Expositionary Arrows: A Demo

Robin Truax

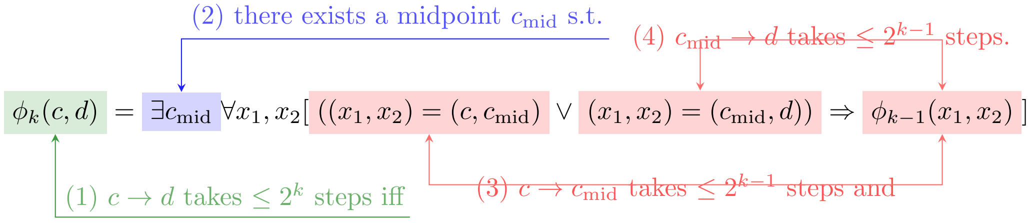
June 2022

1 Introduction

In this document, I'm going to show you how to turn long, hard-to-read equations like this

$$\phi_k(c,d) = \exists c_{\text{mid}} \forall x_1, x_2((x_1,x_2) = (c,c_{\text{mid}}) \lor (x_1,x_2) = (c_{\text{mid}},d)) \Rightarrow \phi_{k-1}(x_1,x_2)].$$

into long, easy-to-read equations like this


(2) there exists a midpoint c_{mid} s.t.
(4) $c_{\text{mid}} \to d$ takes $\leq 2^{k-1}$ steps.
$\phi_k(c,d) = \exists c_{\text{mid}} \forall x_1, x_2[((x_1,x_2) = (c,c_{\text{mid}}) \lor (x_1,x_2) = (c_{\text{mid}},d)) \Rightarrow \phi_{k-1}(x_1,x_2)]$
(1) $c \to d$ takes $\leq 2^k$ steps iff
(3) $c \to c_{\text{mid}}$ takes $\leq 2^{k-1}$ steps and

without using an exorbitant amount of code.

Indeed, the former equation requires the following LaTeX

Listing 1: LaTeX for Equation Without Annotation

```
\begin{equation*}
   \phi_k(c,d) = \exists c_{\text{mid}} \forall x_1,x_2 ((x_1,x_2) = (c,c_{\text{mid}})
      ↪ \lor (x_1,x_2) = (c_{\text{mid}},d)) \Rightarrow \phi_{k-1}(x_1,x_2)].
\end{equation*}
```

whereas the second equation requires the following LaTeX:

Listing 2: LaTeX for Equation With Annotation

```
\expspacing
\begin{equation*}
   \expnode{formula1}{$\phi_k(c,d)$}{ForestGreen} = \expnode{formula2}{$\exists c_{\text
      ↪ {mid}}$}{blue} \forall x_1,x_2 [\expnode{formula3}{$((x_1,x_2) = (c,c_{\text{
      ↪ mid}})$}{red} \lor \expnode{formula4}{$(x_1,x_2) = (c_{\text{mid}},d))$}{red} \
      ↪ Rightarrow \expnode{formula5}{$\phi_{k-1}(x_1,x_2)$}{red}].
\end{equation*}
\exparrow{formula1}{(1) $c \to d$ takes $\leq 2^k$ steps iff}{ForestGreen}{dr}
\exparrow{formula2}{(2) there exists a midpoint $c_{\text{mid}}$ s.t.}{blue}{ur}
\exparrowdual{formula3}{formula5}{(3) $c \to c_{\text{mid}}$ takes $\leq 2^{k-1}$ steps
   ↪ and}{red}{d}
\exparrowdual{formula4}{formula5}{(4) $c_{\text{mid}} \to d$ takes $\leq 2^{k-1}$ steps
   ↪ .}{red}{u}
\expspacing
```

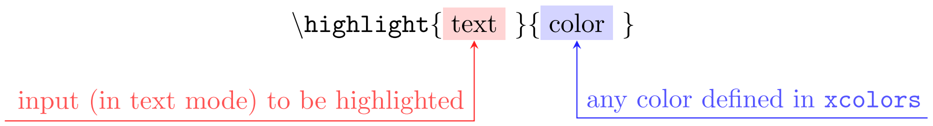
Yes, there is substantially more code in the second listing, but given that the example contains six highlighted sections, six arrows, four labels with lines to carry them, all across three different colors, the alternative (defining all of it manually in TikZ) would be prohibitively difficult for most users. Furthermore, moving around the arrows or editing the contents of the equation or the labels is much easier using this package.

2 Tutorial

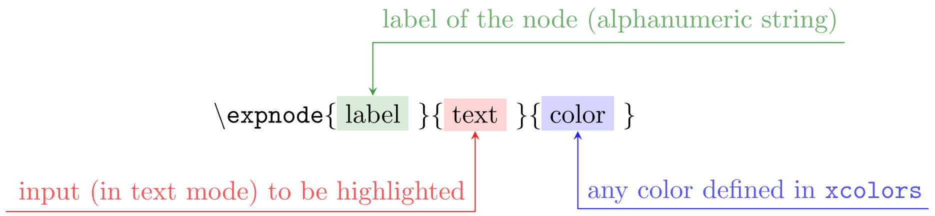
This package consists of the following commmands:

`\highlight` `\expnode` `\exparrow` `\exparrowdual` `\expspacing`

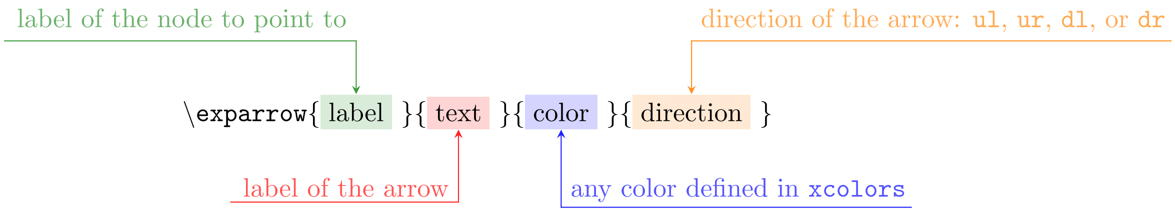
The first of these, `\highlight`, is the simplest. This command colors the background behind any text. For example, `\highlight{hello}{red}` produces hello. The colors are automatically loaded by the `exparrows` package, which loads `xcolors` (both the default colors and the additional colors provided by the options `usenames`, `dvipsnames`, and `svgnames`). Formally, the syntax for the `highlight` command is as follows:


\highlight{ text }{ color }
input (in text mode) to be highlighted
any color defined in xcolors

The next of these, `\expnode`, is used to "lock on" arrows. The `expnode` command is essentially the same as the `highlight` command, except that it has an additional input for the *name* of the expnode (which can later be referenced in an `\exparrow` or `\exparrowdual` command). For example, to highlight the text in the above explanation of `highlight`, instead of using `\highlight`, I used the code `\expnode{11}{text}{red}` and `\expnode{12}{color}{blue}`, respectively. Formally, the syntax for the `expnode` command is as follows:

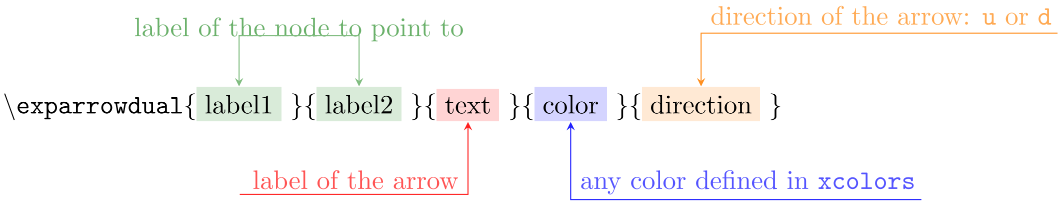

label of the node (alphanumeric string)
\expnode{ label }{ text }{ color }
input (in text mode) to be highlighted
any color defined in xcolors

Now, the eponymous command of the package, `\exparrow`, is used to produce the arrows that we have been scattering throughout the document. The syntax for the `exparrow` command is as follows:


label of the node to point to
direction of the arrow: ul, ur, dl, or dr
\exparrow{ label }{ text }{ color }{ direction }
label of the arrow
any color defined in xcolors

To elaborate on the final parameter, an arrow marked with `ul` will travel up-and-to-the-left. Analogously, arrows marked with `ur`, `dl`, and `dr` travel up-and-to-the-right, down-and-to-the-left, and down-and-to-the-right respectively. For example, to produce the red arrow above, I first produced a node with `\node{32}{text}{red}`, and then I pointed to it with `\exparrow{32}{label of the arrow}{red}{dl}`.

Now, sometimes one would like to mark more than one node with the same arrow. For this, the command `\exparrowdual` is used. The syntax for this arrow is extremely similar:


label of the node to point to
direction of the arrow: u or d
\exparrowdual{ label1 }{ label2 }{ text }{ color }{ direction }
label of the arrow
any color defined in xcolors

The only real difference, besides that two labels are used, is that there are now only two options for direction: `u` for “up” and `d` for “down”. For example, to produce the dual green arrow above, I first produced two nodes with `\expnode{41}{label1}{ForestGreen}` and `\expnode{42}{label1}{ForestGreen}`, and then pointed to them with `\exparrowdual{41}{42}{label of the expnode to point to}{ForestGreen}{u}`.

Now, the final command is the `\expspacing` command, which has no options. After all of the expositionary arrows have been defined, the user *must* use the `\expspacing` command either above the block of equations/arrows (if any arrows pointing up have been defined) or below the block of equations/arrows (if any arrows pointing down have been defined), or both. This command will automatically add spacing necessary to avoid overlap between following text and the equation. The only caveat is that the user must place a newline between the command and any preceding or following paragraphs to ensure that the spacing works. The confused reader is invited to look at the source for this documentation, which contains ample complete examples and is included with this PDF.

Lastly, a command which I recommend using when helpful is the `\vphantom` command. This command produces a zero-width vertical box the height of its argument, and can be used to ensure that nodes have the same height. Consider, for example, the following examples:

`exp` `arrows`   `exp` `arrows`

The former example is produced using the code

`\expnode{vp1}{\texttt{exp}}{ForestGreen}` and `\expnode{vp2}{\texttt{arrows}}{red}`

whereas the latter is produced using the code

`\expnode{vp3}{\texttt{exp}}{ForestGreen}` and
`\expnode{vp4}{\texttt{arrows\vphantom{exp}}}{red}`.

3 Final Notes

This project is an adaptation of the examples shown by Sibin Mohan on his Twitter here. The tweet also contains a link to his Github, from which this package was adapted. This package does have its flaws: the necessity of the `\expspacing` command, for one, and strange behavior when the arrows overlap with page borders, for another. Nonetheless, it has proved to be a helpful tool for writing readable math papers in LaTeXand making complex equations more accessible, and I hope that it is helpful for you too.