

L'extension `hhline`*

David Carlisle

28/10/2014

Ce fichier est maintenu par l'équipe du «`LATEX` Project». Les rapports d'anomalie peuvent être envoyés en anglais à <http://latex-project.org/bugs.html> (catégorie `tools`).

Résumé

Cette extension augmente les possibilités de filets horizontaux et verticaux dans un environnement `array` ou `tabular`. Il introduit la nouvelle commande `\hhline` qui produit un filet comme `\hline` ou un double filet comme `\hline\hline`, sauf pour leur interaction avec les filets verticaux.

1 Introduction

L'argument de `\hhline` est similaire au préambule d'un environnement `array` ou `tabular`. Il consiste en une liste d'unités lexicales (*tokens*) ayant les significations suivantes :

- = Un filet horizontal double sur la largeur d'une colonne.
- Un filet horizontal simple sur la largeur d'une colonne.
- ~ Une colonne sans filet horizontal.
- | Un filet vertical qui croise un filet horizontal simple ou double.
- : Un filet vertical qui est brisé par un filet horizontal double.
- # Un filet horizontal double segment entre deux filets verticaux.
- t La moitié supérieure d'un filet horizontal double.
- b La moitié inférieure d'un filet horizontal double.
- * `*{3}{==#}` se développe en `==#==#==#`, comme avec `*` dans le préambule.

Si un double filet vertical est spécifié (`|` ou `:`) alors les filets horizontaux produits par `\hhline` sont brisés. Pour obtenir l'effet d'un filet horizontal « passant au

*Ce fichier a pour numéro de version v2.03 et a été mis à jour le 28/10/2014. La première traduction, basée sur la version v2.03, a été publiée par Jean-Pierre Drucbert en 2000.

travers » d'un filet vertical double, utilisez un `#` ou omettez les spécificateurs de filets verticaux, selon que vous vouliez que le double filet vertical soit brisé ou pas.

Les unités lexicales `t` et `b` doivent être utilisées entre deux filets verticaux. La combinaison `|tb|` produit les mêmes lignes que `#` mais est moins efficace. Ces unités lexicales sont plutôt pensées pour faire des combinaisons telles que `|t:` (coin supérieur gauche) et `:b|` (coin inférieur droit).

Si `\hline` est utilisée pour faire un simple filet horizontal, alors son argument ne doit contenir que des unités lexicales `-`, `~` et `|` (et des expressions débutant par `*`).

Voici un exemple utilisant la plupart de ces possibilités :

```
\begin{tabular}{||cc||c|c||}
\hline{|t:==:t:==:t|}
a&b&c&d\\
\hline{|:==:|~|~||}
1&2&3&4\\
\hline{|#==#~|=#}
i&j&k&l\\
\hline{||--|--||}
w&x&y&z\\
\hline{|b:==:b:==:b|}
\end{tabular}
```

a	b	c	d
1	2	3	4
i	j	k	l
w	x	y	z

Les filets produits par la commande `\hline` de L^AT_EX sont formées par une seule primitive de T_EX, `\hrule`. Les filets produits par `\hline` sont constitués de nombreux petits segments. T_EX les place très précisément dans le fichier `.dvi`, mais le programme que vous utilisez pour imprimer le fichier `.dvi` peut ne pas les aligner exactement (un problème similaire peut survenir avec les lignes obliques dans l'environnement `picture`).

Si ceci pose un problème, vous pouvez soit essayer un autre programme d'impression (si cela est possible), soit augmenter la dimension `\arrayrulewidth` pour essayer de réduire les effets.

2 Implémentation

1 `<package>`

`\HH@box` Construit une boîte contenant un segment de double filet horizontal. Dans le cas le plus courant, deux filets de longueur `\doublerulesep` seront stockés dans la boîte `\box1`. Ceci n'est pas initialisé tant que `\hline` n'est pas appelé car l'utilisateur peut changer les paramètres `\doublerulesep` et `\arrayrulewidth`. Les deux arguments de `\HH@box` sont les largeurs (autrement dit les longueurs) du filet du haut et du filet du bas.

```
2 \def\HH@box#1#2{\vbox{%
3 \hrule \@height \arrayrulewidth \@width #1
```

```

4 \vskip \doublerulesep
5 \hrule \@height \arrayrulewidth \@width #2}}

\HH@add Développe le préambule dans le registre \toks@.
6 \def\HH@add#1{\toks@\expandafter\the\toks@#1}}

\HH@xexpast Nous « empruntons » la version de \xexpast de l'extension de Frank Mittelbach,
\HH@xexnoop car ceci permet de faire apparaître # dans la liste des arguments.
7 \def\HH@xexpast#1*#2#3#4\@@{%
8   \@tempcnta #2
9   \toks@={#1}\@temptokena={#3}%
10  \let\the@toksz\relax \let\the@toks\relax
11  \def\@tempa{\the@toksz}%
12  \ifnum\@tempcnta >0 \@whilenum\@tempcnta >0\do
13    {\edef\@tempa{\@tempa\the@toks}\advance \@tempcnta \m@ne}%
14    \let \@tempb \HH@xexpast \else
15    \let \@tempb \HH@xexnoop \fi
16  \def\the@toksz{\the\toks@}\def\the@toks{\the\@temptokena}%
17  \edef\@tempa{\@tempa}%
18  \expandafter \@tempb \@tempa #4\@@}
19
20 \def\HH@xexnoop#1\@@{}

\hhline La commande utilise une version simplifiée de \mkpream pour décomposer l'argu-
ment à destination de \hhline. En fait, elle est fortement simplifiée. Elle suppose
que les filets verticaux sont en fin de colonne. Si vous aviez à spécifier c|@{xx}|
dans l'argument du tableau, alors \hhline ne serait pas capable d'accéder au pre-
mier filet vertical (il devrait y avoir une option @{} et l'ajout de \leaders jusqu'à
atteindre la largeur de la boîte contenant l'expression @{}). Nous utilisons une
boucle avec \futurelet plutôt que \tfor afin que nous puissions utiliser # pour
indiquer le croisement d'un filet vertical double avec un filet horizontal double.
\if@firstamp est vraie dans la première colonne et faux sinon.
\if@tempswa est vraie si la précédente entrée était un filet vertical (:, | or #).
21 \def\hhline#1{\omit\@firstamptrue\@tempswafalse
Place deux filets de largeur \doublerulesep dans \box1.
22 \global\setbox\@ne\HH@box\doublerulesep\doublerulesep
Si l'extension array de Frank Mittelbach est chargée, nous n'avons pas besoin d'une
commande \hskip négative autour des filets verticaux.
23 \xdef\@tempc{\ifx\extrarowheight\HH@undef\hskip-.5\arrayrulewidth\fi}%
Maintenant les éléments avec * sont développés et des unités lexicales fictives (
\relax et ' ) sont ajoutés aux deux extrémités de la liste des unités lexicales. La
commande \HH@let est appelée pour commencer le traitement de cette liste.
24 \HH@xexpast\relax#1*0x\@@\toks@{}\expandafter\HH@let\@tempa'}

\HH@let La dernière unité lexicale est écartée et la suivante est analysée.
25 \def\HH@let#1{\futurelet\@tempb\HH@loop}

```

`\HH@loop` La boucle principale. Notez que nous utilisons `\ifx` plutôt que `\if` en version 2 car la nouvelle unité lexicale `~` est active.

```

26 \def\HH@loop{%
Si l'unité lexicale suivante est ‘, la boucle est arrêtée et les filets sont placés dans la rangée.
27 \ifx\@tempb'\def\next##1{\the\toks@\cr}\else\let\next\HH@let
| ajoute un filet vertical (en travers d'un simple ou d'un double filet horizontal).
28 \ifx\@tempb|\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
29 \HH@add{\@tempc\vline\@tempc}\else
: ajoute un filet vertical brisé (à travers un double filet horizontal).
30 \ifx\@tempb:\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
31 \HH@add{\@tempc\HH@box\arrayrulewidth\arrayrulewidth\@tempc}\else
# ajoute un double filet horizontal entre deux filets verticaux.
32 \ifx\@tempb##\if@tempswa\HH@add{\hskip\doublerulesep}\fi\@tempswatrue
33 \HH@add{\@tempc\vline\@tempc\copy\@ne\@tempc\vline\@tempc}\else
~ ajoute une colonne sans filet horizontal (ce qui donne un effet similaire à \cline).
34 \ifx\@tempb~\@tempswafalse
35 \if@firstamp\@firstampfalse\else\HH@add{&\omit}\fi
36 \HH@add{\hfil}\else
- ajoute un filet horizontal simple à travers de la colonne.
37 \ifx\@tempb-\@tempswafalse
38 \if@firstamp\@firstampfalse\else\HH@add{&\omit}\fi
39 \HH@add{\leaders\hrule\@height\arrayrulewidth\hfil}\else
= ajoute un filet horizontal double à travers de la colonne.
40 \ifx\@tempb=\@tempswafalse
41 \if@firstamp\@firstampfalse\else\HH@add{&\omit}\fi
Place autant de copies de \box1 que possible avec la commande \leaders, ceci peut laisser des vides aux extrémités, aussi une boîte supplémentaire est placée à chaque extrémité, superposées aux \leaders.
42 \HH@add
43 {\rlap{\copy\@ne}\leaders\copy\@ne\hfil\llap{\copy\@ne}}\else
t ajoute la moitié haute d'un segment de filet horizontal double dans une commande \rlap afin qu'elle puisse être utilisée avec b.
44 \ifx\@tempb t\HH@add{\rlap{\HH@box\doublerulesep\z@}}\else
b ajoute la moitié basse d'un segment de filet horizontal double dans une commande \rlap afin qu'elle puisse être utilisée avec t.
45 \ifx\@tempb b\HH@add{\rlap{\HH@box\z@\doublerulesep}}\else
Sinon, l'unité lexicale est ignorée, avec un avertissement l'indiquant.
46 \PackageWarning{hhline}%
47 {\meaning\@tempb\space ignored in \noexpand\hhline argument%
48 \MessageBreak}%
49 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi

```

Puis la boucle est parcourue à nouveau.

```
50 \next}
```

```
51 \end{package}
```