TEX言語の展開制御による文書の構造化

expandafter

諏訪 敬之
（東京大学工学部計数工学科数理情報工学コース 3 年）

2014 年 11 月 08 日 TEX ユーザの集い 2014

自己紹介

諏訪敬之（すわ・たかし）

東京大学工学部計数工学科
数理情報工学コース３年

どこにでもいる普通の学部生です.
pLaTeX 歴 5 年（未熟）
TeX 歴半年未満

Twitter: @bd_gfngfn
GitHub: gfngfn

内容は個人的見解です.
また，かなり大雑把な説明を含むこと，既にご存知の方にとっては釈迦に説法であることをご了承ください.

展開制御とは (1)

● TEX の展開制御とは

……TEX によるプリミティヴやマクロの展開が特定の順序に従って行なわれるように指定すること.

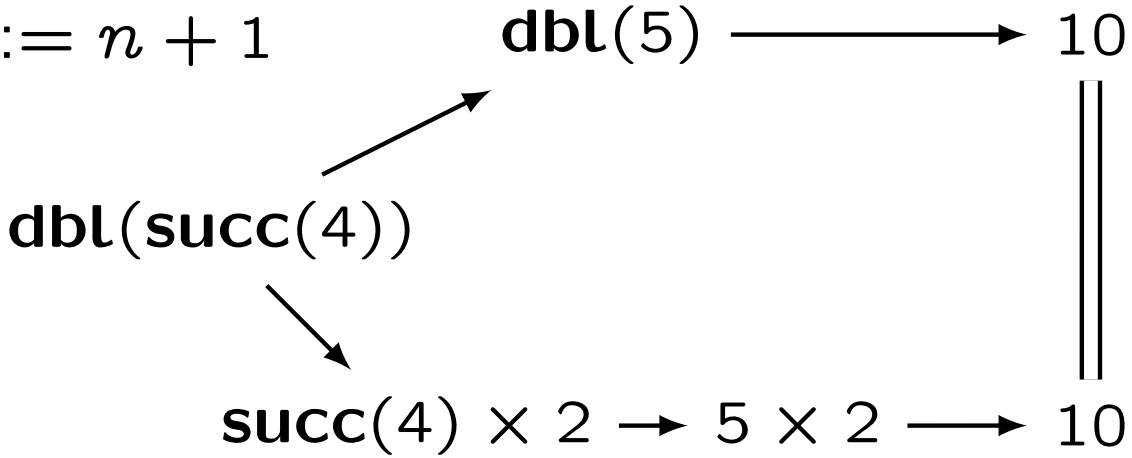
●なぜ展開制御が必要なのか

……TEX の言語仕様が展開に関して合流性を満たさないから.

簡約の合流性のイメージ

$$\mathbf{dbl}(n) := n \times 2$$

$$\mathbf{succ}(n) := n + 1$$



展開制御の実践 (1)

```
\def\showdate#1/#2/#3;{#1年#2月#3日}
\def\MMXIV{2014}
\def\texconfday{11/08}
```

と定義した下で,

\showdate	\MMXIV	/	\texconfday	;	A	/	B	;

を展開すると

2	0	1	4	年	1	1	月	0	8	日	A	/	B	;

……

展開制御の実践 (2)

```
\def\showdate#1/#2/#3;{#1年#2月#3日}
\def\MMXIV{2014}
\def\texconfday{11/08}
```

と定義した下で，

\showdate	\MMXIV	/	\texconfday	;	A	/	B	;

を展開すると

2	0	1	4	年	1	1	月	0	8	日	A	/	B	;

……かと思いきや

2	0	1	4	年	1	1	/	0	8	;	A	月	B	日

になってしまう！

`\showdate` よりも先に `\texconfday` を展開する必要がある

展開制御の実践 (3)

悪名高き `\expandafter` の登場！以降は `X` と略記

- “青赤モデル”の提案

  可能な限り展開して進む青線 | と 1 回だけ展開する赤線 |

- 青線の展開規則

$$|t| \longrightarrow \begin{cases} |t| & (t \text{ が展開可能}) \\ |t| & (t \text{ が展開不可能}) \end{cases}$$

  ただし赤線が存在する間は動かない

- `\expandafter` の展開規則

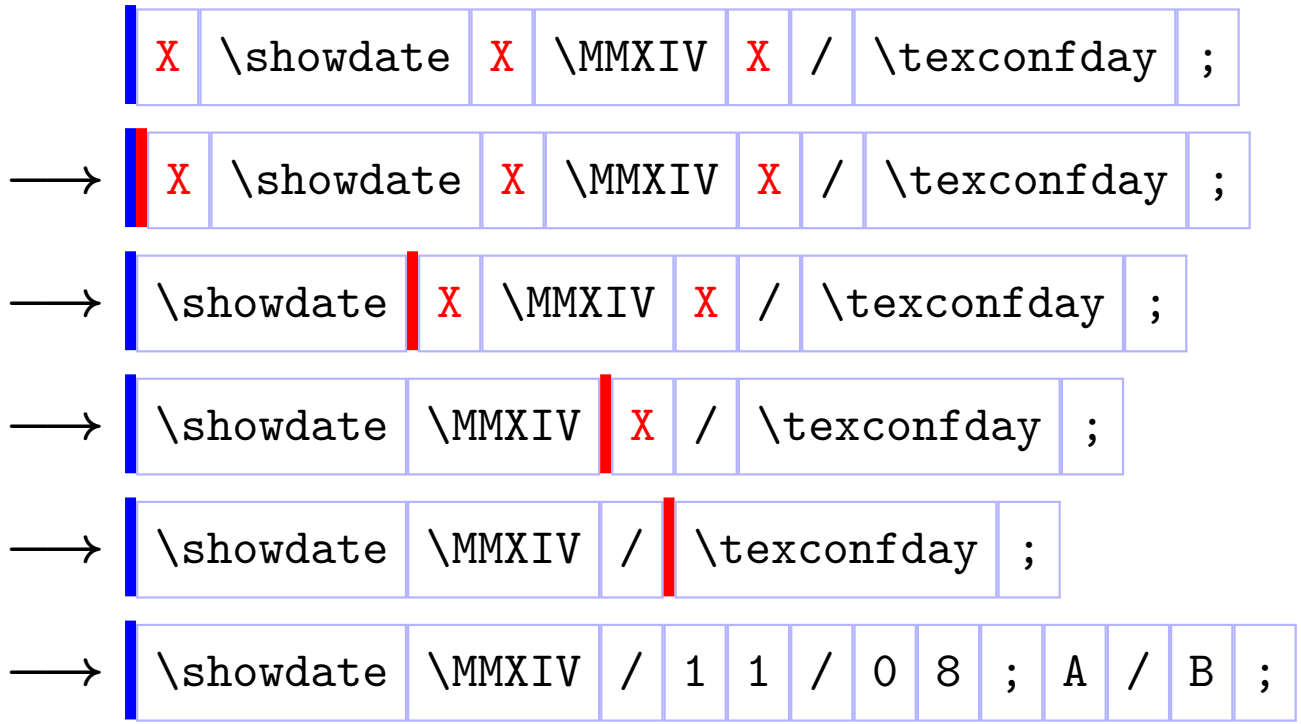
$$|\texttt{X}\ t_1\ t_2 \longrightarrow t_1\ |\ t_2$$

展開制御の実践 (4)

●赤線の展開規則（\expandafter 以外）

$t \longrightarrow \begin{cases} s_1 \cdots s_n & (t \text{ と引数} \longrightarrow s_1 \cdots s_n) \\ t & (t \text{ が展開不可能}) \end{cases}$

先ほどの例も意図通り展開できる（めでたしめでたし）



展開制御の実践 (5)

どうして X を挿入すべき位置がわかるのか

展開制御の実践 (6)

どうして X を挿入すべき位置がわかるのか

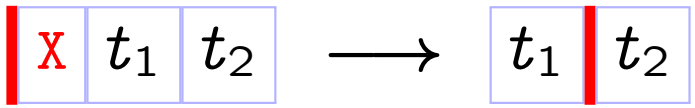
実は展開したい位置から逆算できる！

展開制御の実践 (7)

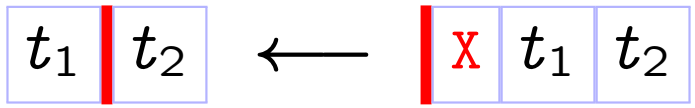
どうして X を挿入すべき位置がわかるのか

実は展開したい位置から逆算できる！

\expandafter の展開規則



を左右反転すると

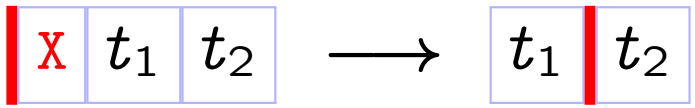


展開制御の実践 (8)

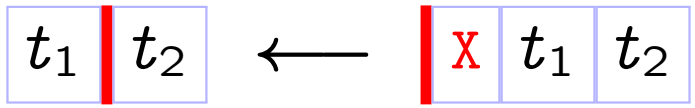
どうして X を挿入すべき位置がわかるのか

実は展開したい位置から逆算できる！

\expandafter の展開規則



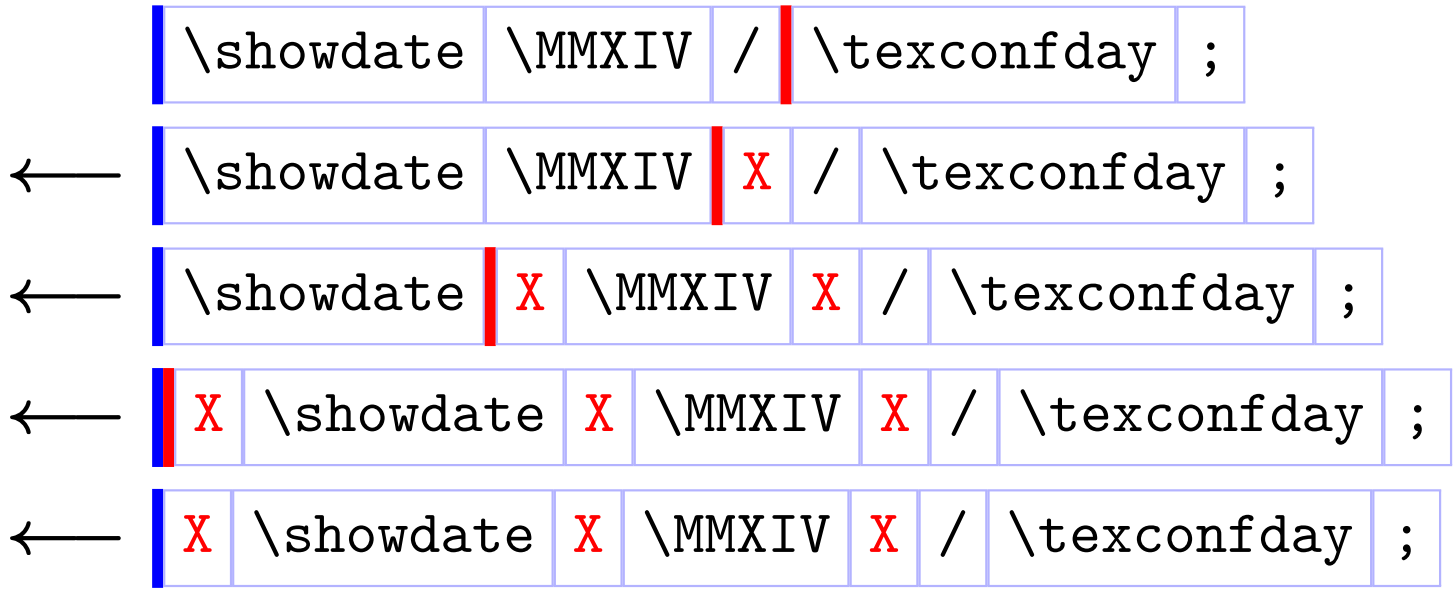
を左右反転すると



……つまり t_2 を展開したいときにこれを適用して

X を挿入していく

展開制御の実践 (9)



無事逆算できた（当然といえば当然）

実際には複数の場所を先に展開したり，同じ位置を 2 回展開したりするが，基本的にはこの逆算がどんな場合でも使える！

展開制御の実践 (10)

逆算さえできれば，展開制御も意外にこわくない！

展開制御による構造化 (1)

●結局，展開制御って何に使えるの?

……`\def`，`\let`，`\csname`〜`\endcsname`，`\ifx` などの条件分岐，
カウント変数と合わせて使うことで柔軟なマクロがつくれる!
（数多くのパッケージや LaTeX 自体もそうして創られている）

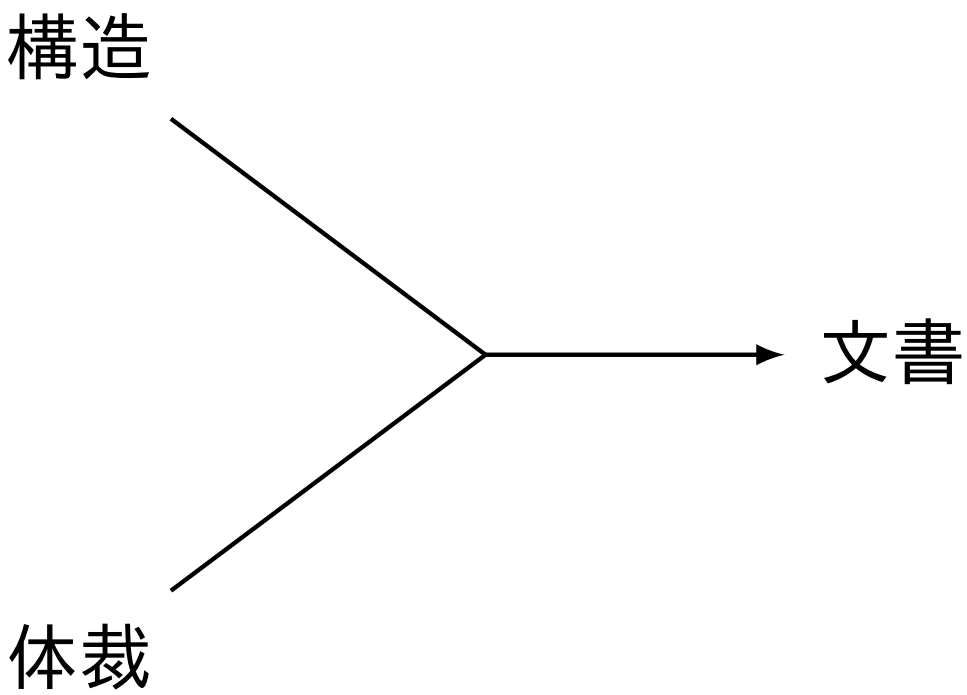
特に個人的に強調したいのは

高度なマークアップに使える

ということ.

展開制御による構造化 (2)

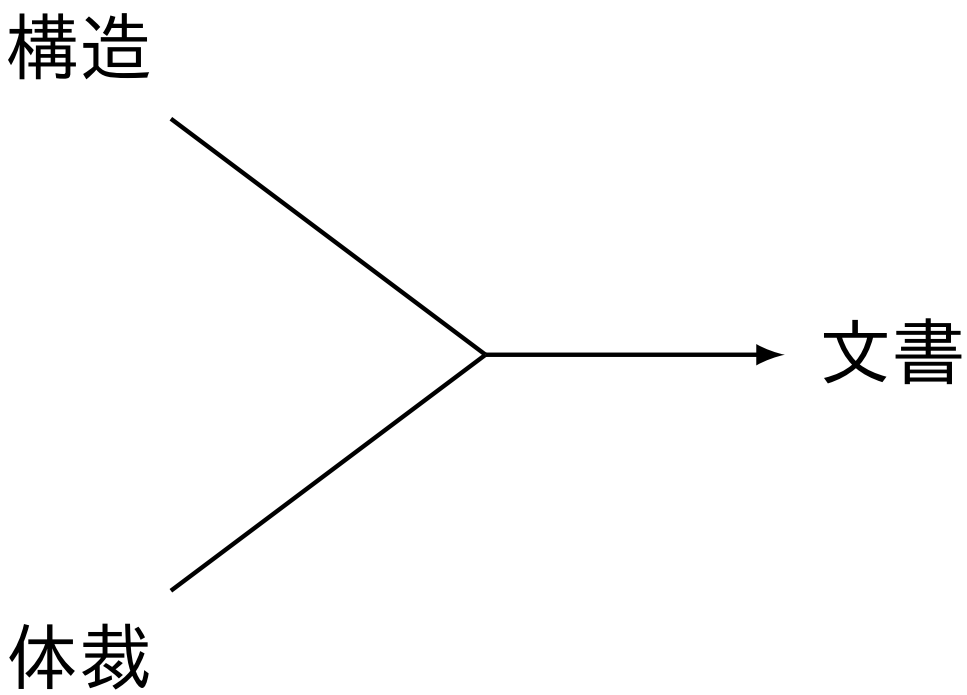
よく言われる文書の構造・体裁分離
HTML・CSS など



展開制御による構造化 (3)

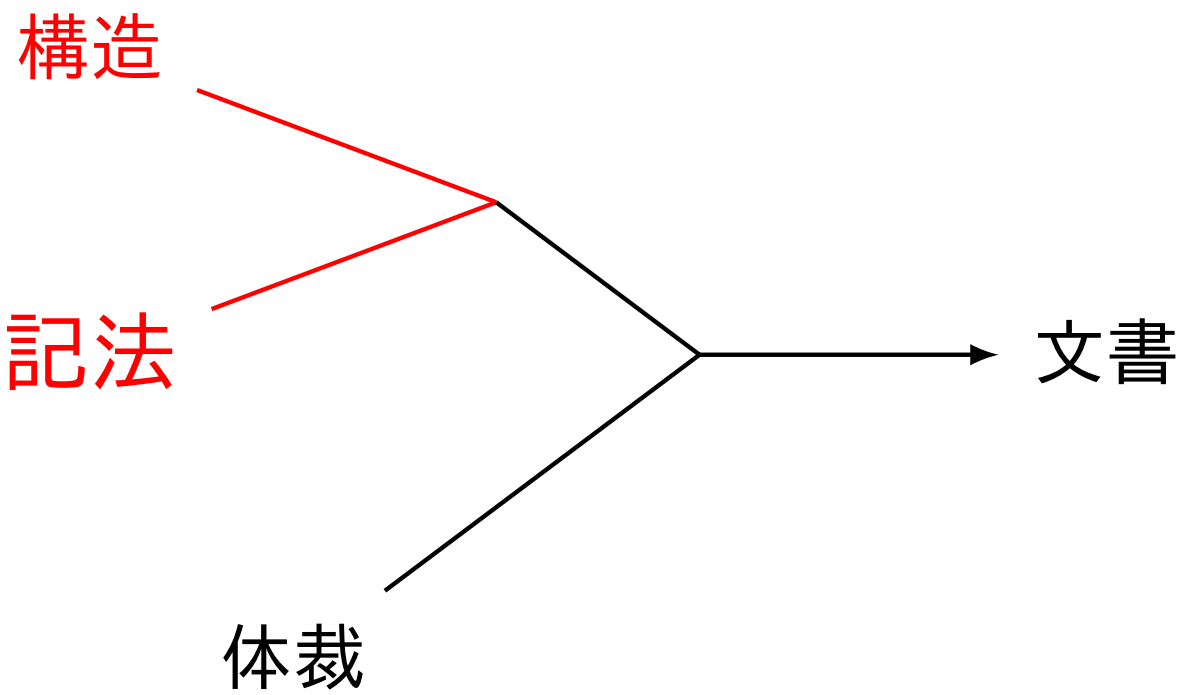
よく言われる文書の構造・体裁分離
HTML・CSS など

しかし本当にこれだけでよいのか?



展開制御による構造化 (4)

構造・記法・体裁分離による
さらに高度なマークアップへ



展開制御による構造化 (5)

- “記法”？

例えば数理論理学：同一の論理式の異なる表記
個々人の流儀により違いがある

$$\forall x, y, z \in L \ (x \preceq y \land y \preceq z \supset x \preceq z)$$

$$\forall x \forall y \forall z \in L. \left(\left(x \preccurlyeq y \land y \preccurlyeq z \right) \rightarrow x \preccurlyeq z \right)$$

記法に依存しない構造だけを取り出して，
記法はオプション的に指定したい!

展開制御による構造化 (6)

●宣伝 (?)

そんな構想で自分なりに実装している試験的パッケージが
gfncmd, gfnlf

http://github.com/gfngfn/gfncmd

gfncmd 検索

(仕様書の整備は発表に間に合いませんでした（すみません))

実はさっきの論理式はどちらも同じトークン列で構造が書かれています

展開制御による構造化 (7)

$$\forall x, y, z \in L \ (x \preceq y \land y \preceq z \supset x \preceq z)$$

```
\useparensingleqtfr %量化子 ∀, ∃ は先頭だけ，括弧必須
\usepreceqaspord %半順序を ≼ に
\useinvertedCaslimpl %論理包含を ⊃ に
\usenormalparen %括弧を普通の\left〜\right に
  ⋮
\lfqtfr{
  \forallin{{x}{y}{z}}{L}
}{
  \lflimpl{
    \lfland{\lffml{x \pord y}}{\lffml{y \pord z}}
  }{
    \lffml{x \pord z}
  }
}
```

展開制御による構造化 (8)

$$\forall x \forall y \forall z \in L.\ \Big(\big(x \preccurlyeq y \land y \preccurlyeq z \big) \to x \preccurlyeq z \Big)$$

```
\usedotpluralqtfr %量化子 ∀，∃ は全てつけ，末尾をドットに
\usepreccurlyeqaspord %半順序を ≼ に
\useexpandingparen %括弧が外側ほど大きくなるようにする
\makeleftparenmandatory{lflimpl}{lfland}
  ⋮          %論理包含の左辺にくる連言の括弧を省略しない
\lfqtfr{
  \forallin{{x}{y}{z}}{L}
}{
  \lflimpl{
    \lfland{\lffml{x \pord y}}{\lffml{y \pord z}}
  }{
    \lffml{x \pord z}
  }
}
```

展開制御による構造化 (9)

●利点

- 保守性が高い．気分次第で簡単に記法を変更できる
- 複数人で文書をつくるときも記法統一が容易
- 他人がつくった文書も自分の好きな記法で読めるかも

●欠点

- 使い手にリテラシーを要求する
- 記法により配置が大きく変わるときに調整困難
  （人工知能並みの組版最適化がコンパイラに求められる）

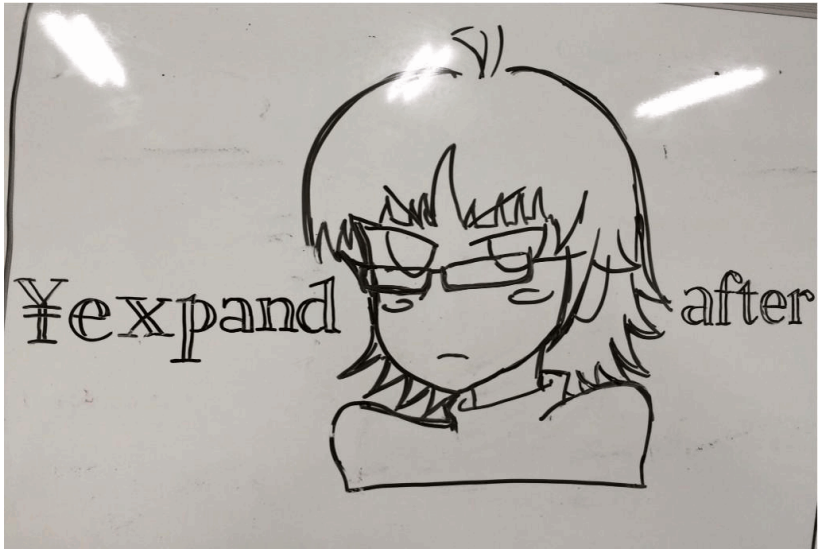
展開制御による構造化 (10)

記法に依存する処理は
どんどんマクロ化して

エレガントに構造化された
文書を打とう！

最適化さえ充実させれば，構造化は大正義

ありがとうございました



●主要参考文献（敬称略）

- 藤田眞作『LaTeX2ε マクロ作法』
- 八登崇之，マクロツイーター
  http://d.hatena.ne.jp/zrbabbler