David Kastrup

Iterating with roman numerals

Appendix D in the TeXbook has the task of defining `\asts` as a macro containing `\number\n` copies of an asterisk. The solutions in the TeXbook are not really fun. Here is one that is all of fun, efficient and simple:

```
\def\asts#1{\if#1m*\expandafter\asts\fi}
\edef\asts{\expandafter\asts\romannumeral\number\n 000\relax}
```

Now for something more general: we want a macro `\replicate` that gets a number in its first argument and arbitrary tokens in its second argument and expands to the given number of repeated token strings.

It is surprisingly hard to pass *both* the shrinking string of `m` as well as the argument to repeated in a useful way into the expanding first macro, and the reader is advised to try it. What I came up with was

```
\long\def\gobble#1{}
\long\def\xii#1#2{\if#2m#1\expandafter\xii\else\expandafter\gobble\fi{#1}}
\long\def\xiii#1\relax#2{\xii{#2}#1\relax}
\def\replicate#1{\expandafter\xiii\romannumeral\number\number#1 000\relax}
```

A somewhat wittier variant that takes its toll on the semantic nest size would be

```
\def\recur#1{\csname rn#1\recur} \long\def\rnm#1{\endcsname{#1}#1}
\long\def\rn#1{}
\def\replicate#1{\csname rn\expandafter\recur
  \romannumeral\number\number#1 000\endcsname\endcsname}
```

Of course, if we are leaving the area of TeX compatibility and take a look at what we can do with ε-TeX, we arrive at the boring

```
\def\replicate#1#2{\ifnum#1>0 #2%
  \expandafter\replicate\expandafter{\number\numexpr#1-1}{#2}\fi}
```