Intelligent brackets
The ibrackets package

Antoine Missier
antoine.missier@ac-toulouse.fr

2023/07/26, v1.2

1 Introduction

Open intervals are commonly represented with parenthesis, e.g. $(0,+\infty)$ but sometimes square brackets are used, especially in French mathematics: $]0,+\infty[$. In that specific case, the space around the square brackets is often inappropriate, as in the expression $x \in]0,+\infty[$. This small package address this issue and redefines brackets symbols [and] for mathematical mode to get correct spacing: $x \in]0,+\infty[$.

Originally implemented in the mismath package [1] and also in frenchmath [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. As a result, blank spaces surrounding the sign would have been too large. This issue was pointed out by Jean-François Burnol, and an easy solution, that has been documented, consists of nesting the operator or the left bound within a pair of braces, e.g. `$x \in ]{-}\infty, 0]$`, or using `\left` and `\right` or even `\mathopen{]}`.

Inspired by Walter Schmidt's icomma package [3], we now provide an improved bracket definition that works correctly without the need for these pairs of braces.

Let's also mention other approaches, such as the `\DeclarePairedDelimiters` macro from the mathtools package [4], or the interval package [5] with its `\interval` macro. However our solution is more lightweight.

2 Usage

With the ibrackets package, you can easily type intervals. For example the code `$x\in]0,\pi[\cup]2\pi, 3\pi[$` yields

$$x \in]0,\pi[\cup]2\pi,3\pi[\quad \text{with ibrackets,}$$
$$\text{instead of} \quad x \in]0,\pi[\cup]2\pi,3\pi[\quad \text{without ibrackets.}$$

For the example in the introduction the spacing is now correct with the following simple code: `$x \in ]-\infty, 0]$`, which gives $x \in]-\infty, 0]$.

In ibrackets, the symbols [and] and are not defined by default as delimiters. Therefore, a line break could occur between the two brackets. However, it is always possible to transform them into delimiters using `\left` and `\right`.

Actually, brackets are set as "active" characters, behaving like ordinary characters in most cases. However, when a bracket is *immediately* followed by a + or - character, it becomes an open delimiter. Therefore, when the left bound contains an operator sign, *you don't have to leave a space between the first bracket and the sign*, otherwise, the spaces surrounding the operator will be too large. For example if you write `$x \in ] -\infty, 0]$` it yields $x \in]-\infty, 0]$ instead of $x \in]-\infty, 0]$. Conversely, when dealing with algebraic expressions involving intervals, *you must leave a space between the second bracket and the +/- operations* to maintain proper spacing. For instance `$[a, b] + [c, d]$` yields $[a, b] + [c, d]$ while `$[a, b]+ [c, d]$` would yield $[a, b]+[c, d]$.

3 Implementation

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

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

```

The active brackets check the next input character. If the next character 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.

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

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

We could have use the internal TEX 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

[1] *mismath – Miscellaneous mathematical macros*. Antoine Missier, CTAN, v2.0 2022/11/11.

[2] *L'extension frenchmath*. Antoine Missier, CTAN, v2.2 2022/12/15.

[3] *The icomma package for LaTeX 2ε*. Walter Schmidt, CTAN, v2.0 2002/03/10.

[4] *The mathtools package*. Morten Høgholm, Lars Madsen, CTAN, v1.21 2018/01/08.

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