret code points; for example, the word Ἀναξάνδρος requires three patterns, the first containing only ASCII characters, `>an'a2x1an`; the second with only double caret code points, `^^82n^^882x1an`; the third in hybrid form, `>an^^882x1an`. This is due to the fact that before hyphenating, the typesetting engine internally changes the word to be hyphenated to lowercase; but since diacritics with the

αδυ-σώ-πη-τος	κα-λω-σό-ρι-σα
αναί-δεια	με-λα-νό-ρι-σα
αναμ-φι-σβή-τη-τος	με-λα-νόμ-μα-τος
Ανά-ξαν-δρος	μα-γι-σα-ψε-δά-φα
άνοι-κος	ξυ-να-γεί-ρα-το
απρό-σκο-πος	πα-λι-νά-γρε-τος
απρό-σκο-πτος	πα-νισ-δό-μην
ει-σίν	προ-σει-σπτάσ-σω
εκλεί-σα-μεν	προ-ϋ-πε-ξορ-μάω
εκλη-ρού-τον	υο-σκύ-α-μος
εκρυ-σ τάλ-λι-σα	υπε-κλή-ψο-μαι
Ελ-λή-σπο-ντος	χα-ρί-σαν-δρος
ενί-δρω-σα	δι-σχί-λιοι
ευ-σύ-νε-τος	

Figure 2: Hyphenation test in monotonic Greek.

ἀ-δυ-σώ-πη-τος	κα-λω-σό-ρι-σα
ἀ-ναί-δεια	με-λα-νό-ρι-σα
ἀ-ναμ-φι-σβή-τη-τος	με-λα-νόμ-μα-τος
Ἀ-νά-ξαν-δρος	μα-γι-σα-ψε-δά-φα
ἄνοι-κος	ξυ-να-γεί-ρα-το
ἀ-πρό-σκο-πος	πα-λι-νά-γρε-τος
ἀ-πρό-σκο-πτος	πα-νισ-δό-μην
εἰ-σίν	προ-σει-σπτάσ-σω
ἐκλεί-σα-μεν	προ-ϋπε-ξορ-μά-ω
ἐκλη-ροῦ-τον	ὑ-ο-σκύ-α-μος
ἐκρυ-στάλ-λι-σα	ὑ-πε-κλή-ψο-μαι
Ἑλ-λή-σπον-τος	χα-ρί-σαν-δρος
ἐνί-δρω-σα	δι-σχί-λιοι
εὐ-σύ-νε-τος	

Figure 3: Hyphenation test in polytonic modern Greek.

ἀ-δυσ-ώ-πη-τος	εὐ-σύν-ε-τος
ἀν-αί-δει-α	κα-λωσ-ό-ρι-σα
ἀν-αμ-φισ-βή-τη-τος	με-λα-νό-ρι-σα
Ἀ-νάξ-αν-δρος	με-λαν-όμ-μα-τος
ἀ-νάξ-αν-δρος	μο-γισ-α-ψε-δά-φα
ἄν-οι-κος	ξυν-α-γεί-ρα-το
ἀ-πρό-σκο-πος	πα-λιν-ά-γρε-τος
ἀ-πρό-σκο-πτος	πα-νισ-δό-μην
εἰσ-ίν	προσ-εισ-πτάσ-σω
ἐ-κλεί-σα-μεν	προ-ϋπεξ-ορ-μά-ω
ἐ-κλη-ροῦ-τον	ὑ-οσ-κύ-α-μος
ἐ-κρυ-στάλ-λι-σα	ὑ-πεκ-λή-ψο-μαι
Ἑλ-λήσ-πον-τος	χα-ρίσ-αν-δρος
ἐλ-λήσ-πον-τος	δισ-χί-λι-οι
ἐν-ί-δρω-σα	

Figure 4: Hyphenation test in polytonic ancient Greek.

initial capital letter of a word do not get ligated in a single character with the diacritic over the base letter, but before the base letter, the pattern must contain the initial letter in ASCII form as if the diacritic had to be ligated; but the ligature takes place at the font level, while direct access to the upper case accented character takes place before the paragraph construction, and therefore before the hyphenation algorithm gets executed.

Even if the hyphenation results of the ancient Greek variant appears pretty unusual, it has to be so. The file I modified contained exactly those hyphen points and Dimitrios Filippou, who supervised and contributed to the creation of the `grahyph5.tex` file, knows exactly how this Greek variant has to be processed; he explained the whole operation that the Greek team brought to a conclusion in his chapter *Hyphenation patterns for ancient and modern Greek* included in the book by Apostolos Syropoulos (FILIPPOU, 2004).

7 Differences between Latin and classic Latin

The existing Latin pattern files that I prepared some 20 years ago, dealt with modern and medieval Latin with just one set of patterns; the language description file for Latin provided differently spelled infix words such as *Praefatio* and *Præfatio*, or *novembris* and *nouembris*; and the pattern files were set up to recognize the use of the letter 'u' in place of the letter 'v' as used in modern spelling. The hyphenation rules were taken from an high school grammar, that synthetically stated that the rules are identical to those for Italian; I verified how such rules were followed in in a "professional" book, typeset in modern spelling, *Novum testamentum graece et latine*, (MERK, 1984), and I verified that such rules were actually followed in a systematic way, except very few inconsistent instances such as *re-gnum* vs. *reg-num*, both being present in different parts of the book.

8 Classic Latin hyphenation rules

Classic Latin is treated in a different way.

The rules reported by FARINA and MARINONE (1979) are the following:

1. The classic Latin alphabet contains only the following 23 letters: *a b c d e f g h i k l m n o p q r s t u x y z* with the corresponding upper case ones, with the exception that *V* is the upper case version of *u*. Supposedly letters *y z* appear only in words deriving from Greek. Moreover letters *j v* do not exist, so that in a correct classic spelling they should be substituted with *i u* respectively.

2. Two vowels can be divided if they do not form a diphthong, but it is preferable to avoid such divisions; for example *re-us cre-a-re gau-di-um* should remain *reus crea-re gau-dium* respectively.

3. Single consonants, the graphemes *ch ph rh th qu*, the letters *x z*, and *i u* when they have the value of a consonant, remain attached to the following vowel; for example *ro-sa me-ta mo-ri-tu-rus phi-lo-so-phus e-quus ae-qui-tas di-xi a-zy-mus ie-iu-ni-um ui-ue-re.*

4. Groups of two consonants where the first is one of *p b t d c g f* and the second one is one of *l r*, are never divided; exceptionally also *gn* is not divided; for example *du-plum pu-bli-cus pa-tres sa-cra a-grus ua-frum sta-gnum.*

5. Any other group of two or more consonants, irrespective if they are equal or different, are divided just before the last one, except when the last two consonants fall in the previous case; for example *ag-ger es-se uel-le fac-tus op-ti-mus ag-men om-nes al-tus mer-ces cer-no mul-tum tem-pus uin-dex sun-to tex-tor scrip-si ip-se cas-tus quaes-tor sanc-tus emp-tor tem-plum nos-tri cas-tra.*

6. In compound words and when an enclitic is present it is customary to divide the elements even if the previous rules are violated, unless the elements are merged in a unique joining letter; examples: *ex-i-re red-i-re ex-ta-re ob-ru-e-re abs-ti-ne-re ads-cri-be-re mul-tas-que uis-ne suap-te pae-nin-su-la quin-quen-ni-um.*

7. Words derived from Greek are divided according to the Greek rules; for example *sce-ptrum dra-chma cy-cnus i-sthmus rhy-thmus A-ri-sto-te-les.*

Such rules appear not particularly difficult to implement with patterns, but at a second look the real difficulties show up. The rules contain too many words or phrases such as "exceptionally", "except when", "it is customary", "derived from".

In facts patterns are not so intelligent as to know when there are exceptions to the rules or when some words are not of Latin origin; something can be done with prefixes or suffixes, but not all of them; patterns for words of Greek origin may be implemented if they imply the digraphs *ch ph th rh* or the letters *y z*, but a word such as *sceptrum* cannot be detected as being of Greek origin.

9 The actual patterns

In this situation it would be more convenient to create the necessary patterns by means of a huge hyphenated word list and to apply *patgen*, (LIANG, 1983); even with 100 000 word lists it is impossible to be sure that wrong break points never show up.

I decided to do my best with the above rules in creating the classic Latin patterns, and leave the user the responsibility of using an exception list by means of the command `\hyphenation` that is available for this purpose.

I temporarily left the ligatures æ and œ among the possible Latin letters, but the user should refrain from using them, because they do not belong to classic Latin; in future releases of this pattern file such letters probably will be eliminated.

I followed the suggestion of Farina and Marinone (1979) to avoid diphthongs; therefore I could avoid the difficulty of deciding if *i u* play the rôle of vowels or consonants. In this way some possible break points are missed, but many errors are avoided. I followed this same philosophy also when preparing the hyphenation patterns for several other languages I created patterns for; it proved to be a wise decision and I continue to follow this philosophy: "a missed break point is better than a wrong one".

I think I did a good job with prefixes *ab abs ob circum ex para sub* and suffixes *que ne*. I could not do the same (supposedly) good job with other prefixes: for example *re* appears in a lot of words where it is not a prefix; the same with *red trans con* and so on. This is a typical application of the `\hyphenation` exception lists.

In any case I created a *shorthand* by means of character `"` that is never used in Latin; both babel and polyglossia allow the definition of shorthands, so that I defined the above character as an *active* one; an active character behaves as a TEX macro; I defined it as to insert a *compound word mark*; such mark inserts a *word boundary* and a discretionary break that, contrary to the default definition of `\-`, it allows hyphenation on both word elements. For example, without this mark the word *transierat* would get divided as *tran-si-e-rat*; but if the mark is inserted at the end of prefix "trans", as in `trans"ierat`, then the word is correctly divided as *trans-ie-rat*. Actually the correct division should be *trans-i-e-rat* but TEX refrains from breaking words leaving strings that are too short before or after the break points; in classic Latin the initial and terminal "shortness" limits are specified in two letters, so that initial or terminal syllables of one letter are forbidden.

The above shorthand should not to be used in every compound word; this means that the user should initially enter the Latin text in the source TEX file without any concern to compound words; when revising the drafts s/he might introduce the shorthand in those rare instances where compound words are not hyphenated correctly on the word boundaries. Alternatively s/he can enter the correctly hyphenated words in the argument of `\hyphenation`. Beware, though; in order to have `\hyphenation` work correctly in every situation, it is not sufficient to enter one hyphenated word; but, since Latin has both declination of nouns and adjectives and conjugation of verbs„ it would be necessary to enter all cases: 12 entries for nouns, 36 entries for adjectives, about 100 entries for verbs for all persons, modes, for the three masculine, feminine and neutral present, past and future participles, gerund, infinitive and for the active and passive diathesis; well, probably in the same document it is very unusual that all forms of a verb are present, but just in case...

With babel (version 3.9 or higher) it is possible to use the command `\babelhyphenation` that extends the funcionality of the standard LATEX `\hyphenation` command; it has the syntax:

`\babelhyphenation[`⟨*language*⟩`]{`⟨*exceptions*⟩`}`

where ⟨*language*⟩ is a language name (in our case `latin` in lower case letters, the name that babel understands) or a list of comma separated language names; if no ⟨*language*⟩ is specified, the ⟨*exceptions*⟩ are valid for all languages used in the document, otherwise they are valid only for the specified language(s).[7]

10 Testing

I test my patterns by means of a small package I wrote myself, testhyphens (Beccari, 2014); this package implements for LATEX the plain TEX code that was contributed to the TEX community by Jonathan Kew (Eijkhout, 1993). This package defines the checkhyphens environment; it contains a space delimited word list, or, even better, some real text in the language that is being tested; by running *pdflatex* on the source file, the words of this text are printed on consecutive lines with their break points.

I find it very convenient and, when a real text is entered in the checkhyphens environment I can see the hyphenation algorithm at work even when the compound word mark shorthand is used. I cannot get the complete test of all my patterns, but I can always add to that argument as many unusual words as I want and check their break points.

For example I tested with `\checkhyphens` the following text from Julius Caesar's *De bello gallico*:

> Flumen est Arar, quo per fines Haeduorum et Sequanorum in Rhodanum influit, incredibili lenitate, ita ut oculis in utram partem fluat iudicari non possit. Id Heluetii ratibus ac lintribus iunctis transibant. Ubi per exploratores Caesar certior factus est tres iam partes Heluetios id flumen traduxisse, quartam fere partem citra flumen Ararim reliquam esse,

7. This functionality is not yet available with polyglossia since it has not been upgraded by the end of August 2014.

Flu-men	u-tram	fac-tus	ter-tia	im-pe-di-tos
est	par-tem	est	ui-gi-lia	et
A-rar,	fluat	tres	cum	i-no-pi-nan-tes
quo	iu-di-ca-ri	iam	le-gio-ni-bus	ad-gres-sus,
per	non	par-tes	tri-bus	mag-nam
fi-nes	pos-sit.	Hel-ue-tios	e	par-tem
Hae-duo-rum	Id	id	cas-tris	eo-rum
et	Hel-ue-tii	flu-men	pro-fec-tus	con-ci-dit:
Se-qua-no-rum	ra-ti-bus	tra-du-xis-se,	est	re-li-qui
in	ac	quar-tam	ad	se-se
Rho-da-num	lin-tri-bus	fe-re	eam	fu-gae
in-fluit,	iunc-tis	par-tem	par-tem	man-da-runt
in-cre-di-bi-li	trans-i-bant.	ci-tra	per-ue-nit	at-que
le-ni-ta-te,	U-bi	flu-men	quae	in
i-ta	per	A-ra-rim	non-dum	pro-xi-mas
ut	ex-plo-ra-to-res	re-li-quam	flu-men	sil-uas
o-cu-lis	Cae-sar	es-se,	trans-ie-rat.	ab-di-de-runt.
in	cer-tior	de	Eos	

FIGURE 5: The output of the checkhyphens environment on the test file taken from *De bello gallico*. Parameters `\lefthyphenmin` e `\righthyphenmin` have been set to 1; but the default language settings assume they equal 2.

> de tertia uigilia cum legionibus tribus e castris profectus est ad eam partem peruenit quae nondum flumen transierat. Eos impeditos et inopinantes adgressus, magnam partem eorum concidit: reliqui sese fugae mandarunt atque in proximas siluas abdiderunt.

The compound word mark shorthand was used in `trans"ibant` and `trans"ierat`; it actually enters the scene in the word *trans-ibant* at the end of its line. Of course with checkhypens all break points are shown; besides the break points created with the compound word mark, it is interesting to see *Hel-ue-tii, per-ue-nit* where the *u* plays the rôle of a consonant as the the letter *v* in modern spelling.

The output of the test is shown in figure 5.

11 The Latin language description file

Package babel accepts the definition of attributes: with version 3.9 they are preferably used as modifiers. Package polyglossia has always accepted option lists in the specification of some of its working languages.

I had to write again both description files `babel-latin.ldf` and `gloss-latin.ldf` in order to add the attribute/modifier `classic`. The new version of `gloss-latin.ldf` now behaves as such: if no options are specified with polyglossia to the specific command `\setotherlanguage{latin}`, the modern spelling is assumed and the shorthand is not defined. The option `babelshorthands` activates the compound word mark shorthand; the keyword `variant` set to one of `modern, medieval, classic` specifies a Latin spelling style (`modern` being the default); these variant values are mutually exclusive and it is not possible to switch style within a given document. The real reason of this constraint is that different hyphenation pattern files are used: one is for classic Latin, while the other works with both modern and medieval Latin.[8]

For babel I had to discover the undocumented trick for specifying a different pattern set within the same language description file; actually the trick is very simple and similar to the one used for the Greek description file: it amounts to associate the control sequence that addresses the language counter to a different value. After getting by this small difficulty, it was simple to integrate classic Latin in the previous `babel-latin.ldf`.

The set of files to be used with *pdflatex* are available with the 2014 distribution of the TEX system; of course I can already use these updates on my computer even when using *xelatex*, because I created them and I know how to integrate hyphen pattern files in the formats of my installation. It is a pretty delicate procedure that I am not going to explain here; in facts it is not necessary to know these inner workings of the TEX system, because such operations are automatically performed each time a user updates its installation with the TEX Live program *tlmgr* or with the MiKTEX wizard.

12 Conclusion

For what concerns Greek the actual hyphenation pattern files for 8-bit aware typesetting engines are not suited for direct literal input of the Greek text in the source files. I discovered the problem and found a solution. The three maintainers of

8. I already uploaded the new version of the `gloss-latin.ldf` but by the end of August 2014 it is not yet available with polyglossia.

the various aspects of the Greek language handling, Mojca Miklavec for the overall file handling in CTAN, Günter Milde for the Greek support to the babel package, Dimitrios Filippou for the hyphenation pattern files, are doing the rest of the work in order to have an efficient and complete set of instruments to enhance the Greek language support. I thank them all very much for what they are doing.

For what concerns classic Latin producing hyphenation pattern files by implementing the grammar rules is not particularly difficult; it requires just some attention and a lot of patience; the TEXbook (Knuth, 1996) explains everything about hyphenation patterns so that creating valid pattern files is not a terrible task.

It goes without saying that the examples set forth by Farina and Marinone (1979) work correctly. The new patterns show their effectiveness in figure 5, but there remain still some problems with certain prefixes. In facts the prefix *trans* had to be "terminated" with a compound word marker in order to have the correct break points. The weak points of my patterns with prefixes and suffixes have been discussed in section 9.

13 Acknowledgements

I acknowledge the help received from Günter Milde, Dimitrios Filippou and Mojca Miklavec who are now the three pillars of the whole Greek support to the babel system; I just happened to spot the problem that is described in this article; I gave suggestions to Mojca Miklavec; I sent some patches to Günter Milde; I reworked the pattern files for Dimitrios Filippou. But the real work, the part of the work that is still under way, is in their hands. When everything will be finished it will be their merit to have put together some typesetting instruments for the Greek language, that will be useful not only to the Greek TEX users, but to the whole TEX community.

I hope that by the time this paper is published, their work is finished and integrated into CTAN and TEX Live.

I want to express again my thanks also to Raffaella Tabacco; without her support I would not have been able to do anything in this field.

References

Beccari, C. (2013a). «`teubner.sty` – An extension to the greek option of the *babel* package». PDF document. The PDF document is in CTAN but it can be easily read by means of the terminal command `texdoc teubner-doc`.

— (2013b). «The `teubner` package –Extensions for Greek philology». PDF document. The PDF file is archived on CTAN, but it can be readily read with the terminal command `texdoc teubner`.

— (2014). «The testhyphens package». PDF document. The document is part of the TEX Live 2014 distribution. It can be read by means of the termial command `texdoc testhyphens`.

Braams, J. and Bezos, J. (2014). «Babel». PDF document. Easily readable with the terminal command `texdoc babel`.

Charette, F. and Reutenauer, F. (2011). *Polyglossia: a Babel replacement for XeLATEX*. TUG. Easily readable with the terminal command `texdoc polyglossia`.

Eijkhout, V. (1993). «The bag of tricks». *TUGboat*, **14** (4), p. 424.

Farina, R. and Marinone, N. (1979). *Metodologia – Guida pratica alle esercitazioni di seminario e alle tesi di laurea per le discipline umanistiche*. Società editirice Internazionale, Torino.

Filippou, D. (2004). «Hyphenation patterns for ancient and modern Greek». In *TEX, XML, and digital typography*, Springer-Verlag, Berlin-Heidelberg.

Knuth, D. E. (1996). *The TEXbook*. Addison Wesley, Reading, Mass., 16th edition.

Levy, S. and Murphy, T. (2010). «Using Greek fonts with LATEX». PDF document. This PDF document is available on CTAN and simply read by means of the terminal command `texdoc lgreekuse`.

Liang, F. (1983). *Word Hy-phen-a-tion by Comput-er*. PhD thesis, Stanford University.

Merk, A. (ed. by) (1984). *Novum testamentum graece et latine*. Sumptibus Pontificii Instituti Biblici, Romae, 10th edition.

Mylonas, C. and Witney, R. (1992). «Complete Greek with adjunct fonts». *TUGbooat*, **13** (1), p. 39–50.

Syropoulos, A. and Milde, G. (2014). «Babel support for the greek language». PDF document. The PDF file `babel-greek.pdf` is available on CTAN and easily readable by means of the terminal command `texdoc babel-greek`.

▷ Claudio Beccari
Professore emerito
Politecnico di Torino
`claudio dot beccari at gmail.com`