LuaTEX-ja 和文処理グルーについて

2011/5/15

本文書では，LuaTEX-ja が（現時点において）和文処理に関わる glue/kern をどのように挿入するかの内部処理について説明する．

予備知識

説明に入る前に，段落や hbox の中身は，TEX の内部では node 達によるリストとして表現されていることに注意する．node の種類については，*The LuaTEX Reference*の第 8 章を参照して欲しい．代表的なものを挙げると，

- *glyph_node*: 文字（合字も含む）を表現する．和文処理グルーを挿入する際には，既に各 *glyph_node* が欧文文字のものか和文文字のものか区別がついている．
- *glue_node*: glue を表す．
- *kern_node*: kern を表す．各 kern_node には subtype という値があり，次の 3 種類を区別できるようになっている．
  - 0: 欧文用 TFM 由来
  - 1: 明示的な\kern か，イタリック補正 (\/) によるもの
  - 2: 非数式アクセント用文字の左右位置調整のためのもの
- *penalty_node*: penalty を表す．
- *hlist_node*: hbox（水平ボックス）を表す．

以後，次のように，node がどのように連続しているかを表すことにする．

$$\boxed{a} \longrightarrow \boxed{b}_{\mathrm{I}} \longrightarrow \boxed{c}$$

右下についている添字は，LuaTEX-ja においてその node の役割を区別するためにつけられた値であり，次のようになっている．

```
I: イタリック補正由来の\kern
T: \[x]kanjiskipに置換されうる\kern
J: JFM由来のglue/kern
K: 禁則処理用penalty
E:
KS: \kanjiskip
XS: \xkanjiskip
```

JFM 由来グルーの挿入 (luatexja-jfmglue.lua)

JFM 由来グルーの処理は，「連続する 2 つの node の間に何を入れるか」という単位で行われる．そのため，

- node 生成を伴わないもの（グループ境界，\relax 等）は全て無視される．
- 一方，node 生成を伴うものは全て「透過しない」．例えば，次のソースにおいて，閉じ括弧と開き括弧の間に入る物は，左と右とで異なる：

  ）（　）\hbox{}（

- JFM 由来グルーの挿入禁止を行う\inhibitglue は，内部では専用の node を作ることによって実装している．この\inhibitglue 用 node は透過する．

以下，q, p を連続する node とする．

■ **2 つの和文文字の間**

この場合，グルー挿入に関係する量は次の通りである．これら 3 つの量の値によって，q と p の間に何が挿入されるかが決定される．これらの記号は他の場合にも用いる．

- g: JFM で指定された，q と p の間に入る glue/kern．JFM で規定されていないときは $\emptyset$ と書こう．両ノードで使われている JFM が異なる時の g の決定方法は，後に記述する．
- w: JFM で指定された，「q の直後で改行が行われた場合，q と行末の間に入るカーン量」の値．$g - w$ で，g の自然長を w だけ減算した glue/kern を表すことにする．
- P: q に対する行末禁則用ペナルティ (post-break penalty) と，p に対する行頭禁則用ペナルティ (pre-break penalty) との和．どちらも設定されていないときは 0 となる．

設計方針としては，

- JFM 由来で入るものが kern の場合，この場所では行分割は許さない．
- そうでない場合，(penalty の値 P があるが) この場所での行分割は可能である．

である．さて，次が実際の場合わけである：

1. $w \neq 0$, $g = \emptyset$ のとき

$$\boxed{q} \longrightarrow \boxed{\text{kern } w}_{\text{E}} \longrightarrow \boxed{\text{penalty } P}_{\text{K}} \longrightarrow \boxed{\text{kern } -w}_{\text{T}} \longrightarrow \boxed{p}$$

2. $w \neq 0$, $g \neq \emptyset$ のとき：

$$\boxed{q} \longrightarrow \boxed{\text{kern } w}_{\text{E}} \longrightarrow \boxed{\text{penalty } P}_{\text{K}} \longrightarrow \boxed{g - w}_{\text{J}} \longrightarrow \boxed{p}$$

3. $w = 0$, g: kern のとき

$$\boxed{q} \longrightarrow \boxed{g}_{\text{J}} \longrightarrow \boxed{p}$$

4. $w = 0$, g: glue のとき

$$\boxed{q} \longrightarrow \boxed{\text{penalty } P}_{\text{K}} \longrightarrow \boxed{g}_{\text{J}} \longrightarrow \boxed{p}$$

5. $w = 0$, $g = \emptyset$, $P \neq 0$ のとき

$$\boxed{q} \longrightarrow \boxed{\text{penalty } P}_{\text{K}} \longrightarrow \boxed{p}$$

6. $w = 0$, $g = \emptyset$, $P = 0$ のとき

$$\boxed{q} \longrightarrow \boxed{p}$$

なお，両ノードで使われている JFM が異なる時の g の決定方法であるが，

1. g_{L} を，q に使用されている JFM における，「q と文字'`diffmet`'」の間に入る glue/kern の値とする．
2. g_{R} を，p に使用されている JFM における，「文字'`diffmet`' と p」の間に入る glue/kern の値とする．
3. 両方から，実際に入る g の値を計算する．
   − g_L, g_R の少なくとも片方が $\emptyset$ のときは，$\emptyset$ でない方をそのまま採用する．
   − 両方とも $\emptyset$ でない場合は，`differentjfm` の値にそって g の値を計算する．

■和文文字と（和文文字，**kern** 以外の **node**）の間

「和文文字の間」の場合に対して，以下が異なる：

- g は，q に使用されている JFM における，「q と文字'jcharbdd'」の間に入る glue/kern の値である．
- p が **penalty** でない場合は，いつもこの位置で行分割できるようにするため，case 6 (w, $P = 0$, $g = \emptyset$) の場合にも，q と p の間には 0 という値の penalty が入る．即ち，次のようになる．

$$\boxed{q} \longrightarrow \boxed{\text{penalty } 0}_{\text{K}} \longrightarrow \boxed{p}$$

■（和文文字，**kern** 以外の **node**）と和文文字の間

この場合も，基本的には「和文文字の間」と似ているが，以下が異なる：

- g は，p の JFM における，「文字'jcharbdd' と p」の間に入る glue/kern の値である．
- 常に $w = 0$ である．
- いつもこの位置で行分割できるようにするため，case 6 (w, $P = 0$, $g = \emptyset$) の場合にも，q と p の間には 0 という値の penalty が入る．

即ち，次の 3 通りになる．

1. g: kern のとき

$$\boxed{q} \longrightarrow \boxed{g}_{\text{J}} \longrightarrow \boxed{p}$$

2. g: glue のとき

$$\boxed{q} \longrightarrow \boxed{\text{penalty } P}_{\text{K}} \longrightarrow \boxed{g}_{\text{J}} \longrightarrow \boxed{p}$$

3. $g = \emptyset$ のとき

$$\boxed{q} \longrightarrow \boxed{\text{penalty } P}_{\text{K}} \longrightarrow \boxed{p}$$

■和文文字と **kern** の間，**kern** と和文文字の間

和文文字の後に kern が続いた場合，あるいは kern の後に和文文字が続いた場合，この間で行分割はできないものとしている．そのため，以下の 3 ケースに限られる：

1. g: kern のとき

$$\boxed{q} \longrightarrow \boxed{g}_{\text{J}} \longrightarrow \boxed{p}$$

2. g: glue のとき

$$\boxed{q} \longrightarrow \boxed{\text{penalty } 10000}_{\text{K}} \longrightarrow \boxed{g}_{\text{J}} \longrightarrow \boxed{p}$$

3. $g = \emptyset$ のとき

$$\boxed{q} \longrightarrow \boxed{p}$$

なお，ここでの g は，

- kern が前だった場合は，q の JFM における，「q と'jcharbdd'」の間に入る glue/kern の値．
- kern が後だった場合は，p の JFM における，「'jcharbdd' と p」の間に入る glue/kern の値．

■要検討の箇所

私が推測するに，欧文では，

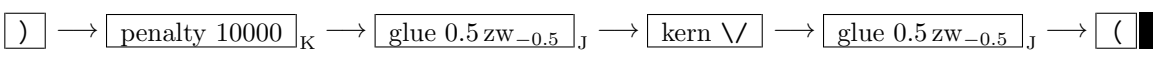
- 単語内ではフォントは変わらない．
- 単語内では，明示的に/ハイフネーションにより挿入された discretionary break 以外では行分割がおきない．

という事情があるため，TFM 由来の kern や合字処理は（node を生成しないもの以外は）何も透過しないという状態になっているものと思われます．

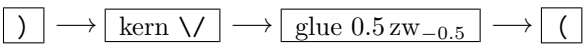
そのため，JFM グルー等の仕様を考える場合，欧文でいう「単語」に対応するようなものは何か，というのを考える必要があります．現実装では，素直に欧文の合字処理と同様のものであると考え，透過する node はない，という仕様にしています．しかし，`\[x]kanjiskip` の処理と共通にしてしまうというのも考え方によってはありかもしれません．

- イタリック補正の **kern** の周囲

  例えば，`jfm-ujis.lua` では，`'jcharbdd'` は文字クラス 0 であるため，今の実装では，「）\/（」という入力からは，次の node の並びを得る：



  一方，イタリック補正を JFM 由来グルーが透過するとしたならば，当然



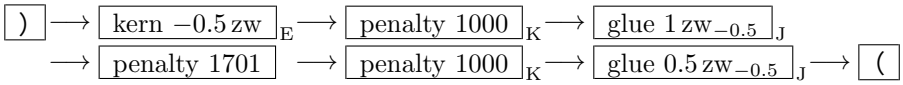
  となる（実際の組版イメージでは，「*斜め体*（」「*斜め体*（」）．どちらにするか？

- **penalty** の周囲

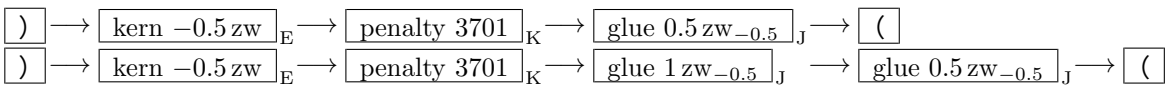
  これも，例えば次の設定の下では，「）`\penalty1701` （」からは以下を得る：

  – 「)」と行末の間に $-0.5\,\mathrm{zw}$ だけ kern を入れる．

  – 「)」「(」の行頭/行末禁則用 penalty の値はどれも 1000．



  例えば penalty を合算することとした場合，上の入力例では本来「)」「(」の間に 1701 の penalty があるのだから，



  のどちらか（上は penalty を透過する場合，下は透過しない場合）にするのが良いと思われます．

- **discretionary break** の取り扱い

  discretionary break (*disc_node*) は，行分割時の行末の内容 ⟨*pre*⟩，行頭の内容 ⟨*post*⟩，それに行分割しないときの内容 ⟨*no_break*⟩ の 3 つをリストの形で持っている．`linebreak.w` を見る限り，LuaTEX でも ⟨*pre*⟩, ⟨*post*⟩, ⟨*no_break*⟩ の中身には glue や penalty を許容していないようだ．

  現行の実装では，⟨*pre*⟩, ⟨*post*⟩, ⟨*no_break*⟩ のどれも，和文フォントへの置換の段階からして行われていない（だから中身は全部欧文扱いとなる）．単純にサボっていました^^; ⟨*pre*⟩, ⟨*post*⟩, ⟨*no_break*⟩ の中身に glue や penalty が許容されないことから，これらに対する和文用処理の方法として，次の 2 種類が挙げられる．私は前者で良いのではないかと思っているのだが……．

  – （現行のまま）discretionary break の中身に和文文字はないものと想定する．例えば ⟨*pre*⟩ の中身に和文文字を入れたい場合は，⟨*pre*⟩ の中身全体を必ず hbox で括ることとする．

– 「glue を挿入」を全部「自然長だけを取り出した kern を挿入」に置き換え，普段の和文処理グルー挿入処理を流用する．

\[x]kanjiskip の挿入

現実装の\[x]kanjiskip の挿入の方針として，

- JFM グルーが挿入されていないところに「標準の空き量」として挿入する．
- 実際の段落/hbox の内容に即して，組版イメージの見た目に関係のないところは透過する．

\[x]kanjiskip 挿入処理では，次の 3 つの node を用いている．

$$\boxed{nr} \longrightarrow \cdots \longrightarrow \boxed{nq} \longrightarrow \cdots \longrightarrow \boxed{np}$$

- nr と np の間に\[x]kanjiskip を挿入しようとする．
- 実際に node の形で挿入しようとする場所は nq の直後である．
- nr, nq は異なる node とは限らない．
- np はリストの先頭から末尾までループで渡る．その過程で nr, nq を適宜更新し，実際の node 挿入処理を行っている．

ループの中で，以下の場合には nr は変化せず，$nq \leftarrow np$ となる．つまり，これらの node に対して\[x]kanjiskip は透過する：

- np が penalty の場合
- np が kern であって，予備知識の項目で述べられた値が

  I (イタリック補正)，E（行末との間），T（一時的）

  であるもの．後者 2 つは JFM グルーの挿入で入るものなので，ユーザは「イタリック補正は透過」と考えればよい．
- np がアクセント由来のもの．この場合は，nq は変化せず，$np \leftarrow next(next(np))$ となる．
- np が insertion, mark, \vadjust, whatsit の node である場合．これらは水平リストからは消え去る運命にある．