

Paweł Jackowki

\badness badness

\badness parameter is supposed to tell something about box stretch or shrinkage, or more precisely, about how far the available stretchability or shrinkability has been used. \badness=0 means no stretch or shrink used, 100 means the entire flexibility consumed, more then 100 denotes underfull or overfull box (the later with \badness equal 1 000 000). So \badness certainly says something about how the content fits the box, but one may be surprised by fine artifacts of the actual algorithm. Lets take a macro that measures \vbox \badness, taking the actual and the available stretch as arguments.

```
\def\test#1#2{% <actual_stretch> <available_stretch>
  \setbox0\vbox to\dimexpr#1{\vskip0pt plus\dimexpr#2}%
  \immediate\write16{badness=\the\badness, \the\ht0}}
```

```
\test{100pt-1sp}{100pt} % \badness=99
\test{100pt}{100pt} % 100
\test{100pt+1sp}{100pt} % 100
\test{100.3366pt}{100pt} % 100
\test{100.3367pt}{100pt} % 101
```

Note that we never get \badness equal 100 if the actual stretch is even 1sp smaller then the available stretch. But T_EX seems to ignore the stretch abuse around 0.0033 and less.

OK so far. Now lets take such insignificantly underfull box of 299sp actual, and 298sp available stretch. We will scale those values with integer in factor.

```
\def\scale#1{\test{299sp*#1}{298sp*#1}}
\scale{1} % 100
\scale{2} % 100
\scale{3} % 100
...
\scale{24182} % 101
\scale{24184} % 101
...
\scale{24948} % 100
```

\badness varies although we keep the stretch ratio fixed!

All because heuristic \badness formula (`tex.web`, line 2320):

```
...
begin if t=0 then badness:=0
else if s<=0 then badness:=inf_bad
else
  begin if t<=7230584 then r:=(t*297) div s {$297^3=99.94\times2^{18}$}
  else if s>=1663497 then r:=t div (s div 297)
  ...
```