

Oriental T_EX by a dummy

Abstract

This article is converted from the slides presented at the conference.

What is Oriental T_EX

- It is a project by Idris Samawi Hamid, Taco Hoekwater and Hans Hagen.
- The project started shortly after we started the LuaT_EX project.
- It boosted development of LuaT_EX thanks to a grant that paid for coding LuaT_EX.
- It also boosted the development of ConT_EXt MkIV and was a real good torture test for OpenType font support.
- This project also costs us a whole lot of time.
- The main objective is to let T_EX typeset high quality (traditional) Arabic.
- Closely related to this is to extend ConT_EXt capabilities to deal with advanced critical editions.
- In the meantime a high quality Arabic OpenType font has become part of the package.

How we proceed

- Of course we were a bit too optimistic when setting the time schedule for this project.
- This is because we need to have quite some bits and pieces in place beforehand.
- For instance, making the font and perfecting OpenType support involves a lot of trial and error and testing.
- This is mostly due to lack of specifications, benchmarks and limitations in tools.
- We have identified the needs for critical editions but have postponed some of that till we have opened up more of LuaT_EX.
- We are also getting a better picture of what is needed for advanced right-to-left typesetting, especially in mixed directionality.

Simple OpenType fonts

- In Latin scripts we have mostly one-to-one and many-to-one substitutions.
- This can happen in sequence (multiple passes).
- Sometimes surrounding characters (or shapes) play a role.
- In some cases glyphs have to be (re)positioned relative to each other.
- Often the substitution logic is flawed and it is assumed that features are applied selectively (DTP: select and apply).
- Of course this is unacceptable for what we have in mind.

The Oriental T_EX approach

- We put as much logic in the font as possible, but also provide a dedicated paragraph builder (written in Lua).
- The so-called First-Order Analysis puts a given character into isolated, initial, middle, or final state.
- The Second-order Analysis looks at the characters and relates this state to what characters precede or succeed it.

- Based on that state we do character substitutions. There can be multiple analysis and replacements in sequence.
- We can do some simple aesthetic stretching and additional related replacements.
- We need to attach identity marks and vowels in proper but nice looking places.
- In most cases we're then done. Contrary to other fonts we don't use many ligatures but compose characters.

But we go further

- The previous steps already give reasonable results and implementing it also nicely went along with the development of Lua \TeX and Con \TeX t MkIV.
- Currently we're working on extending and perfecting the font to support what we call Third-Order Contextual Analysis.
- This boils down to an interplay between the paragraph builder and additional font features.
- In order to get pleasing spacing we apply further substitutions, this time with wider or narrower shapes.
- When this is done we need to reattach identity marks and vowels.
- Optionally we can apply HZ-like stretching as a finishing touch.

Look at luatex

(kheetawul)

- no order (kh ī t ā w [u] l)
- first order
- second order
- second order (Jeem-stacking)
- minimal stretching
- maximal stretching (level 3)
- chopped letter khaa (for e.g. underlining)

لُواتِيخ
لُواتِيخ
لُوائِيخ
لُوائِيخ
لُوائِيخ
لُوائِيخ
لُوائِيخ

Hans Hagen
Pragma ADE, Hasselt