A New Kind of TEX
The evolution of pdfTeX
How we proceed

I. I start with existing engines (pdftex, partial aleph)

II. II move from 8 bit character handling to utf-8

III. III remain (mostly) downward compatible and provide a migration path

IV. IV get rid of some interfering optimizations and clean up code base

V. V provide complete control at each stage using Lua (callbacks)

VI. VI manipulate input at multiple places (e.g. ascii and utf regexp)

VII. VII provide (node) list manipulation (e.g. dedicated regexp)

VIII. VIII in addition to specials, provide attribute states (push/pop)
Reasons for using Lua

I. It's lightweight compared to other scripting languages.
II. We want to permit multiple instances during a run.
III. The interpreter is efficient and fast enough for our purpose.
IV. It has proven to be stable and is widely accepted.
V. It has reached a mature state (version 5).
VI. The language is conceptually clean and concise.
VII. It has some pleasant natural limitations.
VIII. You don't need huge bulky manuals.
Talking and Tweaking
How can we benefit from Lua

I. replace parts of macro packages
II. optimize computational extensive tasks
III. replace external methods by internal ones

But we can also be more drastic:
IV. read from zip files (packaging)
V. use sockets to talk with other processes
VI. replace and/or extending kpse (separation, integration)
VII. extend the typesetting engine with external methods (using serialization)