

# Intelligent commas

Claudio Beccari

Email [claudio dot beccari at gmail dot com](mailto:claudio dot beccari at gmail dot com)  
Address Villarbasse (TO), Italy

Abstract The decimal fractional part of a number must be separated from the integer part with a decimal separator. The ISO regulations specify a different sign for different languages; the internal  $\LaTeX$  mathematical character codes do their best to *avoid* a simple treatment of the decimal separator. Here we describe a few ways to handle this problem.

## 1 Introduction

The history of the decimal separator between the integer and the fractional part of a decimal number is quite interesting, but nowadays, where the ISO regulations standardise everything, the historical traditions reflect themselves only in the language dependent variations of this sign. The ISO regulations require the decimal point when a number is within a text in English, or a decimal comma otherwise. In Great Britain the centered dot is still common, but this nice tradition is in contrast with the ISO regulations.

Therefore this different language dependent treatment of the decimal separator, according to my opinion, should be dealt with during the language selection of the `babel` or the `polyglossia` packages. Up to now the efforts I made to insert a proper treatment connected with the `italian` option to the `babel` package have produced no effects. It's reasonable: why only Italian and not all other languages?

The other side of the medal is the  $\TeX$  point of view: any symbol to be used in mathematics has a mathcode that specifies its rôle. When the decimal point is used as a separator, its mathcode classifies it as a 'normal' symbol, which means that it receives no spacing before and after it; when the period is used as a punctuation mark, this happens only at the end of a mathematical expression and it does not separate anything, so the default null spacing is irrelevant.

Quite the opposite is the case, however, when the comma is used as a decimal separator. It must be treated differently from when it is used as a punctuation mark, for example when it is used as a list item separator. Compare, as an example, the independent variable list  $f(x, y)$  with  $f(x, y)$ ; compare also the decimal separator  $e = 2,718$  and  $e = 2, 718$ . This happens because the rôle of a punctuation mark implies a small space on its right. Of course the spacing may be fixed by using the math spacing commands, but this is contrary to good sense. (L<sup>A</sup>)T<sub>E</sub>X should be able to treat the problem letting the author concentrate on what s/he has to write, not on what it looks like.

## 2 Ready made solutions

There are several packages that solve this problem; on the one hand, there are packages that deal with both the right sign and the substitution of the wrong sign. Among these we can cite numprint [1] and siunitx [4].

The first package not only works in both text and math mode, but it also changes the ‘wrong’ decimal sign into the ‘right’ one, besides transforming the computer output into a mathematical expression: it can change  $12345.6e789$  into  $12345,6 \cdot 10^{789}$  while using the proper fonts both in math and text mode.

The second package is a complete instrument for dealing with units (SI units in particular) and their numerical values not only in math and text modes, but also in tabular mode so as to have a perfect typeset document with the proper decimal signs, the proper fonts, the proper alignment in table columns, the proper unit symbols with the proper unbreakable spacings. Complete service!

On the other hand, there are also packages that deal only with the proper mathcodes in mathematical typesetting; I can cite for example icomma [3] and nccomma [2].

The first package defines an active math comma that tests whether or not it is followed by a space; if it is, it inserts a punctuation comma, if not, it inserts a normal symbol comma.

The second package also defines an active math comma that tests whether or not it is followed by a digit; if it is, it inserts a normal symbol comma, if not, it inserts a punctuation comma. The second package is handier, because the typist does not need to do anything special in setting up the source L<sup>A</sup>T<sub>E</sub>X file, except

when the comma is used as a numerical list separator as in  $\forall i \in 0, 1, 2, \dots, n$  to be input as `\forall_{i \in 0, 1, 2, \dots, n}`.

### 3 A proposal for an intelligent comma

Actually I have created a macro that is simpler than the one by Alexander Rozhenko; his macro exploits a lot of `\expandafter` commands, while I think my version is simpler and more easily understood:

```
\makeatletter % not necessary when this code is in a .sty file

% 1st part
\DeclareMathSymbol{\punctcomma}{\mathpunct}{letters}{"3B}
\DeclareMathSymbol{\decimalcomma}{\mathord}{letters}{"3B}

% 2nd part
\AtBeginDocument{\mathcode'\,="8000}
{\catcode'\,=\active \gdef{\futurelet\let@token\m@thcomma}}

% 3rd part
\def\m@thcomma{\let\@tempB\punctcomma
\@tfor\@tempA:=0123456789\do{%
\expandafter\ifx\@tempA\let@token\let\@tempB\decimalcomma
\@break@tfor\fi}\@tempB}
```

As can be seen from the listing, the code is divided in three simple parts:

1. Two math symbols are defined: `\punctcomma` is the definition of a punctuating comma, while `\decimalcomma` represents the decimal separator. The second parameter of the declarative command clearly states their rôles.
2. The comma sign is made an active character in math mode: the hexadecimal code "8000 is the special active maths code that  $\text{T}_{\text{E}}\text{X}$  defines for mathematical active characters. But this activeness is deferred until the beginning of the actual document, so as not to disturb other packages, should they use other active characters, possibly including the comma. The active comma is

then defined within a group, but the definition is global so it overrides the group boundaries.

3. Eventually the real macro is defined, `\m@thcomma`. This macro first creates a temporary control sequence as an alias to the punctuating comma; then it scans a token list by means of the internal  $\text{\LaTeX}$  recursive macro `\@tfor`. This macro recursively *defines* another temporary macro with one of the tokens of the list contained within the `:=` and the `\do` delimiters (in our case the ten decimal digits) and then executes what appears to be the argument of the `\do` command (actually `\do` is just a delimiter). This temporary macro is first expanded in order to extract the token it contains and then is compared with the `\let@token` that the main macro had `\let` as an alias to whatever token followed the source text comma. If it is a digit, then the temporary control sequence `\@tempB` is `\let` to the decimal separator. The last operation in the definition is used only for terminating the recursiveness as soon as a digit is found in the list that matches the `\let@token` alias. At the very end the temporary control sequence `\@tempB` contains whatever was the last equivalence of this temporary control sequence, *i.e.* either one of the two comma meanings.

Nothing new, except simplicity and versatility. In fact, by making only minor changes to parts 1 or 2 of the small sequence of declarations and definitions, it is possible to change the decimal period into a decimal comma, or any other symbol that the comma or the period character is to be transformed into.

Since the `\iflanguage` test is implemented with both `babel` and `polyglossia`, one may include the `mathcode` active declaration into such a test in order to change the math code, depending on the language used.

## 4 Conclusion

The macros above play well with both *pdf $\text{\LaTeX}$*  and *x $\text{\LaTeX}$* ; with the latter even when using the declaration `unicode-math`.

The whole code amounts to a mere eight lines that may be copied into a personal macro file, or in the preamble of any document where the main language is not one of the English varieties.

I use these few lines of code in any one of my documents written in Italian; the only thing I have to pay attention to, is to insert spaces after commas in the input files when those commas separate numerical items in a list. Otherwise, the comma is intelligent enough to play the rôle it is assumed to play.

I wish that such a simple solution, or a similar better one, might eventually be taken into consideration by the maintainers of the babel and polyglossia packages, so that in due time we can forget about intelligent commas, because they will be incorporated into the language changing mechanism by default.

## References

- [1] HARALD HARDERS, “The numprint package”, in `.../doc/latex/numprint/numprint.pdf`
- [2] ALEXANDER I. ROZHENKO, “The nccomma package”, in `.../doc/latex/ncctools/nccomma.pdf`
- [3] WALTER SCHMIDT, “The package icomma” in `.../doc/latex/was/readme.1st`
- [4] JOSEPH WRIGHT, “siunitx — A comprehensive (SI) units package”, in `.../doc/latex/siunitx/siunitx.pdf`