Next steps for breqn

Morten Høgholm

Technical University of Denmark

TUG 2009, Notre Dame University

Outline

1. What breqn does

A quick example

-
  ```
  \begin{dmath*}
  f(x) = \sum_{k=0}^{\infty}
         \frac{f^{(k)}(c)}{k!}(x-c)^{k}
  = f(c)+f'(c)(x-c) +\frac{f''(c)}{2!}(x-c)^{2}
  + \frac{f^{(3)}(c)}{3!}(x-c)^{3}+\cdots
  \end{dmath*}
  ```
- $$\begin{aligned} f(x) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!}(x-c)^{k} \\ &= f(c) + f'(c)(x-c) + \frac{f''(c)}{2!}(x-c)^{2} \\ &\quad + \frac{f^{(3)}(c)}{3!}(x-c)^{3} + \cdots \end{aligned}$$

Another example

-
  ```
  \begin{dgroup*}
  \begin{dmath}
  f(x) = \sum_{k=0}^{\infty}
         \frac{f^{(k)}(c)}{k!}(x-c)^{k}
  = f(c)+f'(c)(x-c) +\frac{f''(c)}{2!}(x-c)^{2}
  + \frac{f^{(3)}(c)}{3!}(x-c)^{3}+\cdots
  \end{dmath}
  \begin{dmath*}
  E=mc^2
  \end{dmath*}
  \end{dgroup*}
  ```

Another example, cont.

- $$\begin{aligned} f(x) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!}(x-c)^k \\ &= f(c) + f'(c)(x-c) + \frac{f''(c)}{2!}(x-c)^2 \\ &\quad + \frac{f^{(3)}(c)}{3!}(x-c)^3 + \cdots \\ E &= mc^2 \end{aligned} \tag{1}$$

How breqn works

- Categorizing math symbols into classes
- Spacing classes — same as TEX.
- Breaking classes. Left delimiters and right delimiters should act differently.
- Making every math character a macro: `\mathcode"8000`

What about amsmath, memoir, you name it?

Many small bug fixes to make it not break everything else, so. . .

- Good news! It works with amsmath.
- And memoir.
- And most other things.
- But beware of catcodes!

Problems in use

- Not all environments work as advertised
- Lacking QED support
- Interference with other packages changing document catcodes.
- Lacking proper manual overrides.

Tag placement

The tag placement is currently not exactly as desired.

- The original implementation would center tag vertically

$$k = \frac{1}{\sqrt{\frac{2}{3}\pi}} \tag{2}$$

- New algorithm tries to be a little smarter

$$k = \frac{1}{\sqrt{\frac{2}{3}\pi}} \tag{2}$$

- However, this is a little more complicated. . .

The line breaking

- It works most of the time
- But can occasionally produce sub-optimal results
- The algorithm just isn't perfect.
- Needs an inspection/more detailed specification.
- Why do everything yourself? The new version of Presentation MathML contains many ideas as to how to do this.
- Get it to work properly with the paragraph shapes within LaTeX.

Implementing the line breaking

- Currently spaghetti code
- Needs to be rewritten into a more functional/readable style
- This ought to ease maintenance!
- And also porting to other languages.

Unicode

- Difficult!
- Many things have to change under the hood
- But don't worry – it's already been done in the development version

Accessibility

- Also difficult!
- But since breqn reflows a math expression. . .
- Let's add pdf tagging!
- Hope to commence this work soon.

Other nice things to have

- Breaking of binary operators in Slavic style
- Automatic ligatures `sin` $\longrightarrow$ $\sin$.
- Automatic scaling of delimiters, i.e., do what nath does.
- In short: To make it simple for the user!
- And that it doesn't break other packages.