Typesetting Math, Part 3

Math accents

Just one quick thing to note: the `\hat` accent is pretty wimpy: $\hat{a}$. I recommend using `\widehat`: $\widehat{a}$.

The `\stackrel` command can be nice for various relational notations. For example, CLRS uses the following command:

```
\newcommand{\path}[1]{\stackrel{#1}{\leadsto}}
```

In this way, when we write about a path named p that goes from vertex u to vertex v, we type the math-mode expression `$u \path{p} v$`, which produces $u \stackrel{p}{\leadsto} v$.

Spacing

Spacing in math mode can be rather subjective. Lamport gives examples in which it is quite clear that you need to add or remove a little space, e.g., $\iint z\,dx\,dy$ rather than $\int\int zdxdy$.

But here's a place where a subjective opinion is in order. Consider an expression that multiplies a function value by a scalar: $cf(x)$. That looks OK. Would it look better with some space? You make the call: $c\,f(x)$, or even $c\;f(x)$. When we were writing the analysis of disjoint-set union, I thought that $O(m\alpha(n))$ ran the m and α together, so I added a medium space: $O(m\:\alpha(n))$.

Another place in which it pays to add space is when multiplying fractions in a display. Consider the following:

$$\frac{ab}{cd}\frac{ef}{gh} \; .$$

These fractions are too darned close to each other. We can either add some space or put in the dot:

$$\frac{ab}{cd}\,\frac{ef}{gh} \qquad \text{or} \qquad \frac{ab}{cd}\cdot\frac{ef}{gh} \; .$$

If you are unfortunate enough to have a long subscript or superscript in a summation-like formula, you could get more space than you want. Here's something from page 1118 of CLRS:

$$\sum_{S\subseteq\{1,2,\ldots,n\}:|S|=k} \!\!\!\!\!\!\!\!\!\! \Pr\{A_S\} \; .$$

That looks pretty good. Here it is without the negative space:

$$\sum_{S\subseteq\{1,2,\ldots,n\}:|S|=k} \Pr\{A_S\} \; .$$

The subscript under the summation claims too much horizontal space.

Multiletter variables

The italic font used for math is different than the italic font used for regular text. For single-letter variables names, you would never notice the difference; can you distinguish n from *n*? But you'll notice it for multiletter names. Compare $color$ vs. *color*. You can see a difference in spacing between 'l' and 'o'. It becomes even more noticeable in words that have ligatures: $wiffle$ vs. *wiffle*.

If you're going to use multiletter variable names, do not just typeset them directly in math mode. Put them in math mode, but use the `\mathit` command: `$\mathit{wiffle}$`.

There still is a potential problem. Some of the variable names in CLRS have hyphens. In math mode, a hyphen is considered to be a minus sign. So `$\mathit{heap-size}$` typesets as $\mathit{heap-size}$. The clrscode package has a macro `\id` that, in addition to using the correct italic font, also interprets hyphens as hyphens. Thus, `$\id{heap-size}$` typesets as *heap-size*.

The clrscode package also provides macros for typesetting other names in CLRS style:

- `\proc` for procedure names in small caps: the source `$\proc{Matrix-Multiply}$` typesets as MATRIX-MULTIPLY.
- `\const` for constant names in small caps: the source `$\const{red-and-black}$` typesets as RED-AND-BLACK.
- `\func` for fixed function names in roman: `$\func{in-degree}$` typesets as in-degree.

Each of these macros is set up so that it may be used either in or out of math mode.