Intelligent brackets

The \ibrackets package

Antoine Missier
antoine.missier@ac-toulouse.fr

2022/12/26, v1.1

1 Introduction

Open intervals are usually represented with parenthesis $(0, +\infty)$ but sometimes we find also square brackets $]0, +\infty[$, for example in French mathematics. When using these, spacing is often unsuitable, e.g. $x \in ]0, +\infty[$. This small package redefines brackets symbols $[\text{ and }]$ for mathematical mode to get correct spacing: $x \in ]0, +\infty[$.

Originally implemented in the \mismath package \cite{1} and also in \frenchmath \cite{2} since version 2.1, our previous redefinitions produce however incorrect spacing when the left bound of the interval begins with a sign - or +, which was then interpreted as a binary operation. Thus blank spaces surrounding the sign would have been too large. This problem was pointed out by Jean-François Burnol, and an easy solution, that has been documented, consisted to nest the operator or the left bound within a pair of braces, e.g. $x \in \{ -\}\infty, 0\}$, or use \left and \right or even \mathopen{]}).

Inspired by the \icomma package \cite{3} of Walter Schmidt, we now provide an improved bracket definition that works correctly without these pairs of curly brackets.

Let’s also mention other approaches e.g. \DeclarePairedDelimiters, a macro from the \mathtools package \cite{4}, or the interval package \cite{5} with his \interval macro. Nevertheless our solution is lighter.

2 Usage

You just have to type intervals in an easy way: $x \in ]0, \pi \cup] 2\pi, 3\pi[$ produce

\begin{align*}
x & \in ]0, \pi[ \cup]2\pi, 3\pi[ & \text{with \ibrackets,}
\end{align*}

instead of

\begin{align*}
x & \in ]0, \pi[\cup]2\pi, 3\pi[ & \text{without \ibrackets.}
\end{align*}
Generally, [ and ] symbols are not defined anymore as delimiters, but as ordinary characters. Thereby a line break could occur between the two square brackets, but it is always possible to transform them into delimiters with \left and \right.

The problem of a sign following the first bracket is solved with this package, so the example in the introduction is simply obtained with $x \in \]-\infty, 0\]$ which gives $x \in ]-\infty, 0\]$. 

*However, you don’t have to leave a space between the first bracket and the sign: e.g. $x \in \]-\infty, 0\]$ yields $x \in ]-\infty, 0\]$ with bad spacing around the minus sign. Contrariwise, when you want to write algebra on intervals then you must leave a blank space between the second bracket and the +/- operations, e.g. $[a, b] + [c, d]$ yields $[a, b] + [c, d]$ but $[a, b]+ [c, d]$ yields $[a, b] + [c, d]$. To summarize the new behavior of a bracket: it is an ordinary character, but an open delimiter when it is immediately followed by a + or - character.

### 3 Implementation

At \begin{document}, we memorize the \mathcode of the original brackets, in the \math...bracket macros, and we make the brackets in math mode active:

```latex
\AtBeginDocument{%
  \mathchardef\mathopenbracket\mathcode\[%
  \mathcode\[="8000
  \mathchardef\mathclosebracket\mathcode\]%
  \mathcode\]="8000
}\%
```

The active brackets check the next input character. If this is a - or a +, the active brackets return \mathopen with the saved \math...bracket so that no space will be added after the bracket; otherwise, \mathord\math...bracket is returned:

```latex
\{\catcode\[=\active
  \gdef{\{\futurelet\@next\sm@rtopenbracket}}
  \def\@rtopenbracket{%
    \ifx\@next- \mathopen \else
    \ifx\@next+ \mathopen \else
    \mathord \fi \fi \mathopenbracket}

\{\catcode\]=\active
  \gdef{\{\futurelet\@next\sm@rtclosebracket}}
  \def\@rtclosebracket{%
    \ifx\@next- \mathopen \else
    \ifx\@next+ \mathopen \else
    \mathord \fi \fi \mathclosebracket}
```

We could have use the internal \LaTeX\ command \ifnextchar to skip blank spaces after the bracket, and look if there is a + or - after, but then it would become tricky when you really want to follow an interval with an operation plus or minus.
References


[5] *The interval package*. Lars Madsen, CTAN, v0.4 2019/03/06.