PARAGRAPHS

a bit of an upgrade

context 2021 meeting

Note

- Some of the following already is present for a while and has been discussed at previous meetings.
- But . . . occasionally some minor tweak gets added so consider this to be an update.

Spacing

- Spaces in TEX become glue nodes (with optional stretch and shrink).
- In traditional TEX these glue nodes are ref counted copies of the current spacing related variables.
- In LuaTEX we make real copies so that when we mess with the node list changes to glue don't affect other instances.

Parameters

- In traditional TEX the paragraphs bound properties that are in effect when `\par` happens are used when breaking into lines.
- In LuaMetaTEX the paragraphs bound properties are stored with the paragraph and can be frozen when they are set.
- This gives a more predictable (and robust) way of manipulating a paragraph.
- We can for instance get rid of grouping side effects that interfere with `\everypar`.
- Currently three dozen parameters are tracked but they are grouped in categories.

(show code and examples)

Wrapping

- Doing something in front of a paragraph is taken care of by good old `\everypar`.
- In LuaMetaTEX we also have `\everybeforepar` but so far in ConTEXt we haven't used that.
- Adding something to the end of a paragraph can be tricky so we have a wrapper mechanism: `\wrapuppar`.
- The `\wrapuppar` primitive is similar to `\atendofgroup` in the sense that it accumulates tokens (so no `\endofpar`).
- Normally these primitives are not used directly but managed by a more general system of handling paragraphs.

(show code and examples)

Normalizing

- In order to see consistent paragraphs at the Lua end in LuaMetaTEX we can normalize the lines that come from the paragraph builder.
- Normalization results in:
  - the first line having: indent skip
  - each line having: left hang, left skip, right skip, right hang
  - the last line having: left parfill skip, right parfill skip
- It is controlled by `\normalizelinemode` which has additional flags for swapping hanging indentation and par shapes, breaking after dir nodes, removing margin kerns and clipping the line width.
- The clipping options avoids the side effects of TEX using shifts which has the side effect of unreal dimensions. This is one of the tricks/properties of the traditional engine that is perfectly fine until we open up things.

(show code and examples)