Test file to accompany `mathastext` version `1.3zb` of `2023/12/29`

Two features of `mathastext` : extended scope of the math alphabets and added italic corrections

The package makes ! ? , : ; + − = ( ) [ ] < > { }, the asterisk ∗, and . / | \ # $ % & obey the math alphabet commands (this is the maximal list, some characters may have been excluded by the corresponding package options). For the characters listed first the mechanism involves a 'mathematical activation'.

As this process may create incompatibilities, it will be put into action for ! ? , : ; + − = ( ) [ ] < > only if the user makes use of the package command `\MTnonlettersobeymathxx` (and the braces necessitate `\MTexplicitbracesobeymathxx`).

It could be that one such character has been made 'active' in the entire document by some other package, typically a language definition file for the `babel` system. Here for example we have used `babel` with the `french` option, which makes the high punctuation characters ! ? : ; active throughout the document (extra spacing is put in front of the character when used in text ; no change in math but perhaps for other languages and characters this could happen, it is up to the language definition file to decide).

When `mathastext` detects that a character it wants to 'mathematically activate' is already 'active', it does not go further except if it seems that the activation was done by Babel. If the activation was done by Babel, then `mathastext` replaces the expansion of the active character in math mode by what is necessary to achieve its goal. It does not additionally mathematically activate the character ; rather it makes sure that the character is *not* mathematically active. In the present document the colon was made mathematically active by `mathtools` but this was already canceled in the preamble by `mathastext` as it was loaded later. And it is better so, because the combination `babel` (with option `frenchb`) +`mathtools` (with `centercolon`) makes `$:$` create an infinite loop !

But even if someone had mathematically activated the colon after the preamble, or after the loading of `mathastext`, this would be canceled again automatically for each inline or displayed mathematical formula (if the user does `\MTnonlettersobeymathxx`).

The conclusion with `\MTnonlettersobeymathxx` is : if some package has tried to make the character mathematically active, this will be overruled by `mathastext` ; if some package has made the character globally active, then the package wins except if it is Babel, as `mathastext` may in the latter case safely modify the action in math mode (paying attention to the fact that the character should be usable in `\label` and `\ref` in and outside of math mode).

The displayed equations next illustrate the extended scope of the math alphabets which now apply to =, −, (,), [,] (but not to the large delimiters of course). Furthermore, for testing purposes the equations were labeled using such characters, for example the last one has label `eq=7`, to check that the mathematical activation of = does not cause problems with `\label/\ref`.

$$\left\{\left(\left(\left[[\sin(a)+\cos(b)-\log(c)=\sec(d)]\right]\right)\right)\right\} \tag{1}$$

mathnormalbold : $$\left\{\left(\left(\left[\boldsymbol{[sin(a)+cos(b)-log(c)=sec(d)]}\right]\right)\right)\right\} \tag{2}$$

mathrm : $$\left\{\left(\left(\left[[\mathrm{sin(a)}+\mathrm{cos(b)}-\mathrm{log(c)}=\mathrm{sec(d)}]\right]\right)\right)\right\} \tag{3}$$

$$\textbf{mathbf :} \quad \left\{\left(\left(\left[\left[\mathbf{sin(a) + cos(b) - log(c) = sec(d)}\right]\right]\right)\right)\right\} \tag{4}$$

$$\textit{mathit :} \quad \left\{\left(\left(\left[\left[\textit{sin(a) + cos(b) - log(c) = sec(d)}\right]\right]\right)\right)\right\} \tag{5}$$

$$\texttt{mathtt :} \quad \left\{\left(\left(\left[\left[\texttt{sin(a) + cos(b) - log(c) = sec(d)}\right]\right]\right)\right)\right\} \tag{6}$$

$$\mathsf{mathsf :} \quad \left\{\left(\left(\left[\left[\mathsf{sin(a) + cos(b) - log(c) = sec(d)}\right]\right]\right)\right)\right\} \tag{7}$$

Equations above are numbered 1, 2, 3, 4, and 5, 6, and 7.

a!*b*	***a!b***	a!b	**a!b**	*a!b*	`a!b`	a!b
a?*b*	***a?b***	a?b	**a?b**	*a?b*	`a?b`	a?b
a, *b*	***a, b***	a, b	**a, b**	*a, b*	`a, b`	a, b
a; *b*	***a; b***	a; b	**a; b**	*a; b*	`a; b`	a; b
a : *b*	***a : b***	a : b	**a : b**	*a : b*	`a : b`	a : b
a := *b*	***a := b***	a := b	**a := b**	*a := b*	`a := b`	a := b
a := *b*	***a := b***	a := b	**a := b**	*a := b*	`a := b`	a := b
a: *b*	***a: b***	a: b	**a: b**	*a: b*	`a: b`	a: b
a.b	***a.b***	a.b	**a.b**	*a.b*	`a.b`	a.b
a – *b*	***a – b***	a – b	**a – b**	*a – b*	`a - b`	a – b
a + *b*	***a + b***	a + b	**a + b**	*a + b*	`a + b`	a + b
a = *b*	***a = b***	a = b	**a = b**	*a = b*	`a = b`	a = b
a < *b*	***a < b***	a < b	**a < b**	*a < b*	`a < b`	a < b
a > *b*	***a > b***	a > b	**a > b**	*a > b*	`a > b`	a > b
< *x*, *y* >	***< x, y >***	< x, y >	**< x, y >**	*< x, y >*	`< x, y >`	< x, y >
<*x*, *y*>	***<x, y>***	<x, y>	**<x, y>**	*<x, y>*	`<x, y>`	<x, y>
$\langle x, y \rangle$	$\langle \boldsymbol{x, y} \rangle$	$\langle$x, y$\rangle$	$\langle \mathbf{x, y} \rangle$	$\langle x, y \rangle$	$\langle$`x, y`$\rangle$	$\langle$x, y$\rangle$
a/*b*	***a/b***	a/b	**a/b**	*a/b*	`a/b`	a/b
*a**b*	***a\b***	a\b	**a\b**	*a\b*	`a\b`	a\b
a \ *b*	***a \ b***	a \ b	**a \ b**	*a \ b*	`a \ b`	a \ b
a|*b*	***a|b***	a|b	**a|b**	*a|b*	`a|b`	a|b
a | *b*	***a | b***	a | b	**a | b**	*a | b*	`a | b`	a | b
(*a*, *b*)	***(a, b)***	(a, b)	**(a, b)**	*(a, b)*	`(a, b)`	(a, b)
[*a*, *b*]	***[a, b]***	[a, b]	**[a, b]**	*[a, b]*	`[a, b]`	[a, b]
{*a*, *b*}	***{a, b}***	{a, b}	**{a, b}**	*{a, b}*	`{a, b}`	{a, b}

The question mark has been made active by `babel+frenchb`. `mathastext` has imposed in math mode its ways (now `$\mathbf{???}$` gives **???**). As the extra spacing is added by `frenchb` only in text, we had to use the math alphabet to check that indeed `mathastext` overruled Babel.

To double-check we will now make ? mathematically active : `\mathcode'?="8000`. This is a sure cause for disaster normally with Babel (don't do this at home without `mathastext`!). But here with `$?$` no bad surprise (infinite loop!) awaits us : just ?.

Let's take some other character, for example the opening parenthesis, and make it catcode active : `\catcode'(=\active \def ({X}`. Let's try the input `( and $($`. This gives X and *X*. We

see that `mathastext` does not attempt to modify the definition of the active character, as this activation was not done via the `babel` services. We now revert the parenthesis to catcode other (but maintain `\def ({X}` as definition of its active version), and then make it mathematically active using the command `\mathcode'(="8000`. If we try `$((($` we see that the parenthesis is not converted into an X : (((. The mathematically active character was overruled by `mathastext`.

Issuing `\MTnonlettersdonotobeymathxx` we do get the X's from the input `$((($` : *XXX* This shows that `mathastext` now does not modify in math mode the non-letter `(`.

We defined in the preamble of the document a `mathastext`-enhanced math version (named `upright`) having the Latin letters upright in math mode. Let's switch to it :

`\MTversion{upright}`

With a font which is neither italic nor slanted, `mathastext` automatically inserts italic corrections for better positioning of the subscript : `$f_i^i$` gives f_i^i. After `\MTnoicinmath` which turns off this feature, the same input gives f_i^i, which is different. [1]

Again with italic corrections on (`\MTicinmath`) `$f_{abc}^{def}$` gives f_{abc}^{def}, and here is another one : $f_u^{f_{abc}^{def}}$. Without italic corrections : f_{abc}^{def}, and respectively $f_u^{f_{abc}^{def}}$.

`mathastext` does not add these italic corrections inside arguments of math alphabets, as this would prevent the formation of ligatures : ff, ff, *ff*, **ff**, `ff` (no ligature in teletype) and ff. [2]

a!b	**a!b**	a!b	**a!b**	*a!b*	`a!b`	a!b
a?b	**a?b**	a?b	**a?b**	*a?b*	`a?b`	a?b
a, b	**a, b**	a, b	**a, b**	*a, b*	`a, b`	a, b
a; b	**a; b**	a; b	**a; b**	*a; b*	`a; b`	a; b
a : b	**a : b**	a : b	**a : b**	*a : b*	`a : b`	a : b
a := b	**a := b**	a := b	**a := b**	*a := b*	`a := b`	a := b
a := b	**a := b**	a := b	**a := b**	*a := b*	`a := b`	a := b
a: b	**a: b**	a: b	**a: b**	*a: b*	`a: b`	a: b
a.b	**a.b**	a.b	**a.b**	*a.b*	`a.b`	a.b
a – b	**a – b**	a – b	**a – b**	*a – b*	`a - b`	a – b
a + b	**a + b**	a + b	**a + b**	*a + b*	`a + b`	a + b
a = b	**a = b**	a = b	**a = b**	*a = b*	`a = b`	a = b
a < b	**a < b**	a < b	**a < b**	*a < b*	`a < b`	a < b
a > b	**a > b**	a > b	**a > b**	*a > b*	`a > b`	a > b
< x, y >	**< x, y >**	< x, y >	**< x, y >**	*< x, y >*	`< x, y >`	< x, y >
<x, y>	**<x, y>**	<x, y>	**<x, y>**	*<x, y>*	`<x, y>`	<x, y>
⟨x, y⟩	**⟨x, y⟩**	⟨x, y⟩	**⟨x, y⟩**	*⟨x, y⟩*	`⟨x, y⟩`	⟨x, y⟩
a/b	**a/b**	a/b	**a/b**	*a/b*	`a/b`	a/b
a\b	**a\b**	a\b	**a\b**	*a\b*	`a\b`	a\b
a \ b	**a \ b**	a \ b	**a \ b**	*a \ b*	`a \ b`	a \ b
a|b	**a|b**	a|b	**a|b**	*a|b*	`a|b`	a|b
a | b	**a | b**	a | b	**a | b**	*a | b*	`a | b`	a | b
(a, b)	**(a, b)**	(a, b)	**(a, b)**	*(a, b)*	`(a, b)`	(a, b)
[a, b]	**[a, b]**	[a, b]	**[a, b]**	*[a, b]*	`[a, b]`	[a, b]
{a, b}	**{a, b}**	{a, b}	**{a, b}**	*{a, b}*	`{a, b}`	{a, b}

1. last time I tried, this only worked with PDFLaTeX, not with LuaLaTeX or XeTeX.

Changed ! 2. Prior to 1.3i, italic corrections were added to the \mathnormal arguments.