
The state of X_ƒTeX

Arthur Reutenauer

Abstract

X_ƒTeX 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 X_ƒTeX & LuaTeX

Let's start with a quick comparison between X_ƒTeX and LuaTeX, 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: X_ƒTeX transplants a lot of additional features into the core by means of external libraries, while LuaTeX 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 ConTeXt and designer of LuaTeX).

This is quite a significant difference. X_ƒTeX'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, LuaTeX 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 LuaTeX will necessarily be limited.

On a more technical level, the core of X_ƒTeX still uses the original WEB code, while LuaTeX has been rewritten in C.

1 X_ƒTeX + LuaTeX

One idea to shake up X_ƒTeX was thus to use the code base of LuaTeX to progressively replace the WEB functions of the X_ƒTeX 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 LuaTeX'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 LuaTeX, 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 X_ƒLuaTeX, is sustainable. Neverthe-

less, since it also entails considerable work, I have also been exploring other options.

2 X_ƒ + LuaTeX + HarfBuzz

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

3 X_ƒ + lmtx

Another new project is the effort by Hans, always indefatigable, to overhaul LuaTeX 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 ConTeXt. The first official release will be during the 2019 ConTeXt meeting, two weeks from the time of writing, hence I thought that as long as I was contemplating possible futures for X_ƒTeX, I might as well have a look in some detail at the upcoming `lmtx!` HarfBuzz will not be a part of it, since ConTeXt 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? X_ƒTeX already exists and in spite of some misfeatures (for example in the bidirectional models), it has no serious bugs. The absence of new development obviously means that it is very stable.

However, no program keeps being maintained in the long run just by staying exactly identical (TeX90 being a lone exception). X_ƒTeX still has essential features that are unique in the TeX world: complex scripts is the most important one; and the inter-character token mechanism also lacks an equivalent in LuaTeX (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 LuaTeX engine with all of X_ƒTeX'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.

◇ Arthur Reutenauer
 arthur.reutenauer (at) normalesup dot org