The state of $\mathcal{X}$\TeX

Arthur Reutenauer

Abstract

$\mathcal{X}$\TeX was the first \TeX engine to support Unicode natively and was actively developed until recently, but has since then gone into maintenance mode. I will discuss avenues for future development.

0 $\mathcal{X}$\TeX & Lua\TeX

Let’s start with a quick comparison between $\mathcal{X}$\TeX and Lua\TeX, its Unicode-supporting cousin. While both are similar in their overarching goals to support modern encodings and font standards, they differ in an essential tenet of their philosophies: $\mathcal{X}$\TeX transplants a lot of additional features into the core by means of external libraries, while Lua\TeX opens up the engine by allowing large parts of it to be rewritten in the Lua scripting language (the surgical metaphor is freely borrowed from Hans Hagen, main developer of Con\TeXt and designer of Lua\TeX).

This is quite a significant difference. $\mathcal{X}$\TeX’s architecture enables it to delegate crucial tasks, notably shaping (the processes necessary to display complex scripts correctly, such as Arabic and Indic). The library currently used for that task is called HarfBuzz, and was integrated by Khaled Hosny in 2012–2013. Conversely, Lua\TeX depends only on Lua code for the same tasks, but such code has to be written, and the only person currently doing so is Hans. This means that the number of scripts supported in Lua\TeX will necessarily be limited.

On a more technical level, the core of $\mathcal{X}$\TeX still uses the original WEB code, while Lua\TeX has been rewritten in C.

1 $\mathcal{X}$\TeX + Lua\TeX

One idea to shake up $\mathcal{X}$\TeX was thus to use the code base of Lua\TeX to progressively replace the WEB functions of the $\mathcal{X}$\TeX source by their C equivalent. This would be a somewhat sounder basis for future developments. In addition, we would get Lua “for free”, although the interaction with Lua\TeX’s callbacks probably would need to be massaged quite a bit. But the prospect of taking advantage of the very large amount of work already done on Lua\TeX, its comparatively higher development pace, and the possibility of merging efforts, made it a goal worth contemplating.

I have been experimenting last winter in that direction and think this effort, that we would presumably call $\mathcal{X}$\TeX, is sustainable. Nevertheless, since it also entails considerable work, I have also been exploring other options.

2 $\mathcal{X}$ & Lua\TeX & HarfBuzz

At about the same time, Khaled was working on integrating HarfBuzz into Lua\TeX, to support more scripts. This could be a possible future for $\mathcal{X}$\TeX, but it should be noted that the situation currently is a little confused, since the ongoing effort inspired the current Lua\TeX maintainer, Luigi Scarso, to produce his own experimental version of Lua\TeX with HarfBuzz dubbed lualibu\TeX. It may thus be wise to wait for the dust to settle before deciding if that can be the future for $\mathcal{X}$\TeX. And there’s more!

3 $\mathcal{X}$ + lmtx

Another new project is the effort by Hans, always indefatigable, to overhaul Lua\TeX into a leaner engine with a different build system. This lmtx was announced on 1 April (but wasn’t an April fool’s joke) and will become the basis for the next major version of Con\TeXt. The first official release will be during the 2019 Con\TeXt meeting, two weeks from the time of writing, hence I thought that as long as I was contemplating possible futures for $\mathcal{X}$\TeX, I might as well have a look in some detail at the upcoming lmtx! HarfBuzz will not be a part of it, since Con\TeXt is using the Lua shaping code, hence a similar effort as the one mentioned in the previous paragraph would be needed.

4 Why?

Why, one might ask, bother with such considerations at all? $\mathcal{X}$\TeX already exists and in spite of some misfeatures (for example in the bidirectional character token mechanism also lacks an equivalent in Lua\TeX (I’m grateful to Henri Menke for bringing the latter to my attention during the conference). If the developments outlined in section 2 do give rise to an extended Lua\TeX engine with all of $\mathcal{X}$\TeX’s high-level capabilities, it will be time to bridge the gap by adding all the small missing bits and pieces, and merge the two projects together (which obviously is my ultimate goal). Until such time, however, experiments are in order.

\begin{itemize}
  \item Arthur Reutenauer
  \texttt{arthur.reutenauer (at) normalesup dot org}
\end{itemize}