From the president

Arthur Rosendahl

A few months ago, an article[1] with the intriguing title "Inconsistencies in TEX-Produced Document" was making the rounds. It had appeared in the proceedings of the 33rd International Symposium on Software Testing and Analysis, organised in September 2024 under the auspices of the ACM. The two authors in short presented a large-scale study of over 400 TEX documents that they compiled with the three TEX engines pdfTEX, XETEX and LuaTEX, as well as successive versions of TEX Live from 2020 and 2023, and noted the differences in output—which they dubbed inconsistencies. They developed an automated system to that effect. They found that the vast majority of documents yielded different results, including no result, when compiled with the different TEX engines; and there were considerable differences between the successive versions of TEX Live too. The authors went on to discuss what they called the root causes for these differences; in particular, they uncovered and reported two new bugs in the process, as well as five old ones that had been fixed independently. Most differences were in line breaks, image placement, and number of pages.

These facts will come as no surprise to experienced users of TEX, and it might even seem naïve to dedicate a whole study to this; but it is interesting to have data on it, and reporting bugs is certainly always welcome. Is that the final word, then, or is there more to it?

It seems to me that there is. Anyone who has moved in TEX circles long enough will have heard stories of old TEX documents that still compiled perfectly decades later, to the exact same typeset result, etc., etc. These stories are usually told in a way that hints at their applying to *any* document composed with TEX.

It is hard to pinpoint anything that can be proved wrong, in particular because actual documents are never exhibited (I do sometimes ask); but these anecdotes have made me increasingly uncomfortable over time as they seem to imbue TEX with almost magical properties, that makes anything it touches immune to the passage of time. Sometimes that conviction is explicit: I have, for example, had occasion to offer my help to a mailing list user who wanted to compile a 15-year-old document and ran into issues. The main one was that that document contained images included with a command that relied, via the use of a `\special`, on an unmaintained DVI driver called `dvi2ps` which I had never heard of at that point. I pointed the user to the `graphicx` package and offered to replace all the offending commands in his document myself. But that wouldn't do. The document needed to be compiled as is, he said. He had been told this should work. I was in a whimsical mood, so I turned to CTAN where I found the source code of `dvi2ps`, in C, and tried to compile it—only to find that the compiler was baulking at some particular construct (I can't remember what) and refused to produce an executable. I persevered and found a way to replace *those* commands to make the source code acceptable to a then-current C compiler, and emailed the user back. It really is very simple, I said, go to CTAN and do this and this and that to compile `dvi2ps` (again, I can't remember the details), and voilà! You can use the obsolete `\special` in your document. At that point the user admitted that, perhaps, it was after all simpler to just use `\includegraphics` in place of the old `\special` to include images, of which there were three in the whole document.

Another discussion touched on how to input the en dash '–' and em dash '—' in XETEX and LuaTEX. As most readers will know, these characters were traditionally typed as `--` and `---`, and were implemented as ligatures in the common fonts; that is to say, the program TEX itself wasn't doing anything in particular and was only setting the existing ligatures, just like typing `fi` will yield 'fi' in Computer Modern. This created a conundrum when XETEX came along, since it suddenly became possible to easily use many more fonts, most of which did not have those ligatures. Hence, typing `--` would show something like '--' in the printed output.

Technical solutions were to be found, and a decision also needed to be made to respond to users' expectations, but what struck me was this comment, made by a contributor in the course of the discussion: "in the last 3 decades, essentially any font usable with TEX ... had the ligatures, so the statement in *The TEXbook* that `---` is the way to get an em dash was/is true with virtually all fonts in use in pdfTEX. So it is not just a quirk of Computer Modern fonts, it is largely a feature of the ecosystem ... and as such it is a pity in my humble opinion if it becomes a breaking change when processing a document with a Unicode engine."

I can agree with the premise, but the conclusion is most curious: as mentioned, there was *no change* in that respect in either XETEX or LuaTEX, and if anything, what broke was the ecosystem, that overnight saw the addition of thousands more readily

[1] doi.org/10.1145/3650212.3680370

doi.org/10.47397/tb/46-3/tb144pres

available[2] fonts. The person who wrote so was a very experienced LaTeX user whose understanding of the technical issues is beyond any doubt, so it was surprising to see him describe the situation in that way, as if the programs' behaviour had changed.

The last example involves the Master himself, Don Knuth, and my own reaction to something he said on a public mailing list quite a while ago. The year was 2007, and he was installing the then-latest version of TeX Live on a new computer, asking for help and reporting on his progress as he went along. In his last email[3] he said that the installation had been successful and that he had needed to make some adjustments to the manuscript of *The Art of Computer Programming* because, out of the more than 1300 illustrations—made using MetaPost—165 had slightly different bounding boxes with the new TeX installation. Ultimately, the differencen were all caused by the removal of a single bug: the bounding box was calculated incorrectly before, now it was correct. However, this had a knock-on effect on page breaks, and Knuth needed to change the manuscript in four places to avoid bad breaks. He concludes with "So my worries were well founded; tiny changes to spacing can indeed have drastic consequences."

At that point, I don't mind admitting that I had something of a cognitive dissonance, and distinctly remember thinking "Knuth doesn't like it when we fix bugs". I wasn't the only one, since the person who fixed the bug told me, years later, that he had felt embarrassed too. Yet, there was no reason to be: in that very message, Knuth expresses his thanks and even writes "But of course I can't complain, because MetaPost now works correctly at last."

It is time to admit it: we have been haunted by the spectre of compatibility for too long. There is nothing magical at play, it takes a lot of hard work to achieve real compatibility, and sometimes breaking changes happen outside of our sphere. Just before sending his email about the MetaPost bugfix, Knuth sent another email where he wrote that he "has promised thousands of people that *their* books and papers and other documents will still typeset the same in the year 2100 as they were in the year 2000, if they are done with the frozen parts of TeX (plain TeX)". This is certainly true, and the volunteers that develop TeX distributions do their best to ensure that that promise continues to be fulfilled; but most users of TeX-based systems *today*, who number in the millions, use and need extensions and packages that go far beyond the frozen parts of TeX. They cannot be expected to understand the difference between the frozen and updated parts of TeX, or have even heard of the distinction.

In those circumstances, it is not surprising that researchers would want to study the TeX ecosystem as if it was immutable. The authors of the article mentioned above quote two sources to illustrate this, one from the TUG website and the other from the TeX (formerly UK-TUG) FAQ, neither of which make the point that TeX engines should behave the same across all distributions very forcefully.[4] They must have got that notion from somewhere else, such as discussions like the ones I've shared in this column.

By the way, I didn't mention what percentage of the documents they studied did produce the exact same output across engines: it was 0.2%. Out of the 432 documents they picked randomly from arXiv, 99.8% showed differences between pdfTeX and XeTeX.

In the end, what makes the strength of TeX is the versatility of its algorithms and its remarkable adaptability: precisely the opposite of a frozen program—which still exists and is useful for users who need such a feature. I no longer believe tales of "my wife's PhD thesis still compiles today *exactly* like it did in the 80s",[5] but I know for a fact that if a 30-year-old TeX document were found anywhere, it would probably be straightforward to adapt it to a current distribution and have it compile to something that is very similar to the original output. With other systems, this would hardly be possible, or require an expensive upgrade, or some other hurdle that would be hard to overcome. That, not some (nonexistent) magical and all-encompassing backward compatibility, is the strength of TeX.

◇ Arthur Rosendahl
Uppsala University
TUG president
president (at) tug (dot) org

[2] To be excruciatingly correct, it was already *possible* to use system fonts before, for example with pdfTeX thanks to `ttf2afm`, and even before that, by converting fonts to Type 1 and generating the necessary TFM files, etc.; but XeTeX, and later LuaTeX, made it easier by orders of magnitude since all that was needed was essentially a single command in the TeX source, without touching any font files at all. Those users who did go to the trouble of setting up new fonts for use with TeX would usually know of the dash ligatures, and either avoid or emulate them.

[3] Sent by his associate Maggie McLoughlin. The full archive of his messages can be found at `tug.org/pipermail/tex-k/2007-November` and I encourage everyone to read it.

[4] References 4 and 1 in the article. The former, extracted from `tug.org/whatis`, doesn't actually make that point at all, as the authors implicitly acknowledge — which just makes it more likely that they heard that idea somewhere else.

[5] I really did hear that! It was at a restaurant during a TUG conference, so it was hard to challenge the speaker to produce said thesis and check.