Attic (or Athenian) Numerals

Apostolos Syropoulos
366, 28th October Str.
GR-671 00 Xanthi, HELLAS
Email:`apostolo@platon.ee.duth.gr`

2003/08/24

1 Introduction

`\athnum` This LaTeX package implements the macro `\athnum`. The macro transforms an Arabic numeral, i.e., the kind of numerals we all use (e.g., 1, 5, 789 etc), to the corresponding *Athenian* (or *Attic* numeral. Athenian numerals were in use mainly in ancient Athens. The package requires a Greek text font and the `greek-fontenc` package.[1]

2 The Numbering System

The athenian numbering system, like the roman one, employs letters to denote important numbers. Multiple occurrence of a letter denote a multiple of the "important" number, e.g., the letter I denotes 1, so III denotes 3. Here are the basic digits used in the Athenian numbering system:

- I denotes the number one (1)
- Π denotes the number five (5)
- Δ denotes the number ten (10)
- H denotes the number one hundred (100)
- X denotes the number one thousand (1000)
- M denotes the number ten thousands (10000)

Moreover, the letters Δ, H, X, and M under the letter Π, denote five times their original value, e.g., the symbol 𐅇, denotes the number 5000, and the symbol 𐅄,

[1] Up to version 1.1, 'athnum' required that Babel was loaded with the `greek` option. Version 2.0 sets up Greek font support autonomously.

denotes the number 50. It must be noted that the numbering system does not provide negative numerals or a symbol for zero.

The Athenian numbering system is described, among others, in an article in Encyclopedia $\Delta o\mu\acute{\eta}$, Vol. 2, page 280, 7th edition, Athens, October 2, 1975.

3 The Code

Before we do anything further, we have to identify the package.

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}[1996/01/01]
\ProvidesPackage{athnum}[2022/12/08\space v2.0]
\typeout{Package: ‘athnum’ v2.0\space <2022/12/08> (A. Syropoulos)}
```

4 Font setup

Typesetting Greek texts requires a font with Greek characters:

For 8-bit LaTeX, this package uses fonts with the LGR font encoding. With the XeTeX or LuaTeX engines and Unicode fonts, the user must ensure that the selected font contains the required glyphs (only few fonts support the "GREEK ACROPHONIC ..." characters). LGR-encoded fonts can also be used alongside Unicode fonts with XeTeX/LuaTeX to enable the input of Greek letters via the Latin transliteration provided by this font encoding.

`\greekfontencoding`

We test for available font encodings and set `\greekfontencoding` and the internal macro `\athnum@greek@fontencdef` with the name of the font definition file that defines the required Greek LICR macros. The default is LGR for 8-bit TeX engines and TU for Xe/LuaTeX.

```
\ifdefined \UnicodeEncodingName % set by LaTeX for XeTeX/LuaTeX
  \providecommand*{\greekfontencoding}{\UnicodeEncodingName}
  \providecommand*{\athnum@greek@fontencdef}{tuenc-greek}
\else
  \providecommand*{\greekfontencoding}{LGR}
  \providecommand*{\athnum@greek@fontencdef}{lgrenc}
\fi
```

Ensure that the Greek LICR macros are available:

```
\@ifl@aded{def}{\athnum@greek@fontencdef}{}
{% else
  \InputIfFileExists{\athnum@greek@fontencdef .def}{}{% else
    \PackageError{athnum}{Font support for the Greek script missing.\\
              can’t typeset Greek symbols.\\
              Install the "greek-fontenc" package\\
              or use XeTeX/LuaTeX.}
             {I can’t find the \athnum@greek@fontencdef .def file
              for the Greek fonts (encoding \greekfontencoding)}
    \@@end
  }
}
```

`\ensuregreek` The TextCommand `\ensuregreek` sets it argument in `\greekfontencoding` if the current font encoding does not provide a (typically empty) local variant.

```
\ProvideTextCommandDefault{\ensuregreek}[1]{%
  \leavevmode{\fontencoding{\greekfontencoding}\selectfont #1}}
```

`\@@athnum` Now, we turn our attention to the definition of the macro `\@@athnum`. This macro uses one integer variable (or counter in TEX's jargon.)

```
\newcount\@ath@num
```

The macro `\@@athnum` is also defined as a robust command.

```
\DeclareRobustCommand*{\@@athnum}[1]{%
```

After assigning to variable `\@ath@num` the value of the macro's argument, we make sure that the argument is in the expected range, i.e., it is greater than zero, and less or equal to 249999. In case it isn't, we simply produce a `\space`, warn the user about it and quit. Although, the `\athnum` macro is capable to produce an Athenian numeral for even greater intergers, the following argument by Claudio Beccari convised me to place this above upper limit:

> According to psychological perception studies (that ancient Athenians and Romans perfectly knew without needing to study Freud and Jung) living beings (which includes at least all vertebrates, not only humans) can perceive up to four randomly set objects of the same kind without the need of counting, the latter activity being a specific acquired ability of human kind; the biquinary numbering notation used by the Athenians and the Romans exploits this natural characteristic of human beings.

```
        \@ath@num#1\relax
        \ifnum\@ath@num<\@ne%
          \space%
          \PackageWarning{athnum}{%
          Illegal value (\the\@ath@num) for athenian numeral}%
        \else\ifnum\@ath@num>249999%
          \space%
          \PackageWarning{athnum}{%
          Illegal value (\the\@ath@num) for athenian numeral}%
        \else
```

Having done all the necessary checks, we are now ready to do the actual computation. If the number is greater than 49999, then it certainly has at least one 𐅇 "digit". We find all such digits by continuously subtracting 50000 from `\@ath@num`, until `\@ath@num` becomes less than 50000.

```
            \@whilenum\@ath@num>49999\do{%
               \textpentemuria\advance\@ath@num-50000}%
```

We now check for tens of thousands.

```
            \@whilenum\@ath@num>9999\do{%
               \textMu\advance\@ath@num-\@M}%
```

Since a number can have only one 𐅇 "digit" (equivalent to 5000), it is easy to check it out and produce the corresponding numeral in case it does have one.

```
            \ifnum\@ath@num>4999%
               \textpenteqilioi\advance\@ath@num-5000%
            \fi\relax
```

Next, we check for thousands, the same way we checked for tens of thousands.

```
            \@whilenum\@ath@num>999\do{%
               \textChi\advance\@ath@num-\@m}%
```

Like the five thousands, a numeral can have at most one 𐅅 "digit" (equivalent to 500).

```
            \ifnum\@ath@num>499%
               \textpentehekaton\advance\@ath@num-500%
            \fi\relax
```

It is time to check hundreds, which follow the same pattern as thousands

```
            \@whilenum\@ath@num>99\do{%
               \textEta\advance\@ath@num-100}%
```

A numeral can have only one 𐅄 "digit" (equivalent to 50).

```
            \ifnum\@ath@num>49%
               \textpentedeka\advance\@ath@num-50%
            \fi\relax
```

Let's check now decades.

```
            \@whilenum\@ath@num>9\do{%
               \textDelta\advance\@ath@num by-10}%
```

We check for five and, finally, for the digits 1, 2, 3, and 4.

```
            \@whilenum\@ath@num>4\do{%
               \textPi\advance\@ath@num-5}%
            \ifcase\@ath@num \or \textIota
       \or \textIota\textIota
       \or \textIota\textIota\textIota
       \or \textIota\textIota\textIota\textIota\fi%
   \fi\fi}
```

`\@athnum` The command `\@athnum` has one argument, which is a counter. It calls the command `\@@athnum` to process the value of the counter.

```
\def\@athnum#1{%
     \expandafter\@@athnum\expandafter{\the#1}}
```

`\athnum` The command `\athnum` is a wrapper that declares a new counter in a local scope, assigns to it the argument of the command and calls the macro `\@athnum`. This way the command can process correctly either a number or a counter.

```
\def\athnum#1{%
     \ensuregreek{%
       \@ath@num#1\relax
       \@athnum{\@ath@num}}%
     }
⟨/package⟩
```

Acknowledgment

I would like to thank Claudio Beccari for reading the documentation and for his very helpful suggestions. In addition, Antonis Tsolomitis spotted a bug in the first version, which is corrected in the present version.

Dedication

I would like to dedicate this piece of work to my son

Demetrios-Georgios.