

tabu

tabu and longtabu

Flexible L^AT_EX tabulars

FC

2011/01/15 – version 1.9

Abstract

This package defines a single environment `tabu` to make all kinds of tabulars in text or in math mode provided that they do not split across pages.

Starting from version 1.5, the environment `longtabu` is provided to make tabulars that can stretch out on several pages. `longtabu` works like `tabu` and is defined only if the package `longtable` is loaded.

`tabu` is more flexible than `tabular`, `tabular*`, `tabularx` and `array` and extends the possibilities. All tabulars in this document were made with the `tabu` environment. Widths computations for `tabu` X columns is *optimised* compared to `tabularx` implementation. Last (but not least) `tabu` is fully compatible with any package that provides commands to format tabulars. Indeed, `tabu` does not modify any of the macros of `array.sty`.

`tabu` requires ϵ -T_EX and the standard package `array.sty`. Natural widths of columns are computed by (but not printed by) the code of `varwidth` by D. Arseneau (for the cases of X columns in `tabu spread` and *negative width coefficients*¹). Finally `longtabu` is based on `longtable`.

Contents

Summary of the features provided by <code>tabu</code>	3
1 The <code>tabu</code> environment	4
1.1 <code>tabu</code> , <code>tabu to</code> and <code>tabu spread</code>	4
1.2 <code>longtabu</code> , <code>longtabu to</code> and <code>longtabu spread</code>	4
1.3 <code>tabu</code> X columns	5
1.3.1 X columns with “ <code>tabu spread</code> ”	
1.3.2 Negative width coefficients for X columns	
1.3.3 Multicolumn in <code>tabu</code>	
1.3.4 Nesting <code>tabus</code> with X columns	
1.4 Lines inside <code>tabu</code>	7
1.4.1 First important remarks	
1.4.2 Vertical lines: <code> </code> has an optional parameter	
1.4.3 More style for horizontal lines: <code>\tabucline</code>	
1.5 <code>tabu</code> in math mode	9
1.6 Verbatim inside <code>tabu</code> with X columns.....	10
1.7 Printing numbers inside <code>tabu</code> with <code>numprint</code> and <code>siunitx</code>	10
1.7.1 Just make it easy !	
1.7.2 You should know how it works... ..	
1.8 <code>delarray</code> shortcuts.....	11
2 Saving and restoring a <code>tabu</code> : <code>\savetabu</code> , <code>\usetabu</code> and <code>\preamble</code>	12
3 Modifying the font and the alignment in one row: <code>\rowfont</code>	13
4 Differences between <code>tabu</code> , <code>tabular</code> , <code>tabularx</code> and <code>longtable</code>	14
4.1 Paragraph indentation.....	14
4.2 Custom environments.....	14
4.3 Inversion of tokens	14
4.4 Improved process for rewriting columns (<i>for keen readers</i>).....	14
5 The package options	16

This documentation is produced with the DocStrip utility, and required `tabu` with its `linegoal` option.

- To get the package, run: `etex tabu.dtx`
- To get the documentation run (thrice): `pdflatex tabu.dtx`
- To get the index, run: `makeindex -s gind.ist tabu.idx`

The `.dtx` file is embedded into this pdf file thank to `embedfile` by H. Oberdiek.

1. The documentation is compiled with `enumitem-zref`

5.1	The <code>debugshow</code> package option	16
5.2	The <code>linegoal</code> package option	16
5.3	The <code>delarray</code> package option	16
5.4	The <code>light</code> package option	16
6	Corrections of some bugs (<i>available only inside tabu</i>)	17
6.1	Correction for <code>colortbl</code> and <code>arydshln</code> : compatibility with <code>delarray</code>	17
6.2	Correction for <code>arydshln</code> : @ columns	17
7	To do for even better tabus	17
8	IMPLEMENTATION	18
8.1	Identification, requirements and options	18
8.2	Some constants	19
8.3	<code>\tabu</code> , <code>\endtabu</code> , <code>\longtabu</code> and <code>\endlontabu</code>	23
8.4	Footnotes inside <code>tabu</code>	25
8.5	Flow chart of expansion	27
8.5.1	<i>General case: tabu, tabu to and tabu spread</i>	
8.5.2	<i>tabu to with X column</i>	
8.5.3	<i>tabu spread with X column</i>	
8.6	<code>tabu X</code> column definition	29
8.7	Before trials to reach the target: setup after <code>\@mkpream</code>	31
8.8	One trial after another	32
8.9	Algorithms: Measuring the <code>tabu</code>	33
8.9.1	<i>The arithmetic of X columns</i>	
8.9.2	<i>The arithmetic of X columns for tabu spread</i>	
8.9.3	<i>Reporting in the .log file (debugshow option)</i>	
8.9.4	<i>Measuring the natural width for tabu spread and negativ width-coefficients</i>	
8.10	<code>\everycr</code> setup and <code>\tabuphantomline</code>	38
8.11	Lines inside <code>tabu</code>	39
8.11.1	<i>Vertical lines</i>	
8.11.2	<i>Horizontal lines: \tabucline</i>	
8.12	Verbatim inside <code>tabu</code> with X columns	43
8.13	Numbers in <code>tabu</code>	43
8.13.1	<i>\tabudecimal</i>	
8.14	<code>\savetabu</code>	44
8.15	<code>\rowfont</code>	45
8.15.1	<i>Setting font and alignment specification</i>	
8.15.2	<i>Preparing stuff to be able to use \rowfont</i>	
8.15.3	<i>Neutralisation of glues and alignment modification</i>	
8.16	Utilities	50
8.16.1	<i>tabu \fbox</i>	
8.16.2	<i>tabu \fcolorbox</i>	
8.16.3	<i>\centering, \raggedright, \raggedleft</i>	
8.17	Corrections	51
8.17.1	<i>delarray comptability fix for colortbl and arydshln</i>	
8.17.2	<i>arydshln @ columns</i>	
8.17.3	<i>arydshln, colors without colortbl and empty p columns</i>	
9	References	54
10	History	54
	[2011/01/15 v1.9]	54
	[2010/12/28 v1.8]	54
	[2010/12/18 v1.7]	54
	[2010/12/07 v1.5]	54
	[2010/11/22 v1.4]	55
	[2010/11/18 v1.3]	55
	[2010/11/15 v1.2]	55
	[2010/10/28 v1.1]	55
11	Index	55

Summary of the features provided by **tabu**

tabu	is like tabular in text mode and like array in math mode when there is no X column in its preamble
longtabu	is like longtable with the possibility to use tabu X columns and vertical lines with the extended syntax.
 [width,color]	vertical lines have an optional parameter.
{tabu} to $\langle dimen \rangle$	specifies the target width of the whole tabular. This is like tabular* with an automatic stretchability that can be overwritten with $\text{@}\{\backslash extracolsep\langle dimen \rangle\}$ in front of the preamble.
{tabu} spread $\langle dimen \rangle$	has no equivalent in L ^A T _E X: the final width is $\langle dimen \rangle$ wider than the natural width that can be obtained with spread Opt .
X[coef,align,type] X[coef,align,type,\$]	X columns widths are adjusted in order for the whole tabular to fit the target width. The target width is a dimension either: <ul style="list-style-type: none"> → directly specified with {tabu} to $\langle dimen \rangle$ → computed from the natural width: {tabu} spread $\langle dimen \rangle$ → by default \linewidth (or \linegoal with the linegoal package option). coef scales the widths of the X columns, if there are more than one X column. align is either r , c , l or j and type can be p (default), m or b . X[\$] makes a math X column (<i>ie.</i> $\langle \text{\$} \rangle \text{X} \langle \text{\$} \rangle$)
X[-coef,align,type]	X columns widths are first computed with the absolute value: coef . Then the width is made narrower down to the natural width of the column if possible. In any case, the final width does not exceed the one obtained with X[coef] . The final tabu can be narrower than its target if: <ul style="list-style-type: none"> • all coefs are negativ, and • no target has been specified (<i>ie.</i> the default target is used). In this case, tabu behaves as if it was said: {tabu} spread Opt .
\rowfont[align]{font spec}	Modify the font and optionally the alignment of each cell in one row.
\savetabu{user-name}	Saves the tabu preamble and its parameters. The command must appear at the end of a line: after \ or \hline .
\usetabu{user-name}	Makes a tabu of exactly the same shape as the one saved with \savetabu . All parameters (target , preamble , stretch <i>etc.</i>) are restored. This command is put alone in the preamble in place of the columns specifications.
\preamble{user-name}	Makes a tabu with the same preamble as the one saved with \savetabu . The preamble only is restored, not the target nor any other parameter. This command is put alone in the preamble in place of the columns specifications.
\tabuphantomline	inserts a phantom (<i>ie.</i> invisible) line inside the tabu May be usefull with \multicolumn in some cases.
\tabucline[spec]{start-stop}	Draws a line comparable to \hline . The line $\langle spec \rangle$ can contain information for making a dash or dotted line (f.ex. [on 3pt off 6pt]) and a color name. The line spec can also be defined with \tabulinestyle
\tabulinestyle{name=spec,...}	Defines a line style for use with \tabucline
\tracingtabu \tracingtabu=2	Reports informations in the .log file about the steps of the algorithm for tabu X columns, and the informations saved by \savetabu .

| **X** **\usetabu** and **\preamble** are defined as new column types loaded only inside the **\@mkpream** group inside the **tabu** environment.

1 The tabu environment

1.1 tabu, tabu to and tabu spread

```
\begin{tabu} [pos] {tabular preamble}
\begin{tabu} to <dimen> [pos] {tabular preamble}
\begin{tabu} spread <dimen> [pos] {tabular preamble}
```

The `tabu` environment behaves exactly like `tabular`: the preamble is parsed by the macros in `array.sty` with no modification. `tabu` improves `tabular` and `array`:

- **footnotes** and index words are allowed inside `tabu`, unlike `tabularx`, footnote links are not broken when compiled with `hyperref`. The syntax `\footnote[<number>]{<text>}` is allowed in `tabu` and `longtabu` (this is not implemented for `longtable` yet...)
- **X columns** are implemented with an *optional* parameter for the **width-coefficient** (which can be *negativ*: see next section), the **alignment** (`r`, `c`, `l`, or `j`, and `R`, `C`, `L` or `J` for `ragged2e` settings) and the **column type** (`p`, `m`, or `b`).
`tabu` has a *default target width* when used with **X columns**, making nesting even easier.
- You are used to the `tabular` environment in text mode, and `array` in math mode, but `tabu` works in both modes and its name does not change... **X columns** are also possible in math mode; `delarray` shortcuts for delimiters are available in both math and text modes.
- A `tabu` environment can contain another tabular of any kind: `tabular`, `tabular*`, `tabularx` or `tabu` itself can be placed in any cell of a `tabu`. Conversely, `tabu` can be placed in a `tabular`, `tabularx` *etc.*.
- `tabu` provides facilities for *vertical and horizontal lines*, and for the insertion of *verbatim text* inside **X columns**.
- `tabu` is more than compatible with `arydshln` (for dashed and dotted lines) and `colortbl`: actually some corrections of those packages are loaded as soon as you enter a `tabu` environment. Compatibility with `delarray`, `hhline`, `makecell`, `booktabs`, `siunitx`, `dcolumn`, `warpcol`, *etc.* is fine too. When you are inside a `tabu` environment, you can use `\raggedleft`, `\raggedright` and `\centering` without special care about `\arraybackslash` and conversely `\\` has its “normal” meaning inside a list of items that may appear in a **X column**...

`\begin{tabu} to<dimen>` is like `tabular*` but the inter-columns space is given a stretchability of `1fil`, in other words `@{\extracolsep{0pt plus 1fil}}` is inserted by default at the beginning of the tabular preamble, unless another value for `\extracolsep` is specified. Therefore “`tabu to`” fills in width the specified `<dimen>`.

`\begin{tabu} spread<dimen>` does a tabular whose width is `<dimen>` wider than its natural width. `@{\extracolsep{0pt plus 1fil}}` is inserted by default if `<dimen> > 0`.

1.2 longtabu, longtabu to and longtabu spread

```
\begin{longtabu} [l | c | r] {tabular preamble}
\begin{longtabu} to <dimen> [l | c | r] {tabular preamble}
\begin{longtabu} spread <dimen> [l | c | r] {tabular preamble}
```

`longtabu` is just like `tabu` but page breaks are allowed between rows of the table. `longtabu` is based on the `longtable` package which must be loaded, and all features of the `longtable` environment works inside `longtabu`: `\endhead`, `\endfirsthead`, `\endfoot`, `\endlastfoot` and `\caption`.

`longtabu` enhances the possibilities of `longtable` with the possibility to use **X columns** and line specification for *vertical rules*. `longtabu` is thus much easier than `ltxtable`.

The following commands provided for `tabu` do not work with `longtabu`:

tabu command	Not available	Not implemented	Comment
<code>\tabucline</code>		✱	<code>\tabucline</code> does not care of page breaks presently: use <code>\hline</code> instead.
<code>\usetabu</code>	✗		but <code>\savetabu</code> and <code>\preamble</code> work.
mathematical mode	✗		<code>longtable</code> is not designed to work in math mode.
delarray shortcuts	✗		a delimiter cannot be spanned over pages...
<code>\tabuphantomline</code>	✗		useless inside <code>longtabu</code>

However, `tabu` X columns, `\rowfont`, `\verbatim` and `\tabudecimal` work inside `longtabu`.

1.3 tabu X columns

`tabu` X columns can be viewed as an enhancement of `tabularx` X columns, but do not interact with them, for they are defined only for a short time during the parsing of the preamble:

- width coefficients** can optionally be given to X columns
ex. `X[2.5]X[1]` is the same as `X[2.5]X` and the same as `X[5]X[2]`
This means that the first X column will be two and a half wider than the second one or that the first X column width will be 5/7 of the whole tabular width.

X[2.5]	X
--------	---

- negativ width coefficients** can be given to X columns:
ex. `X[-2.5]X[1]` or `X[-2.5]X` or `X[-5]X[2]`
In this case, the first X column will be *at most* two and a half wider than the second one, and if the *natural width* of the first X column is finally less than 2.5 × (the width of the second column) then it will be narrowed down to this natural width.
The following tabus have the same preamble:

`\begin{tabu} to\linewidth{ | X[-2.5c] | X[c] | }:`

X[-2.5]	X
Negativ coefficients make X columns close to standard l, c and r columns.	X

- horizontal alignment specification is made easier with `X[5,r]X[2,c]` for example. Vertical alignment can be specified as well with `X[5,r,m]X[2,p,c]` (commas are not required, but `X[2cm]` or `X[4pc]` could be misunderstood – not by T_EX: by you...).

Modifier	Meaning	Default
l, c, r, j, L, C, R, J	left, centered, right, justified	j
p, m, b	X column is converted into p, m or b column	p
\$	X[\$] is a shortcut for: <code>>{\$}X<{\$}</code>	

- `tabu` X columns can be spanned with `\multicolumn`.
- `tabu` X columns can be used with “`tabu spread`” for small tabulars.
- `tabu` X columns can contain any type of `tabular`, `tabular*`, `tabularx` or `tabu` without special care about the syntax. `tabu` can also be put inside `tabular`, `tabular*` and `tabularx`. As long as `tabu` with X columns has a *default target*, nesting `tabu` with X columns is easy. Furthermore, the default global alignment of a nested `tabu` is t (for top) while the default global alignment of a `tabu` in a paragraph is c (for centered).
- The “algorithm” (or the arithmetic) to get the target width for `tabu` X columns is the same as the one used by `tabularx`. `\hfuzz` is the “tolerance” for the whole tabular width. We use ε -T_EX `\dimexpr` instead of T_EX primitives (with round/truncate bias correction).
- Convergence to the target width is optimised: the `\halign` preamble is not re-built at each trial, but only expanded again, until the target is reached. Though optimized, the process is the same as the one implemented for `tabularx` and in particular the content of

the `tabu` environment is collected as soon as a `tabu X` column is found in the preamble. This implies restrictions on catcode modifications and verbatim text inside a `tabu` with `X` columns.

- If the width of the whole tabular is not specified with “`tabu to`” it is considered to be `\linewidth`. The `linegoal` package option makes the default width equal to `\linegoal`. Compilation must then be done with pdf_TE_X either in `pdf` or `dvi` mode, and package `linegoal` is loaded. `\linegoal` requires pdf_TE_X for its `\pdfsavepos` primitive and the `zref-savepos`: if the `tabu` is not alone in its paragraph *ie.* if the target is not `\linewidth`, then two compilations (or more) are required to get the correct target. Default target for nested `tabu` environments is always `\linewidth`, which equals to the column width inside `p`, `m`, `b` and `X` columns.
- As long as the `\halign` content is expanded more than once, protections against counters incrementation, *whatsits* (*write*) index entries, footnotes *etc.* are set up: the mechanism of `tabularx` is reimplemented and enhanced for `tabu X` columns. `\tabuDisableCommands` can be used to neutralize the expansion of additional macros during the trials.

X columns with “`tabu spread`”

`tabu X` columns can be used with “`tabu spread`” to adjust the column widths of tabulars that contain only small pieces of text. The question is: how to make a tabular the width of the line, with 6 columns; the columns 1, 2, 5 and 6 are of equal widths and the widths of columns 3 and 4 are only one half. As possible solution:

```
\begin{tabu} to\linewidth{ |X[2]|X[2]|X|X|X[2]|X[2]| } \hline
1 & 2 & 3 & 4 & 5 & 6 \\\hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

But the text in each cell is very short: one single character, and you prefer the table to be tight, but don’t know the exact width of the whole:

```
\begin{tabu} spread 0pt{ |X[2]|X[2]|X|X|X[2]|X[2]| } \hline
1 & 2 & 3 & 4 & 5 & 6 \\\hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

But now it’s definitely too narrow, then give it some more space:

```
\begin{tabu} spread 2in{ |X[2]|X[2]|X|X|X[2]|X[2]| } \hline
1 & 2 & 3 & 4 & 5 & 6 \\\hline
\end{tabu}
```

1	2	3	4	5	6
---	---	---	---	---	---

`tabu spread` is useless with long columns: the following tabular was made with this preamble:

```
\begin{tabu} spread 3cm{@{}X[9]X[4]|X|}
```

"Like the air we breathe, Sherlock Holmes is everywhere. His pipe-smoking, deer stalked image peers at us from ads in Yellow Pages, to signs for neighbourhood crime-watch; from billboards to the classroom; from film and television to the public library, and now over the Internet. He long ago transcended the boundaries of 19th Century London² to become an international best-seller and has been accepted as part of British folklore. Holmes is alive to millions."

There the text was too long, and `tabu spread` behaves as if you didn’t give it a target.

The result of this example is the same as if one had written `\begin{tabu}to\linewidth.`

Sherlock Holmes

The “official” web site: <http://www.sherlockholmes.com/>

In the preamble, `@{}` means that the margin is removed.

2. Capital of the U.K. (too see a linked footnote)

Negativ width coefficients for X columns

```
\begin{tabu}{|[3pt ForestGreen]X[-1m]|X[c m]||[3pt ForestGreen]}
\tabucline[3pt ForestGreen]-
$\begin{tabu}({X[-1$]X[-1$c]})$ % equivalent to spread 0pt
\alpha & \beta \\
\gamma & \delta + \epsilon + \zeta + \eta + \theta
\end{tabu}$
&
This is a tabu with negativ width coefficients for \texttt X columns
\\ \tabucline[3pt ForestGreen]-
\end{tabu}
```

$\begin{pmatrix} \alpha & \beta \\ \gamma & \delta + \epsilon + \zeta + \eta + \theta \end{pmatrix}$	This is a tabu with negativ width coefficients for X columns
------------------------------------------------------------------------------------------------------	--------------------------------------------------------------

Multicolumn in tabu

`\tabuphantomline`

The process of `\multicolumn` implies the T_EX primitive `\omit` which discard the tabular preamble for the spanned columns. Discarding the preamble means discarding the information about the widths of the columns. This explains why the following example does not work properly:

```
\begin{tabu}{|X|X|X[2]|} \tabucline-
\multicolumn2{|c|}{Hello} & World \\ \tabucline-
\end{tabu}
```

Hello	World
-------	-------

The correct result can be obtained by the mean of a phantom line, that will remain invisible unless your preamble contains special `@` or `!` columns that prints some text:

```
\begin{tabu}{|X|X|X[2]|} \tabucline-
\multicolumn2{|c|}{Hello} & World \\ \tabucline-
\tabuphantomline
\end{tabu}
```

Hello	World
-------	-------

Remember you may need `\tabuphantomline` in conjunction with `\savetabu` and `\usetabu` with `\multicolumn`. Even if it is possible to add a `\tabuphantomline` in any line of the `tabu`, it is a good practice to append it *at the end* of the `tabu`, for it may introduce undesirable side effects on vertical alignment otherwise, when `tabu` is nested inside another tabular.

Nesting tabus with X columns

This section should contain some examples but I've no time presently...

However, this document has plenty of nested `tabus`!

1.4 Lines inside tabu

First important remarks







The features provided in this section are quite experimental: they are not generally taken for good typography. You can use `tabu` with package `booktabs` for example, which provides properly designed commands for horizontal rules in tabulars. `arydshln` is pretty good too, but it modifies a huge amount of macros of `array.sty`, something that `tabu` does not.

Lines in `tabu` printed in this document are mostly made with `booktabs`.

Vertical lines: | has an optional parameter

Inside `tabu` environment, the vertical line marker `|` has an *optional* argument which is the width of the vertical rule. The default width remains `\arrayrulewidth` of course. The optional argument for `|` can also contain the name of a color. *color names* are only possible, not a color specification by the mean of a color model. The width of the line if specified, must come before the color name and... as for **X** columns parameters, commas are optional.

Example:

 Hello  World 	The <code>tabu</code> you see on left <code>\begin {tabu}{ [5pt c c 5pt }</code> was made with the code <code>Hello & World</code> displayed on the right. <code>\end {tabu}</code>
 Hello  World 	The <code>tabu</code> you see on left <code>\begin {tabu}{ [5pt,red c c 5pt blue }</code> was made with the code <code>Hello & World</code> displayed on the right. <code>\end {tabu}</code>

This example was printed inside a `tabu` whose preamble is: `X[-1m]X[m]X[2m]`.

Note that it is always a good idea to protect the optional argument with braces: `{...}`. But it's not necessary because `tabu` takes care the `|` token to be rewritten before any other column type (just after `*` however, for obvious reasons). But if you use the optional argument for the vertical line into a user-defined column type (declared with `\newcolumntype` for example), you can get an error. In this case, it is compulsory to protect the optional argument by braces. Finally, it's not very often that a user-column type contains a vertical bar...

More style for horizontal lines: \tabucline

`\tabucline[<style or spec.>]{start-end}`

`\tabucline` is an attempt to give a versatile command to make horizontal lines:

- `\tabucline` is pretty good with vertical lines even if the thickness of the line grows up,
- `\tabucline` takes care of `\extrarowheight`,
- `\tabucline` can make horizontal dashed lines, with a `pgf/TikZ` syntax:
`\tabucline[<width> on<dash> off<gap>]{<first column>-<last column>}`
- alternatively, you can give `\tabucline` a `\hbox` to make a leader with it: The `<spec.>` must then begin with `\hbox`, `\box` or `\copy`,
- finally you can give `\tabucline` a *color name*, after the line specification.

Any parameter can be omitted.

[1pt on 1.5pt off 2pt]

[1.5pt]

default

[on 2pt red]

<code>\tabucline[1pt on 1.5pt off 2pt]{1-4}</code>	draws a horizontal dashed line of width 1pt . Dashes are 1.5pt long and gap width is 2pt . The line is drawn between columns 1 and 4. Here there are only 2 columns and the line stops at column 2.
<code>\tabucline[1.5pt]{-}</code>	draws a horizontal solid line of width 1.5pt between the first and the last column.
<code>\tabucline{2-}</code>	draws a horizontal solid line of width <code>\arrayrulewidth</code> between the second column and the last one.
<code>\tabucline[on 2pt red]{-5}</code>	draws a horizontal dashed line between columns 1 and 5 of width <code>\arrayrulewidth</code> . Dashed are 2pt long and gap width is 4pt (the default).

`\tabucline*{<style or spec.>}{start-end}`

For fine tuning, the star form `\tabucline*` can be used to keep the vertical lines that might cross the horizontal line. As a consequence, the content of special `@` of `!` columns will interrupt the horizontal line either. This might be usefull when `\extrarowheight` is high. Example:

Hello	world
Hello	world

`\tabucline`

Hello	world
Hello	world

`\tabucline*`

Hello	world
Hello	world

`\firsthline` and `\hline` and `\lasthline`

In each table, `\extrarowheight` is equal to 8 pt.





`\tabuclines` tries to put the line in the middle, so that the text is centered in its cell. `\tabuclines*` tries to take care of vertical lines. Well this is not perfect: it works pretty well with simple vertical lines, but not really with double lines, triple lines with colours etc... But it can help in simple cases...

`\tabulinestyle{<style=spec., style=spec., style=spec. ...>}`

This command defines a line style to be used in the first optional argument of `\tabucline`:

Define the line style	Use the line style
<code>\tabulinestyle{myline=0.4pt on 1.5pt off 1pt red}</code>	<code>\tabucline[myline]{-}</code>
<code>\tabulinestyle{myleader=\hbox{\$\scriptstyle\star\$}}</code>	<code>\tabucline[myleader]{1-3}</code>

This way it is easy to change dashed lines into solid ones

Dashed or dotted	Dash	Gap	Dash	Gap...	...or leader
And below is the default					
	This one was thick		Interesting ?		

Now we stop
it because things
are be- com-
ing too ugly
and go to have
a look at `tabu`
in mathematical
mode...

1.5 tabu in math mode

On the left, you can see the famous Maxwell-Lorentz equations for electromagnetic field in vacuum, published in 1873.	<div>$\left\{\begin{array}{l} \operatorname{div} \vec{E} = \frac{\rho}{\epsilon_0} \\ \operatorname{div} \vec{B} = 0 \\ \operatorname{rot} \vec{E} = -\frac{\partial \vec{B}}{\partial t} \\ \operatorname{rot} \vec{B} = \mu_0 \vec{j} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \end{array}\right.$</div>
-----------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

In this example, the big `tabu` is: `\begin{tabu} to\linewidth{X[1.5]X[r$]}`.

The nested `tabu` (in math mode) uses `delarray` shortcut: its preamble is:

`\begin{tabu}(<{\Gape[8pt]\displaystyle}r>{\displaystyle}l).`

You immediately see the interest for `tabu` to have a default target.

`\Gape` is a `makecell` macro to adjust the height and depth of the rows.

Horizontal rules are `booktabs` `\toprule`, `\midrule` and `\bottomrule`.

array	tabu	tabu spread 1em
$\begin{array}{ c c } \hline \alpha & \beta \\ \hline \gamma & \delta \\ \hline \end{array}$	$\begin{array}{ c c } \hline \alpha & \beta \\ \hline \gamma & \delta \\ \hline \end{array}$	$\begin{array}{ c c } \hline \alpha & \beta \\ \hline \gamma & \delta \\ \hline \end{array}$

Here, vertical lines are made with `delarray` shortcuts: `\begin{tabu} spread 1em |{cc}|`

Vertical lines inside the tabular preamble gives:

$\begin{array}{ c c } \hline \alpha & \beta \\ \hline \gamma & \delta \\ \hline \end{array}$

This was an example of `\savetabu... \usetabu` to keep the alignment.

1.6 Verbatim inside tabu with X columns

The process of computing X columns widths implies to make “trials”: this means that the tabular is first printed in internal mode, inside a box which is measured until the target width is reached. Such trials require to collect the content of the `tabu` environment. Therefore, the category codes of the characters read in the environment are fixed, and `\verb` commands are not allowed.

`\verbatim{<general text>}`

To get round this limitation, `tabu` provides the command `\verbatim` which allows to put some pieces of verbatim text inside a `tabu`: it is based on ϵ -T_EX `\scantokens` primitive and is defined only inside the `tabu` environment. `\verbatim` has some other limitations you must know:

<code>^^@</code>	can be used to get a carriage return inside <code>\verbatim</code>
<code>^</code>	consequently <code>^</code> is a normal subscript character
<code>{ }</code>	curly braces must be equilibrated
<code>%</code>	The percent character <code>%</code> is for commentaries: the content of the environment is collected before <code>\verbatim</code> does its job, therefore commentaries are removed...

If you need more verbatim inside a `tabu` environment:

- you should avoid the use of X columns
- or if you really want to use X columns, you must save your verbatim text before the `tabu`, for example with the `SaveVerbatim` environment provided by the package `fancyvrb`.

1.7 Printing numbers inside tabu with numprint and siunitx

Just make it easy !

`tabu` provides a *facility* to print numbers inside columns. This facility is not implemented to replace `siunitx` `S` and `s` columns or `numprint` `n` and `N` columns or other packages that provide alignment such as `warpcol`, `dcolumn` or `rccol`. It just make easy to apply a macro you get already on each number in a column of a `tabu`.

`\tabudecimal` has been developped mainly because it makes possible to align numbers inside `tabu` X columns.

`\tabudecimal{⟨user-macro⟩}`

`\tabudecimal` can be used in the preamble of a `tabu` before a column specification. The `⟨user-macro⟩` is a macro with one parameter that has to be defined before.

Example with `\numprint`:

```
\def\usermacro#1{\numprint[\officialeuro]{\zap@space #1 \@empty}}
\nprounddigits{2} \npprintnull \npthousandsep{\,} \npunitseparator{~}

\begin{tabu}{|[GreenYellow]*2>{\tabudecimal \usermacro }X[r]| [GreenYellow]}
\rowfont [c]{\bf } January & February \\
12.324 & 745.32 \\
21.13 & 0 \\
213.3245 & 12.342 \\
2143.12 & 324.325 \\
\end{tabu}
```

January	February	...
12,32 €	745,32 €	...
21,13 €	0,00 €	...
213,32 €	12,34 €	...
2 143,12 €	324,33 €	...

Example with `\SI`:

```
\def\usermacro#1{\SI[group-four-digits=true,           % thousand separator
                      round-mode=places,                 % round numbers
                      round-precision=2,                  % with 2 decimal digits
                      round-integer-to-decimal=true,      % add trailing 0 if necessary
                      per-mode=symbol]{#1}{\officialeuro\per\kilo\gram}}

\begin{tabu}spread Opt{|[GreenYellow]*2>{\tabudecimal \usermacro }X[r]| [GreenYellow]} ...
```

January	February	...
12.32 €/kg	745.32 €/kg	...
21.13 €/kg	0.00 €/kg	...
213.32 €/kg	12.34 €/kg	...
2 143.12 €/kg	324.33 €/kg	...

As you can see, the columns widths are exactly the same, whatever their content.

You should know how it works...

Yes you should know how it works to avoid problems. `tabu` has a small scanner based on `\futurelet` to grab all numbers, blank spaces, commas and dots + and – sign and also the letter `e` and `E` for exponents. The scanner stops as soon as something else than a number, blank space, comma, dot, +, –, `e`, `E` is found, and even if it is a macro that contains a number.

This explains why there is `\zap@space` in the definition of `\usermacro`: because the scanner scans blank spaces and because `\numprint` does not allow blank spaces in its mandatory argument, quite strangely...

1.8 delarray shortcuts

When you enclose your tabular with math delimiters using `delarray` shortcuts, `tabu` tries to reach its target for the whole: the tabular and the delimiter(s). You can see the difference:

with overflow hboxes
(15.8pt and 16.1pt two wide)
for tabularx

(This is `tabu` with `delarray` shortcut for parenthesis around)

(This is `tabularx` with `delarray` shortcut for parenthesis around)

{ This is `tabu` with `delarray` shortcut for curly brackets around }

{ This is `tabularx` with `delarray` shortcut for curly brackets around }

2 Saving and restoring a tabu: `\savetabu`, `\usetabu` and `\preamble`

`\savetabu{⟨user-name⟩}`

The command `\savetabu` can be used at the end of any line of a `tabu` environment to save the target width (or the spread), the columns specifications (tabular preamble) and the widths of `tabu X` columns (if there is any). This possibility allows to easily make tabulars which share exactly the same shape throughout your document. This can also be used as a kind of `tabbing` environment which is able to remember the tabs positions.

If the `⟨user-name⟩` has been used before, an info is displayed in the `.log` file and the previous settings are overwritten.

With the `debugshow` package option (or `\tracingtabu`) the informations saved by `\savetabu` are reported in the `.log` file.

`\usetabu{⟨user-name⟩}`

`\usetabu` is the complement of `\savetabu`: it can be put alone in the `tabu` preamble instead of the usual columns specifications to restore any previous settings saved with `\savetabu`. The `⟨user-name⟩` must exist otherwise, you get an error. `\usetabu` is a help to **make several tabulars of exactly the same shape, same target, same preamble**. The only parameter that can be changed is the optional vertical position parameter for the whole tabular.

`\usetabu` does not work with `longtabu`.

`\usetabu` locally restores:

- the preamble³.
- the vertical position `[c]`, `[b]` or `[t]`, unless another position is specified.
- the target width of the `tabu` in points: the saved target width does not contain any control sequence: it is fixed and stored in points.
- the width of `tabu X` columns: those widths are not calculated any more, even in the case of negativ coefficients and `X` columns are directly transformed into `p`, `m` or `b` columns of the same widths as the ones that where calculated at the time of `\savetabu`
- `\tabcolsep` (or `\arraycolsep` in math mode) and `\arraystretch`
- `\arrayrulewidth`, `\doublerulesep`, `\extrarowheight` and `\extratabsurround`
- `\minrowclearance`, `\arrayrulecolor` and `\doublerulesepcolor` (package `colortbl`)

Example:

```
\extrarowheight=5pt\tabcolsep=12pt\arrayrulecolor{ForestGreen}
\begin{tabu}to .7\linewidth{|XXX|X[c]|}\savetabu{mytabu}\tabucline*[on1pt]-
This & is & & tabu & & package\\ \tabucline*[on1pt]-
\end{tabu}
\arrayrulecolor{black}
```

This	is	tabu	package
------	----	------	---------

```
\begin{tabu}{\usetabu{mytabu}}\tabucline*[on1pt]-
\multicolumn{3}{|c|}{This is tabu} & package\\ \tabucline*[on1pt]-
\tabuphantomline
\end{tabu}
```

This is tabu	package
--------------	---------

If one day you use `tabu`, you will have the idea to restore a `tabu` while modifying its target, or

3. The complete `\halign`-preamble is restored.

adding new columns... `\savetabu` and `\usetabu` have not been thought for this purpose, and you may have unexpected results.

`\preamble{⟨user-name⟩}`

`\preamble` can also be used after `\savetabu`. This is a variant of `\usetabu` that locally restores:

- the `tabu` (or `longtabu`) preamble.
- the vertical position `[c]`, `[b]` or `[t]` (or `[c]`, `[l]` or `[r]` for `longtabu`), unless another position is specified.
- the `tabu` / `longtabu` target width, unless another target is specified.

Any other tabular parameter is not restored.

Put `\preamble{⟨user-name⟩}` alone inside the `tabu` (or `longtabu`) preamble in place of the usual columns specifications.

`\preamble` works exactly as if you defined a [custom environment](#) for `tabu`.

`\preamble` works with `longtabu`.

Example (continued...):

```
\begin{tabu} to\linewidth{\preamble{mytabu}} \tabucline[1pt off1pt]-
This      &is &tabu &package      \\\tabucline[1pt off1pt]-
\end{tabu}
```

This	is	tabu	package
------	----	------	---------

`\arrayrulecolor`, `\tabcolsep` *etc..* are not restored from `\savetabu`.

3 Modifying the font and the alignment in one row: `\rowfont`

`\rowfont[⟨alignment⟩]{font specification}`

Inside a `tabu` environment, you can modify the font for each cell in a row. `\rowfont` has priority over column font specification, exactly like `\rowcolor` (package `colortbl`) has priority over `\columncolor`.

The alignment of each cell in one row can also be changed to:

l = left	or for <code>ragged2e</code> settings:	L
c = center		C
r = right		R
j = justify		J

Any other value for the optional `⟨alignment⟩` parameter is silently ignored. If `ragged2e` is not loaded, L R C and J are synonymous with the lowercase equivalent.

```
\begin{tabu}{|X|X[-1]|} \tabucline-
\rowfont[c]\bfseries
This      &&Is      \\\tabucline[on 2pt,blue]-
\xpackage{tabu} &&package      \\\tabucline[off 2pt blue]-
\rowfont[r]\itshape
for      &&\texttt{tabu} and \texttt{longtabu} \\\tabucline-
\end{tabu}
```

This	Is
tabu	package
for	tabu and longtabu

4 Differences between tabu, tabular, tabularx and longtable

4.1 Paragraph indentation

tabu takes care of paragraph indentation when it is used with X columns and its default target, no matter if it has been loaded or not with the `linegoal` option. Example with L^AT_EX default: `\parindent = 20pt`.

This is `tabu` with its default target in an indented paragraph.

This is `tabu` with its default target, preceded by `\noindent`

This is `tabularx` with target: `\linewidth` in an indented paragraph.

This is `tabularx` with target: `\linewidth`, preceded by `\noindent`

4.2 Custom environments

Unlike `tabularx`, it is possible to define your own environment using `tabu`:

```
\newenvironment{foo}
  {\begin{tabu}{X[1.2]||1pt grayX}}
  {\end{tabu}}
```

`tabu` environment, even when X columns are used, may appear in the definition of your custom `tabular` environment.

You can also use the commands `\savetabu` and `\usetabu` for this purpose.

4.3 Inversion of tokens

When you typeset the following `tabular`:

```
\begin{tabular}{|>\bfseries>{ before }l<{ one }<{ two }|}
  cell content
\end{tabular}
```

You get the following result: | before **cell content** **two** **one** |

→ The word *before* is not bold, and **two** comes before **one**.

The reason is explained in the documentation of `array.sty`, and is related to the `array` environment in math mode when using `\newcolumntype`.

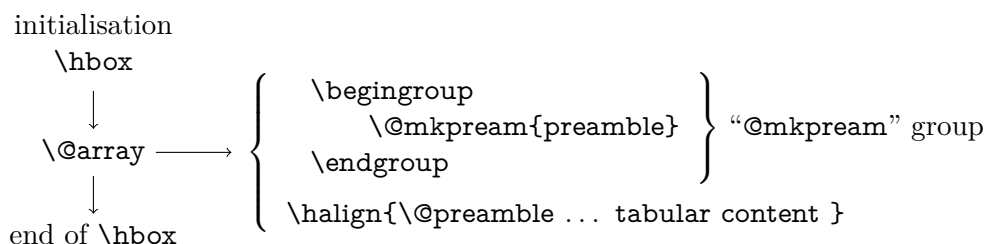
This rather strange inversion of tokens may be justified in math mode (otherwise, errors may occur) but not in text mode in our opinion. Inside a `tabu` environment, when not in math mode, the tokens are not reversed and you get the intuitively expected result:

| before cell content **one** **two** |

In math mode however, tokens are in the reverse order in the `tabu` environment like they are in the `array` environment.

4.4 Improved process for rewriting columns (*for keen readers*)

Any `tabular` that does not split accross pages is made with the following process:



For more details, see the [Flow chart of expansion](#).

`\@mkpream` works in two times inside a (semi-simple) group:

First the rewriting process:

Each special column in the tabular preamble is transformed into one the columns defined by `array.sty`.

Second the building of the `\halign` preamble:

The “rewritten preamble” is parsed and transformed in a preamble for the T_EX primitive `\halign`. The result is stored into the `\@preamble` macro.

Any special columns of `tabu` are defined only inside the “`@mkpream`” group.

In the following example, you get an error with `tabular` and no error with `tabu`. With `tabular`, and `siunitx` `S` column, the *rewritting process* is as follow:

<pre> \documentclass{minimal} \usepackage{numprint,siunitx,xcolor} \usepackage{tabu} \begin{document} \begin{tabular}{*2{S[color=green]}} 123,45 \end{tabular} \begin{tabu}{*2{S[color=green]}} 123,45 \end{tabu} \end{document} </pre>	<p>Inside <code>tabular</code>:</p> <ol style="list-style-type: none"> 1) Rewrite <code>S</code>: not found because inside <code>{...}</code> 2) Rewrite <code>*</code> 3) Rewrite <code>n</code> column defined by package <code>numprint</code> Then the ‘n’ in <code>green</code> is rewritten → problem <p>Inside <code>tabu</code>:</p> <ol style="list-style-type: none"> 1) Rewrite <code>*</code> 2) Rewrite <code> </code> (there is none here) <p>go back</p> <ol style="list-style-type: none"> 3) Rewrite <code>*</code> 4) Rewrite <code> </code> 5) Rewrite <code>S</code> 6) Rewrite <code>n</code> → not found because <code>S</code> was rewritten before, according to <code>siunitx</code> definition.
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The process of rewriting columns is usually longer inside `tabu` than inside `tabular`, but conversely `tabu` with `X` columns is optimised compared to `tabularx`, because the preamble is built only once, and not rebuilt before each trial as `tabularx` does. Thus `tabu` is much quicker than `tabularx`.

The process of rewriting is very sensitiv to the order in which columns are actually rewritten. This becomes critical when columns are defined with an optional argument like `tabu X` and `|` columns or `siunitx S` column.

If it possible to define a new column types using the `X` token for use with `tabu`:

```
\newcolumntype{C}{X[c]}
```

it is not recommended no nest such constructions like:

```
\newcolumntype{Q}{>{\color{green}}C}
```

In fact, a problem may arise in nested `tabus` if such a `Q` column is defined before the `C` column...

Well, just avoid to nest new column types definitions !

5 The package options

5.1 The debugshow package option

```
\tracingtabu
\tracingtabu=2
```

The control sequence `\tracingtabu` has the same effect as the `debugshow` option:

- `tabu` will report the widths it computes at each attempt to read the target, when `X` columns are used.
- Saved informations on the `tabu` are reported in the `.log` file when `\savetabu` is used.

Typical information in the `.log` file:

```
(tabu) Try      tabu X      tabu Width      Target      Coefs      Update
(tabu)  1)  386.67296pt    797.34592pt    386.67296pt    2.0pt    -205.33649pt
(tabu)  2)  181.33647pt    386.67294pt    386.67296pt    2.0pt     0.00002pt
(tabu)  2) Target reached (hfuzz=0.1pt) *****
```

What does it mean?

- 1) The first attempt was performed with `X=386.67296pt`
 The `tabu` width (797.34592pt) exceeded the target by 410.67296pt.
 Thus `X` has been updated: $410.67296\text{pt} / 2 = 205.33649\text{pt}$ and then:
 $X = 386.67296\text{pt} - 205.33649\text{pt} = 181.33647\text{pt}$
- 2) The second attempt lead to a `tabu` width of 386.67294pt: the target is reached.
 The final width of each `X` column is the product of `tabu X` by its width coefficient.

`\tracingtabu=2` gives even more information about the measuring of natural widths of columns.

5.2 The linegoal package option

With the `linegoal` option, the default target for `tabu` with `X` columns is `\linegoal` instead of `\linewidth`. The `linegoal` package must be loaded and compilation must be done with pdf_TE_X, otherwise, a warning is displayed and the `linegoal` option has no effect: the default target remains `\linewidth`. `\linegoal` works with pdf_TE_X in pdf mode **and in dvi** mode.

If for some reason, you wish to turn down the `linegoal` option in your document, you can say (in a group for example): `\let\tabudefaulttarget=\linewidth`

In any case, specifying the target overwrites the default: `\begin{tabu} to\linewidth`

5.3 The delarray package option

`delarray` option has the single effect to load `delarray.sty` for delimiters shortcuts around `tabu`. Delimiters shortcuts work both in math and text mode.

5.4 The light package option

When you enter a `tabu` environment, two macros amongst the smallest possible of `array.sty` are modified: these are `\prepnext@tok`, which is expanded while `\@mkpream` builds the `\halign` preamble and `\save@decl` to avoid inversion of tokens in text mode (see [Inversion of tokens](#)).

Modification of `\prepnext@tok` is loaded only inside the `tabu` environment, while the modification of `\save@decl` is loaded only inside the group in which `\@mkpream` works: this is very very local to `tabu` and cannot interfere with any other tabular !

`\prepnext@tok` could have been loaded inside the `\@mkpream` group as well, but then `\rowfont` would not have worked inside `array` or `tabular` nested into a `tabu`. Thus the choice.

As you see, these modifications do no modify `tabular`, `tabular*`, `tabularx`, `longtable` *etc.etc.etc.*

If for some reason you prefer the original macros of `array.sty`, then you can load `tabu` with the “light” option. As a consequence: `\rowfont` and `\tabucline` will not be available, and vertical

lines | will not have an optional argument for their widths and color. However, `tabu X` columns will work normally.

6 Corrections of some bugs (*available only inside tabu*)

6.1 Correction for `colortbl` and `arydshln`: compatibility with `delarray`

Both `colortbl` and `arydshln` forget the control sequence `\@arrayright` in their implementation, quite strangely because both of them take care of `\@arrayleft`. As a result, `delarray` shortcuts for delimiters around a tabular does not work if `colortbl` and/or `arydshln` are loaded.

Those control sequences are used by the `delarray` package to put variable size delimiters around the array:

<code>\begin {tabu}\{{X}.</code>		<code>\left \{ \begin {tabu}{X}</code>
<code>...</code>	is like:	<code>...</code>
<code>\end {tabu}</code>		<code>\end {tabu} \right .</code>

6.2 Correction for `arydshln`: @ columns

A bug in `\adl@xarraydashrule: !-arg` columns (class 1) and `@-arg` columns (class 5) should be treated the same as far as rules are concerned.

With this correction, the “known problem number 1” in `arydshln` documentation is solved.

7 To do for even better tabus

In decreasing order of priority:

- ➡ Implement `\tabulineskiplimit` in a manner close to `tbls \tabulinesep` or `cellspace S` columns-modifier.
- ➡ Make double `\tabucline` compatible with `colortbl \doublerulesepcolor`
- ➡ Multiple `\tabucline` between different columns: extended specs:
`\tabucline[(line spec)]{start-stop, start-stop}[(line spec)]{start-stop} ...`
- ➡ Presently, `longtabu` with `X` columns works only if `\LTchunksize` is greater than the number of rows. I compiled a `longtabu` of 56 pages on my PC with `\LTchunksize = 2 000` without problem. Presently `\LTchunksize` is set to 10 000 during trials when `longtabu` contains `X` columns.
- ➡ Make `\tabucline` work with page breaks (one line on the top of the page, one line on the bottom of the previous).
- ➡ An environment “`calstabu`” based on the `cals` package but with the syntax of `tabular` if possible...

8 Implementation

8.1 Identification, requirements and options

The package namespace is **tabu@**.

```

1 <*package>
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{tabu}[2011/01/15 v1.9 - flexible LaTeX tabulars (FC)]
4 \RequirePackage{array}[2008/09/09]
5 \RequirePackage{varwidth}[2009/03/30]

```

Minimal catcode accertaining for loading **tabu** in good conditions:

```

6 \AtEndOfPackage{\tabu@AtEnd\let\tabu@AtEnd\@undefined}
7 \let\tabu@AtEnd\@empty
8 \def\TMP@EnsureCode#1#2{%
9   \edef\tabu@AtEnd{%
10     \tabu@AtEnd
11     \catcode#1 \the\catcode#1\relax
12   }%
13   \catcode#1 #2\relax
14 }% \TMP@EnsureCode
15 \TMP@EnsureCode{33}{12} % !
16 \TMP@EnsureCode{124}{12}% |
17 \TMP@EnsureCode{0}{12}% ^^@
18 \TMP@EnsureCode{36}{3}% $ = math shift

```

\tracingtabu (debugshow option) **\tracingtabu** is the same as option debugshow.

```

19 \let\tabu@message      \@gobble
20 \let\tabu@message@save \@gobble
21 \let\tabu@debug        \@gobble
22 \let\tabu@timerstart   \relax
23 \def\tracingtabu{%
24   \let\tabu@message \message
25   \let\tabu@message@save \tabu@tracing@save
26   \ifdefined \pdfelapsedtime
27     \def\tabu@timerstart{\edef\tabu@starttime{\the\pdfelapsedtime}}%
28     \let\tabu@elapsedtime \tabu@message@etime\fi
29   \@ifnextchar={\tabu@tracing}{}}
30 \def\tabu@tracing{\begingroup\afterassignment\tabu@tr@cing\count@
31 \def\tabu@tr@cing{\expandafter\endgroup\ifnum\count@>\@ne
32   \let\tabu@debug\message\else\let\tabu@debug\@gobble\fi}
33 \DeclareOption{debugshow}{\AtEndOfPackage{\tracingtabu\relax}}

```

linegoal (package option)

```

34 \let\tabubefaulttarget \linewidth
35 \DeclareOption{linegoal}{%
36   \AtEndOfPackage{%
37     \RequirePackage{linegoal}[2010/12/07]%
38     \let\tabubefaulttarget \linegoal% \linegoal is \linewidth if not pdfTeX
39   }}

```

delarray (package option)

```

40 \DeclareOption{delarray}{%
41   \AtEndOfPackage{\RequirePackage{delarray}}%
42 }

```

light (package option)

```

43 \DeclareOption{light}{%

```

```

44 \AtEndOfPackage{%
45 \let\tabu@prepnext@tok \prepnext@tok \let\tabu@save@decl \save@decl
46 \let\tabu@rowfont \tabu@norowfont \let\tabu@cline \tabu@nocline
47 \let\tabu@firstcline \relax \let\tabu@lines \relax
48 \let\tabu@setup@everycr\relax
49 }
50 }
51 \def\tabu@norowfont{\PackageError{tabu}
52 {\string\rowfont\space is not available with option ‘light’}\@ehd}
53 \def\tabu@nocline{\PackageError{tabu}
54 {\string\tabu@cline\space is not available with option ‘light’}\@ehd}

55 \ProcessOptions

```

At Begin Document, a fix for `arydshln` and `colortbl` compatibility with `delarray` shortcuts available inside `tabu`: requirement for this fix is checked by `\tabu@fix@arrayright`.

Then the switch `\iftabu@colortbl` is set.

Finally, `\longtabu` is defined only if the `longtable` package is detected.

```

56 \AtBeginDocument{%
57 \expandafter\in@ \expandafter\@arrayright\expandafter{\endarray}%
58 \ifin@ \let\tabu@endarray \endarray
59 \else \tabu@fix@arrayright \fi % <fix for colortbl & arydshln (delarray)>
60 \@ifpackageloaded{colortbl} \tabu@colortbltrue \tabu@colortblfalse
61 \@ifpackageloaded{arydshln}{%
62 {\let\tabu@adl@fix \relax \let\tabu@adl@endTRIAL \@empty}%
63 \@ifpackageloaded{longtable}{\let\longtabu \tabu@nolongtabu}%
64 }

```

8.2 Some constants

<code>\tabu@cnt</code>	Used in <code>\tabu@arith</code> (the number of trials) and <code>\tabu@prepnext@tok</code> (for <code>\rowfont</code>).
<code>\tabu@nbcols</code>	A counter that save the number of columns of the <code>tabu</code> .
<code>\tabu@X@cols</code>	Used only when <code>tabu X</code> columns are used with “ <code>tabu spread</code> ”.
<code>\tabu@nested</code>	A counter that stores the deepness of nested <code>tabu</code> ’s.
<code>\tabu@start</code> <code>\tabu@stop</code>	Counters used for <code>\tabu@cline</code> to get the starting and ending columns. <pre> 65 \newcount \tabu@cnt 66 \newcount \tabu@nbcols 67 \newcount \tabu@X@cols 68 \newcount \tabu@nested 69 \newcount \tabu@start 70 \newcount \tabu@stop </pre>
<code>\tabu@target</code>	Stores the <code>tabu</code> target (either “ <code>to</code> ” or “ <code>spread</code> ”).
<code>\tabu@spreadtarget</code> <code>\tabu@naturalX</code>	Used only when <code>tabu X</code> columns are used with “ <code>tabu spread</code> ”, or with negativ coefficients. <pre> 71 \newdimen \tabu@target 72 \newdimen \tabu@spreadtarget 73 \newdimen \tabu@naturalX </pre>
<code>\tabu@colX</code>	The <code>dimen</code> corresponding to the preamble token <code>X[1]</code> : the standard width of <code>X</code> columns.
<code>\tabu@X@sum</code>	Stores the sum of all width coefficients for <code>X</code> columns.
<code>\tabu@colX@error</code>	In case of <code>X</code> columns too narrow. <pre> 74 \newdimen \tabu@colX 75 \newdimen \tabu@X@sum </pre>

	76 \newdimen \tabucolX@error \tabucolX@error=1em
\iftabu@measuring	This switch is set to true by \tabu@arith if the trial did not reached the target. It is also temporarily set to true when the first X column is encountered in the tabu preamble, at the time \mkpream scans it to built the \halign preamble. The first X column found actually triggers some special setup to be expanded before \halign (see the flow chart...)
\iftabu@spread	A switch whether “tabu spread” is used or not.
\iftabu@firstcline	A switch to adapt \tabucline automatically if it comes first in the tabu (similar to \firsthline).
\iftabu@negcoef	A switch set to true in case of negativ coef (natural width if less than X[coef]). 77 \newif \iftabu@measuring 78 \newif \iftabu@spread 79 \newif \iftabu@firstcline 80 \newif \iftabu@negcoef
\tabu@box	Stores the whole tabu when an attempt to adjust X columns is performed. It is also used by \tabucline to save the \@arstrutbox when inserting a horizontal line. 81 \newsavebox \tabu@box 82 \chardef \tabu@arstrutbox=\@arstrutbox 83 \let\tabu@vcenter \vcenter
\tabu@gobblespace	A macro which is needed when scanning tokens with \futurelet.
\tabu@gobbleopt	A macro that gobbles an optional argument. 84 \def\tabu@gobblespace#1 {#1} 85 \def\tabu@gobbleopt#1{\@ifnextchar [{\tabu@gobblept{#1}}{#1}}% 86 \def\tabu@gobblept#1[#2]{#1}
\tabu@save@decl	No inversion on tokens in the tabu preamble, when not in math mode. 87 \def\tabu@save@decl{% 88 \toks\count@ =\expandafter{\the\toks\expandafter\count@ \@nextchar}% 89 }% \tabu@save@decl
\tabu@newcolumntype	A helper macro to create new column types for tabu. The column types are not appended to \NC@list in order to keep them local to tabu.
\tabu@privatecolumntype	Columns types defined with \tabu@privatecolumntype are "mounted" only inside the \@mkpream group of tabu. 90 \def\tabu@newcolumntype #1{% 91 \expandafter\tabu@new@columntype 92 \csname NC@find@\string#1\expandafter\endcsname 93 \csname NC@rewrite@\string#1\endcsname 94 {#1}% 95 }% \tabu@newcolumntype 96 \def\tabu@new@columntype #1#2#3{% 97 \def##1##3{NC@{##1}}% 98 \let#2\relax \newcommand*#2% 99 }% \tabu@new@columntype 100 \def\tabu@privatecolumntype #1{% 101 \expandafter\tabu@private@columntype 102 \csname NC@find@\string#1\expandafter\endcsname 103 \csname NC@rewrite@\string#1\expandafter\endcsname 104 \csname tabu@NC@find@\string#1\expandafter\endcsname 105 \csname tabu@NC@rewrite@\string#1\endcsname 106 {#1}% 107 }% \tabu@privatecolumntype 108 \def\tabu@private@columntype#1#2#3#4{%

```

109 \g@addto@macro\tabu@privatecolumns{\let#1#3\let#2#4}%
110 \tabu@new@columnntype#3#4%
111 }% \tabu@private@columnntype
112 \let\tabu@privatecolumns \empty

```

X (private column type) This is the rewrite macro for tabu X columns. Such a column has an optional argument: the width coefficient for the tabu X column whose default value is 1, and may be some alignments parameters. The coefficient is used in the expression: `p{\dimexpr<coef>\tabucolX}`

```

113 \tabu@privatecolumnntype X[1][]{\tabu@rewrite@X{#1}%
114 \let\@halignto \relax
115 \expandafter \NC@find \tabucolX@rw}

```

The next part of the definition (`\tabu@rewrite@X`) can be found page 29.

\usetabu (private column type) `\usetabu` is defined as a tabu new column type: loaded only inside the `\@mkpream` group inside the tabu environment.

```

116 \tabu@privatecolumnntype \usetabu [1]{%
117 \ifx\#1\\\tabu@saveerr{}\else
118 \ifundefined\tabu@saved@string#1}
119 {\tabu@saveerr{#1}}
120 {\let\tabu@rewrite@X \tabu@rewrite@Xrestore
121 \csname tabu@saved@string#1\expandafter\endcsname\expandafter\z}%
122 \fi
123 }% \NC@rewrite@\usetabu

```

\preamble (private column type) `\preamble` is defined as a tabu new column type: loaded only inside the `\@mkpream` group inside the tabu environment.

```

124 \tabu@privatecolumnntype \preamble [1]{%
125 \ifx\#1\\\tabu@saveerr{}\else
126 \ifundefined\tabu@saved@string#1}
127 {\tabu@saveerr{#1}}
128 {\csname tabu@saved@string#1\expandafter\endcsname\expandafter\@ne}%
129 \fi
130 }% \NC@rewrite@\preamble

```

\tabu@rewritefirst (new column type) This new column type is not really a column type! It is always added to a tabu preamble in order to do some setup before any other column is rewritten by `\@mkpream`.

Thus, `\NC@do\tabu@rewritefirst` is added **at the beginning of** `\NC@list` at the entry of a (not nested) tabu environment.

This “column type” sets up the new column type `\tabu@rewritelast` which is added to **at the end of** `\NC@list`, and defines the token X to be rewritten by `\tabu@NC@rewrite@X` (in case `tabularx` is used with `tabu`, this modification of the X column occurs only inside the group where `\@mkpream` does its job).

```

131 \tabu@newcolumnntype \tabu@rewritefirst{%
132 \tabu@X@cols \z@
133 \let\tabu@dimexpr \relax
134 \let\tabu@savepreamble \relax
135 \let\NC@rewrite@X \tabu@NC@rewrite@X
136 \gdef\tabu@global@temp {\tabu@savepreamble}%
137 \aftergroup \tabu@global@temp
138 \let\tabu@islast \relax
139 \tabu@privatecolumns
140 \NC@list{\NC@do\usetabu \NC@do\preamble \NC@do *}%
141 \tabu@lines % defines NC@rewrite@ for tabu only (inside @mkpream group)
142 \NC@list\expandafter{\the\expandafter\NC@list \expandafter\NC@do \expandafter X%
143 \tabu@NC@list \NC@do X\NC@do\tabu@rewritemiddle}%
144 \ifcat$\dollar\end\else
145 \let\save@decl \tabu@save@decl \fi % no inversion of tokens in text mode

```

```

146 \NC@find \tabu@rewritemiddle
147 }% \NC@rewrite@\tabu@rewritefirst
148 \def\tabu@savepreamble{%
149 \gdef\tabu@global@temp{}}%
150 \NC@list\expandafter{\tabu@NC@list}%
151 \let\tabu@savedpreamble \@preamble
152 }% \tabu@savepreamble

```

`\tabu@rewritelast` (new column type) This new column type is rewritten after X columns, because it is declared by when the column `\tabu@rewritefirst` is actually rewritten. In the case where `\tabu@target` is > 0 (either because of “`tabu to`” or “`tabu spread`” has been called) and if there is no X column, then `@{\extracolsep}\@flushglue` is added at the beginning of the preamble.

To avoid duplicate margin in the `tabu` we have to test the next token in the preamble. If the next token is `|` or `!` then no margin must be added and `@{\extracolsep}\@flushglue` can be inserted at the beginning of the preamble.

Otherwise, we must insert `!\extracolsep\@flushglue` in order to keep the margin.

`\tabu@rewritelast` column type is loaded by `\tabu@rewritefirst` column type, only inside the `\@mkpream` group inside the `tabu` environment.

```

153 \tabu@newcolumnntype \tabu@rewritemiddle{%
154 \NC@list\expandafter{\tabu@NC@list \NC@do\tabu@endrewrite \NC@do\tabu@rewritelast}%
155 \tabu@checklast
156 }% \NC@rewrite@\tabu@rewritemiddle
157 \def\tabu@checklast#1\relax{%
158 \def\tabu@temp{#1}%
159 \ifx \tabu@temp\tabu@islast
160 \expandafter\tabu@lastrewrite
161 \else \let\tabu@islast \tabu@temp
162 \expandafter\NC@find \expandafter\tabu@rewritelast
163 \fi
164 #1\relax
165 }% \tabu@checklast
166 \tabu@newcolumnntype \tabu@rewritelast{\NC@find \tabu@endrewrite}
167 \tabu@newcolumnntype \tabu@endrewrite{\tabu@checklast}
168 \def\tabu@lastrewrite{%
169 \let\tabu@next \NC@find
170 \ifx \@halignto\relax % found a X column
171 \edef\tabu@global@X{\tabu@global@X \tabu@X@cols\the\tabu@X@cols\relax}%
172 \ifx \tabu@halignto\relax % X column measure
173 \tabu@global@Xnested
174 \iftabu@negcoef \aftergroup\tabu@negcoeftrue \fi
175 \aftergroup \tabu@prep@TRIAL
176 \else \aftergroup \tabu@setup@everycr
177 \fi
178 \else
179 \aftergroup \tabu@setup@everycr
180 \ifdim \tabu@target=\z@ \else
181 \let\tabu@next \tabu@extracolsep \fi % <need \extracolsep ?>
182 \fi
183 \let\@halignto \tabu@halignto
184 \tabu@next
185 }% \tabu@lastrewrite
186 \def\tabu@extracolsep{\futurelet\tabu@temp \tabu@extracol@sep}
187 \def\tabu@extracol@sep{%
188 \ifx \tabu@temp\@sptoken
189 \expandafter\tabu@gobblespace \expandafter\tabu@extracol@sep
190 \else
191 {\endlinechar\m@ne\scantokens{\let\x |}}%

```

```

192     \expandafter}\ifx \x\tabu@temp
193     \def\tabu@next {\NC@find @{\extracolsep\@flushglue}}\else
194     {\endlinechar\m@ne\scantokens{\let\x !}}%
195     \expandafter}\ifx \x\tabu@temp
196     \def\tabu@next {\NC@find @{\extracolsep\@flushglue}}\else
197     \def\tabu@next {\NC@find !{\extracolsep\@flushglue}}\fi\fi
198     \expandafter\tabu@next
199     \fi
200 }% \tabu@extracol@sep
201 \def\tabu@global@Xnested{\expandafter\tabu@global@X@nested
202     \csname tabu@global@X\the\tabu@nested\endcsname}
203 \def\tabu@global@X@nested#1{%
204     \global\let#1\tabu@global@X
205     \aftergroup\let \aftergroup\tabu@global@X \aftergroup#1%
206 }% \tabu@global@Xnested

```

`\iftabu@colortbl` The switch `\iftabu@colortbl` is used by `\rowfont` when modifying the alignment, because `colortbl` changes the glue put inside the `\halign` preamble to make standard alignments. This switch is set At Begin Document.

```
207 \newif\iftabu@colortbl
```

`\tabu@nowrite` A trick (from the T_EX-book) to forbidd `\write` when a trial is done on the `\halign`.

```

208 \def\tabu@nowrite#1#{{\afterassignment}\toks@}
209 \let\tabu@write\write
210 \let\tabu@immediate\immediate
211 \protected\def\tabu@GenericError{\begingroup
212     \def\immediate\write{\aftergroup\endgroup
213         \tabu@immediate\tabu@write}\tabu@GenericErrorORI}%
214 \let\tabu@GenericErrorORI \GenericError
215 \protected\def\tabu@warn{\begingroup
216     \def\immediate\write{\aftergroup\endgroup
217         \tabu@immediate\tabu@write}\PackageWarning{tabu}}

```

8.3 `\tabu`, `\endtabu`, `\longtabu` and `\endlontabu`

`\tabu` `\tabu` is the command of the environment.

`\endtabu` `\endtabu` is `\endtabular` or `\endarray` in math mode.

```

218 \def\tabu{%
219     \ifmmode \def\tabu@{\array}\def\endtabu{\endarray}%
220     \else \def\tabu@{\tabular}\def\endtabu{\endtabular}\fi
221     \let\tabu@arrayleft@measure \tabu@arrayleftmeasure
222     \tabu@setup \tabu@settarget
223 }% \tabu
224 \def\longtabu{%
225     \ifmmode\PackageError{tabu}{longtabu not allowed in math mode}\fi
226     \def\tabu@{\longtable}\def\endlontabu{\endlongtable}%
227     \let\tabu@arrayleft@measure \tabu@long@arrayleft@measure
228     \tabu@setup \tabu@settarget
229 }% \longtabu
230 \def\tabu@nolongtabu{\PackageError{tabu}
231     {longtabu requires the longtable package}\@ehd}

```

`\tabu@setup` This macro sets the `tabu` X column definition at the beginning of the `tabu` environment.

```

232 \def\tabu@setup{%
233   \tabu@adl@fix \let\endarray \tabu@endarray % <fix> colortbl & arydshln (delarray)
234   \let\tabu@global@X \@empty
235   \ifx \verbatim\tabu@sanitizetext      % <nested tabu>
236     \tabu@setup@saveglobal
237     \advance\tabu@nested \@ne
238     \def\tabu@aligndefault{t}\let\tabu@defaulttarget \linewidth
239   \else \tabu@timerstart
240     \def\tabu@aligndefault{c}\edef\tabu@hfuzz {\the\hfuzz}%
241     \global\tabu@footnotes{}\aftergroup\the\aftergroup\tabu@footnotes
242     \ifdim\parindent>\z@ \ifx\linewidth\tabu@defaulttarget
243       \everypar\expandafter{%           % correction for indentation
244         \the\everypar\everypar\expandafter{\the\everypar}%
245         \setbox\z@=\lastbox
246         \ifdim\wd\z@>\z@ \advance\linewidth -\wd\z@\fi
247         \box\z@
248       }%
249     \fi\fi
250   \fi
251   \let\@footnotetext \tabu@footnotetext \let\@xfootnotetext \tabu@xfootnotetext
252   \let\@xfootnote \tabu@xfootnote \let\centering \tabu@centering
253   \let\raggedright \tabu@raggedright \let\raggedleft \tabu@raggedleft
254   \tabu@trivlist %<restore \\=\@normalcr inside lists>
255   \def\tabu@decimal {\tabu@tabu@decimal}\let\verbatim \tabu@sanitizetext
256   \let\fbbox \tabu@fbbox \let\color@b@x \tabu@color@b@x
257   \let\rowfont \tabu@rowfont
258   \let\prepnext@tok \tabu@prepnext@tok %<for \rowfont and \tabu@cline>
259   \everycr{}\let\tabu@everycr \everycr
260   \tabu@spreadfalse \tabu@measuringfalse \tabu@negcoeffalse
261   \edef\tabu@NC@list{\the\NC@list}\NC@list{\NC@do \tabu@rewritefirst}%
262 }% \tabu@setup
263 \def\tabu@setup@saveglobal{%
264   \expandafter\xdef\csname tabu@global@\the\tabu@nested\endcsname{%
265     \global\tabu@X@cols \the\tabu@X@cols\relax
266     \global\tabu@naturalX \the\tabu@naturalX\relax
267     \gdef\noexpand\tabu@naturalX@min {\tabu@naturalX@min}%
268     \gdef\noexpand\tabu@naturalX@max {\tabu@naturalX@max}}%
269   \expandafter\aftergroup\csname tabu@global@\the\tabu@nested\endcsname
270 }% \tabu@setup@saveglobal

```

`\tabu@settarget` The macro sets `\tabu@target` (a dimen) to the value specified for “`tabu to`” or “`tabu spread`”.

```

\tabu@begin
271 \def\tabu@settarget{\futurelet\@let@token \tabu@settargetget}
272 \def\tabu@settargetget{\tabu@target\z@
273   \ifcase \ifx \bgroup\@let@token 0\else
274     \ifx [\@let@token 0\else
275     \ifx \@sptoken\@let@token 1\else
276     \if t\@let@token 2\else
277     \if s\@let@token 3\else
278     \m@ne\fi\fi\fi\fi\fi\relax
279   \expandafter\tabu@begin
280   \or \expandafter\tabu@gobblespace\expandafter\tabu@settarget
281   \or \expandafter\tabu@to
282   \or \expandafter\tabu@spread
283   \else \expandafter\tabu@begin
284   \fi
285 }% \tabu@settargetget

```

```

286 \def\tabu@to to{\def\tabu@halignto{to}\tabu@gettarget}
287 \def\tabu@spread spread{\tabu@spreadtrue\def\tabu@halignto{spread}\tabu@gettarget}
288 \def\tabu@gettarget {\afterassignment\tabu@linegoaltarget \tabu@target}
289 \def\tabu@linegoaltarget {\futurelet\tabu@temp \tabu@linegoal@target}
290 \def\tabu@linegoal@target {%
291   \ifx \tabu@temp\LNG\setlinegoal
292     \LNG\setlinegoal \expandafter \@firstoftwo \fi % @gobbles \LNG\setlinegoal
293   \tabu@begin
294 }% \tabu@linegoal@target
295 \def\tabu@begin #1{%
296   \edef\tabu@halignto{\ifdim\tabu@target>\z@ \tabu@halignto\the\tabu@target\fi}%
297   \@testopt \tabu@@begin \tabu@aligndefault #1\@nil
298 }% \tabu@begin
299 \def\tabu@@begin [#1]#2\@nil#3{%
300   \tabu@preparesave{#3}% <for \savetabu>
301   \def\tabu@align {#1}%
302   \tabu@ [\tabu@align]#2{\tabu@rewritefirst #3}%
303 }% \tabu@@begin

```

`\tabu@preparesave` This macro temporarily saves the `tabu` preamble and the settings for `\arrayrulewidth`, `\extrarowheight` *etc.*.

Those values are retrieved by `\savetabu`, if invoked inside the `tabu`.

```

304 \def\tabu@preparesave #1{%
305   \iftabu@colortbl \temptokena{}%
306   \tabu@prepare@save\CT@arc@
307   \tabu@prepare@save\CT@drsc@
308   \fi
309   \edef\tabu@savedparam{%
310     \ifmmode \col@sep \the\arraycolsep
311     \else \col@sep \the\tabcolsep \fi\relax
312     \arrayrulewidth \the\arrayrulewidth\relax
313     \doublerulesep \the\doublerulesep\relax
314     \extrarowheight \the\extrarowheight\relax
315     \extratabsurround \the\extratabsurround\relax
316     \def\noexpand\arraystretch {\arraystretch}%
317     \iftabu@colortbl \minrowclearance\the\minrowclearance\relax
318     \the\temptokena
319   \fi}%
320   \def\tabu@savedpream {\edef\tabu@halignto{to\the\tabu@target}\NC@find #1}%
321 }% \tabu@preparesave
322 \def\tabu@prepare@save #1{%
323   \ifx\relax#1\else
324     \temptokena\expandafter{\the\expandafter\temptokena
325       \expandafter\def\expandafter#1\expandafter{#1}}\fi
326 }% \tabu@prepare@save

```

8.4 Footnotes inside tabu

`\tabu@footnotes` The (global) token that stores the footnotes inside `tabu`:

```

327 \newtoks\tabu@footnotes

```

`\tabu@footnotetext` The macros in case `hyperref` is not used, or used with the option `hyperfootnotes=false`:

```

328 \long\def\tabu@footnotetext #1{%
329   \edef\@tempa{\the\tabu@footnotes
330     \noexpand\footnotetext [\the\csname c@\@mpfn\endcsname]}%
331   \global\tabu@footnotes\expandafter{\@tempa {#1}}}%
332 \long\def\tabu@xfootnotetext [#1]#2{%
333   \global\tabu@footnotes\expandafter{\the\tabu@footnotes
334     \footnotetext [{#1}]{#2}}
335 \let\tabu@xfootnote \@xfootnote

```

`\tabu@Hy@ftntext` The macros in case `hyperref` is loaded with the option `hyperfootnotes=true`:

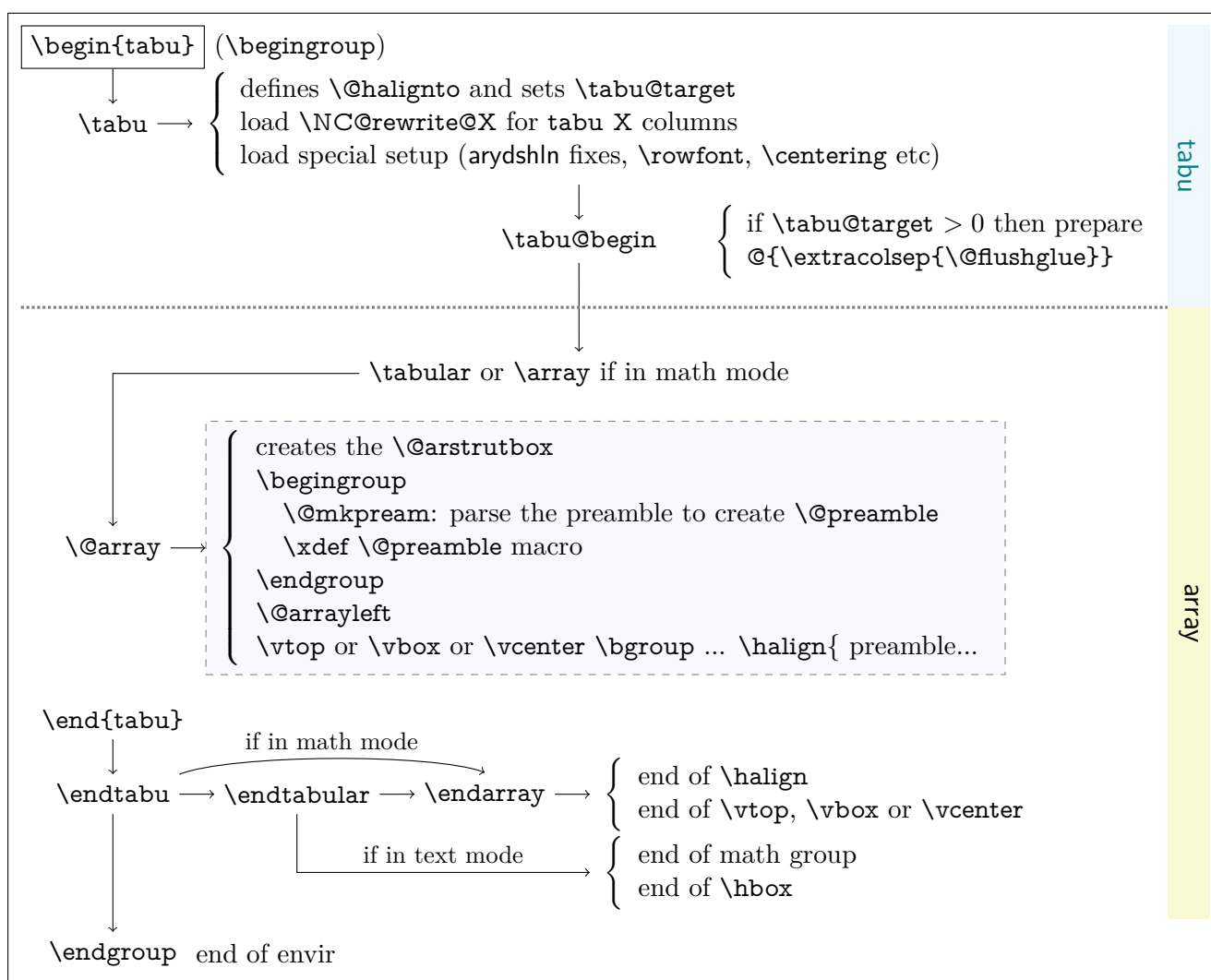
```

\tabu@Hy@xfootnote
336 \long\def\tabu@Hy@ftntext{\tabu@Hy@ftntxt {\the \c@footnote }}
337 \long\def\tabu@Hy@xfootnote [#1]{%
338   \begingroup
339     \value\@mpfn #1\relax
340     \protected@xdef \@thefnmark {\thempfn}%
341   \endgroup
342   \@footnotemark \tabu@Hy@ftntxt {#1}%
343 }% \tabu@Hy@xfootnote
344 \long\def\tabu@Hy@ftntxt #1#2{%
345   \edef\@tempa{%
346     \the\tabu@footnotes
347     \begingroup
348       \value\@mpfn #1\relax
349       \noexpand\protected@xdef
350         \noexpand\@thefnmark{\noexpand\thempfn}%
351       \expandafter\noexpand\expandafter\tabu@Hy@footnotetext
352       \expandafter{%
353         \Hy@footnote@currentHref
354       }%
355     }%
356     \global\tabu@footnotes\expandafter{%
357       \@tempa {#2}%
358     \endgroup
359   }%
360 }% \tabu@Hy@ftntxt
361 \long\def\tabu@Hy@footnotetext #1#2{%
362   \H@@footnotetext{%
363     \ifHy@nesting
364       \hyper@@anchor {#1}{#2}%
365     \else
366       \Hy@raisedlink{%
367         \hyper@@anchor {#1}{\relax}%
368       }%
369       \def\@currentHref {#1}%
370       \let\@currentlabelname \@empty
371       #2%
372     \fi
373   }%
374 }% \tabu@Hy@footnotetext
375 \AtBeginDocument{%
376   \csname\ifcsname ifHy@hyperfootnotes\endcsname
377     ifHy@hyperfootnotes\else iffalse\fi\endcsname
378     \let\tabu@footnotetext \tabu@Hy@ftntext
379     \let\tabu@xfootnote \tabu@Hy@xfootnote
380   \fi
381 }

```

8.5 Flow chart of expansion

General case: `tabu`, `tabu to` and `tabu spread`



`tabu to` with `X` column

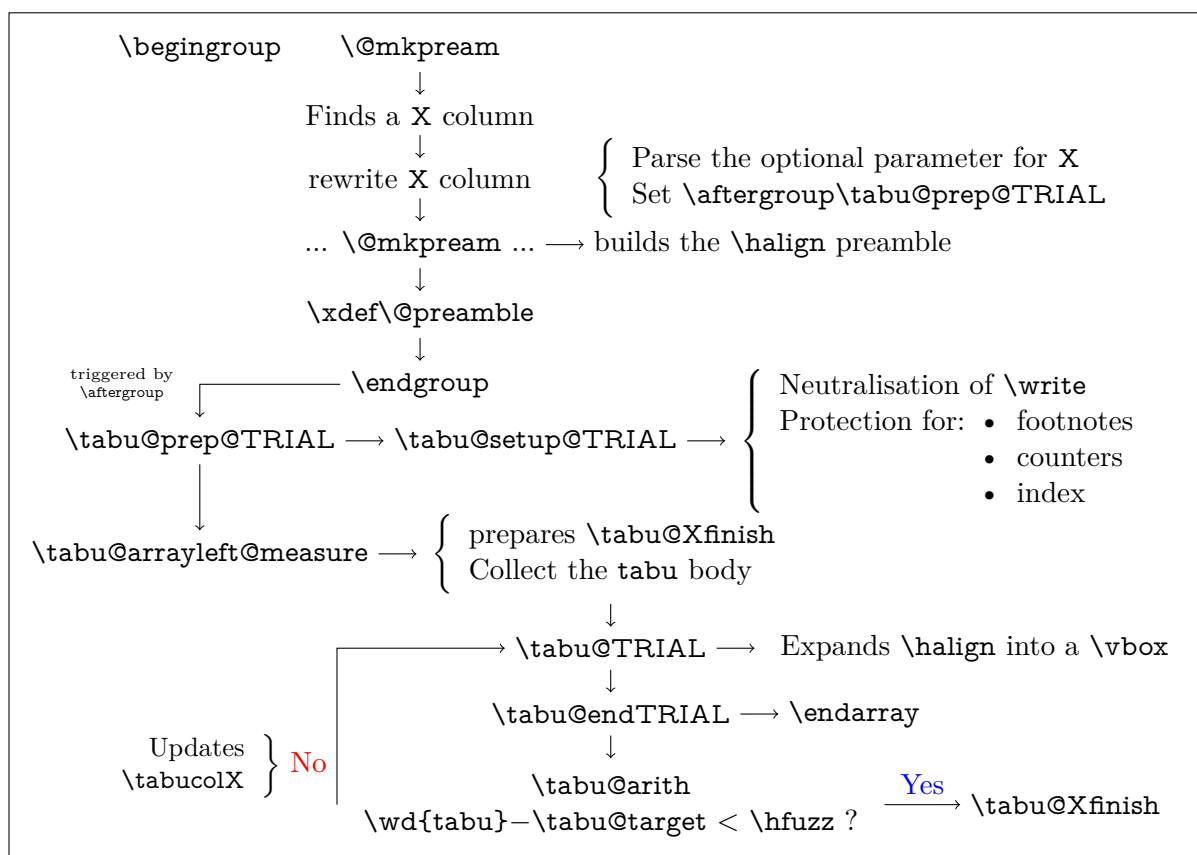
The important part of the job is made inside the dashed box above: `\@mkpreamble` expands the columns definitions, which can be user defined. Hopefully, it does its job inside a group, therefore a user-column can set a macro to be expanded `\aftergroup`. This implementation allows much modifications in the tabular preparation, without any change in the macros of `array.sty`.

When a `tabu X` column is found in the preamble by `\@mkpreamble`, `tabu` changes his strategy: the macro `\tabu@prep@TRIAL` is set to be expanded `\aftergroup` that is, just after the preamble (`\@preamble`) has been built. This macro does some setup for `tabu` trials to reach the target with variable `X` column widths and gobbles everything until the next `\bgroup` which corresponds to the `\vtop`, `\vbox` or `\vcenter` for the whole tabular. This part of `\@array` is stored into `\tabu@Xfinish` to be expanded after the last trial gave satisfaction to reach the `tabu` target. Then `\tabu@collect` is expanded to find the end of the `tabu` environment, temporarily storing the environment content into a token register.

The last part of `\@array` until `\halign` is expanded inside a `\vbox` which is stored into the box register `\tabu@box` for measuring purpose. `\halign` ends by `\endarray` which stops the `\tabu@box` as well, and then `\tabu@arith` is expanded to compute the gap between the width of `\tabu@box` and the target, and `\tabucolX` (the dimension that correspond to `X[1]`) is updated accordingly.

The trials are “protected” by `{\ifnum0=}\fi` : they occur in a group that will be closed at the very beginning of `\tabu@Xfinish`, when the final tabular will be printed actually. This protection is absolutely necessary to be able to collect the environment body in the case of nested `tabu` with `X` columns. This is related to T_EX mechanism of expansion inside `\halign` (T_EX stops

reading when it encounters a & alignment tab character and goes backward expanding anything that were not expanded before).



tabu spread with X column

In the case of “tabu spread” with X columns, the process is the same as the one described for “tabu to” with X columns. However, the first trial is different because we have first to measure the *natural width* of the tabular. The process is the following:

- `\tabu@target` is first set to `\linewidth` (or `\linegoal` with the `linegoal` package option).
- The X column corresponds to a `\vbox` with `\hsize` fixed to `\tabu@target`.
- Inside this `\vbox` the cell content is written into a `\hbox` whose width is limited to `\tabu@target`. This `\hbox` is captured into the box register `\tabu@box`.
- At the end of the cell, the `\badness` of the `\hbox` is checked:
 - if the `\badness` is > 1000 then the text is too long and “tabu spread” is useless: `tabu to \tabu@target` give the same result.
 - Otherwise, we get the natural width of the cell content by: `\setbox \tabu@box \hbox {\unhbox \tabu@box}`
- At the end of the first trial, `\tabu@spreadarith` checks if:

$$\text{width}(\text{tabular}) + \text{spread} < \text{\linewidth (or \linegoal)}$$

- if not, then `tabu to \tabu@target` give the same result
- Otherwise, the target for `tabu to` will be:

$$\text{width}(\text{tabular}) + \text{spread} - \underbrace{\sum_i \text{natural widths } X_i + \text{Max}_i \left(\frac{\text{natural width } X_i}{\text{coef}_i} \right) \times \sum_i \text{coef}_i}_{\text{minimal natural width that can be obtained with the given coefs}}$$

And the next trial will be done as if the user called “tabu to” with this target.

8.6 tabu X column definition

`\tabu@rewrite@X` This macro is expanded by `\@mkpream` in case a X column is found.

`\tabu@X@sum` (a dimen) store the sum of the width coefficients. For the first X column found in the preamble, a special setup occurs:

- the default target (either `\linewidth` or `\linegoal` if available) is set if it has not been specified by the user.
- `\@halignto` is `\let` to `\relax` to avoid its expansion in `\xdef\@preamble` just after `\@mkpream`. Indeed as long as we have to measure the natural width of the tabular, `\@halign` must be empty for trial steps.
- The rest of the setup is made `\aftergroup` (*ie.* after `\xdef\@preamble` which occurs inside a group) by `\tabu@prep@TRIAL`.

```

382 \def\tabu@rewrite@X #1{\tabu@Xarg {#1}%
383   \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \relax
384   \iftabu@measuring % not the first X column found in preamble
385     \edef\tabu@global@X {\tabu@global@X
386       \def\csname tabu@X\the\tabu@X@cols\endcsname {\strip@pt\tabu@X@sum}%
387       \advance\tabu@X@sum \the\tabu@X@sum\relax}%
388   \else % first X column found in preamble
389     \tabu@measuringtrue
390     \iftabu@spread \tabu@spreadtarget\tabu@target \tabu@target\z@
391     \else          \tabu@spreadtarget \z@ \fi
392     \ifdim\tabu@target=\z@
393       \setlength\tabu@target \tabu@defaulttarget
394       \tabu@message{(tabu) Default target:
395         \ifx\tabu@defaulttarget\linewidth\string\linewidth
396         \else\ifx\tabu@defaulttarget\linegoal\string\linegoal
397         \fi\fi=\the\tabu@target \on@line, page \the\c@page}\fi
398     \tabu@message{\tabu@titles}%
399     \edef\tabu@global@X {\tabu@global@X
400       \def\csname tabu@X1\endcsname{\strip@pt\tabu@X@sum}%
401       \tabu@X@sum \the\tabu@X@sum\relax
402       \tabu@target \the\tabu@target\relax
403       \tabu@spreadtarget \the\tabu@spreadtarget\relax}%
404     \let\tabu@halignto \relax
405     \fi
406 }% \tabu@rewrite@X
407 \let\tabu@@rewrite@X \tabu@rewrite@X

```

`\tabu@rewrite@Xnested` This macro replaces `\tabu@rewrite@X` when tabu makes a trial for X columns.

`\tabu@rewrite@Xrestore` This macro replaces `\tabu@rewrite@X` in the case of `\usetabu`.

```

408 \def\tabu@rewrite@Xnested #1{\in@ ${#1}\ifin@
409   \def\tabucolX@rw {>{$}p{\tabucolX}<{$}}\else
410   \def\tabucolX@rw {p{\tabucolX}}\fi}
411 \def\tabu@rewrite@Xrestore #1{\tabu@gobbleopt{\def\tabucolX@rw{1}}}}

```

`\tabu@Xarg` A tedious (and fastidious) macro to parse the optional argument of X columns. The aim is to built `\tabucolX@rw` which expands to the column specification:

`\tabu@Xparse` `\tabucolX@do` `>{alignment} p or m or b {\dimexpr coef \tabucolX\relax}`

After that `array.sty` make it easy: `\expandafter\NC@find\tabucolX@rw`

```

412 \def\tabu@Xarg#1{%
413   \advance\tabu@X@cols \@ne
414   \tabu@wdlet\tabu@X@cols \z@
415   \ifx\\#1\\% <shortcut when no option>
416     \tabu@X@sum \p@

```

```

417 \edef\tabucolX@rw {p{\tabu@dimexpr{\the\tabu@X@cols}{\tabucolX}}}%
418 \edef\tabu@temp {p{\the\tabucolX}}% <required for \usetabu>
419 \else
420 \tabu@X@sum \z@
421 \let\tabucolX@align \@empty \let\tabucolX@rw \@empty
422 \let\tabu@Xmath \@empty \let\tabu@temp \@empty
423 \tabu@Xparse {}#1,\@nnil
424 \fi
425 }% \tabu@Xarg
426 \def\tabu@Xparse #1{\futurelet\@let@token \tabu@Xtest}
427 \expandafter\def\expandafter\tabu@Xparsespace\space{\tabu@Xparse{}}
428 \def\tabu@Xtest{%
429 \ifcase \ifx \@nnil\@let@token \z@ \else
430 \if ,\@let@token \m@ne\else
431 \if p\@let@token 1\else
432 \if m\@let@token 2\else
433 \if b\@let@token 3\else
434 \if l\@let@token 4\else
435 \if c\@let@token 5\else
436 \if r\@let@token 6\else
437 \if L\@let@token 7\else
438 \if C\@let@token 8\else
439 \if R\@let@token 9\else
440 \ifx \@sptoken\@let@token 10\else
441 \if .\@let@token 11\else
442 \if -\@let@token 11\else
443 \ifcat $\@let@token 12\else
444 13\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\relax
445 \tabucolX@do \expandafter \remove@to@nnil
446 \or \tabu@Xtype {p}%
447 \or \tabu@Xtype {m}%
448 \or \tabu@Xtype {b}%
449 \or \tabu@Xalign \raggedright
450 \or \tabu@Xalign \centering
451 \or \tabu@Xalign \raggedleft
452 \or \tabu@Xalign \RaggedRight
453 \or \tabu@Xalign \Centering
454 \or \tabu@Xalign \RaggedLeft
455 \or \expandafter \tabu@Xparsespace
456 \or \expandafter \tabu@Xcoef
457 \or \let\tabu@Xmath=$\expandafter \tabu@Xparse
458 \or \tabu@Xcoef{}%
459 \else\expandafter \tabu@Xparse
460 \fi
461 }% \tabu@Xtest
462 \def\tabu@Xalign#1{%
463 \ifx \tabucolX@align\@empty \else \PackageWarning{tabu}
464 {Duplicate horizontal alignment specification}\fi
465 \def\tabucolX@align{#1}\expandafter\tabu@Xparse
466 }% \tabu@Xalign
467 \def\tabu@Xtype#1{%
468 \ifx \tabucolX@rw\@empty \else \PackageWarning{tabu}
469 {Duplicate vertical alignment specification}\fi
470 \def\tabucolX@rw{#1}\expandafter\tabu@Xparse
471 }% \tabu@Xtype
472 \def\tabu@Xcoef#1{\edef\tabu@temp{\tabu@temp#1}%
473 \afterassignment\tabu@Xc@ef \tabu@cnt\number\if-#10\fi
474 }% \tabu@Xcoef

```

```

475 \def\tabu@Xc@ef{%
476   \advance\tabu@X@sum \tabu@temp\the\tabu@cnt\p@
477   \tabu@Xparse{}%
478 }% \tabu@Xc@ef
479 \def\tabucolX@do{%
480   \ifx\tabucolX@rw\@empty \def\tabucolX@rw{p}\fi
481   \ifdim\tabu@X@sum<\z@ \tabu@negcoeftrue
482     \tabu@X@sum \m@ne\tabu@X@sum
483     \tabu@wdlet \tabu@X@cols \p@
484   \else \ifdim \tabu@X@sum=\z@ \tabu@X@sum \p@ \fi
485   \fi
486   \begingroup
487     \toks@{\expandafter {\tabucolX@align}}%
488     \edef\tabu@temp {\strip@pt\tabu@X@sum \tabucolX}%
489     \edef\tabucolX@rw##1{%
490       \ifx$\tabu@Xmath>{\noexpand\the\toks@ $}\else
491       \ifx \tabucolX@align\@empty \else
492         >{\noexpand\the\toks@}\fi\fi
493       \tabucolX@rw{##1}%
494       \ifx$\tabu@Xmath<{$}\fi}%
495     \edef\x{\endgroup
496       \def\noexpand\tabu@temp {\tabucolX@rw{\the\dimexpr\tabu@temp}}%
497       \def\noexpand\tabucolX@rw{%
498         \tabucolX@rw{\tabu@dimexpr{\the\tabu@X@cols}{\tabu@temp}}}%
499     }\x
500 }% \tabucolX@do
501 \def\tabu@dimexpr#1#2{%
502   \ifdim \tabu@wd{#1}>\p@
503     \ifdim \tabu@wd{#1}<#2 \tabu@wd{#1}\else#2\fi\else #2\fi
504 }% \tabu@dimexpr

```

8.7 Before trials to reach the target: setup after \@mkpream

`\tabu@prep@TRIAL` `\@mkpream` does its job inside a semi-simple group. At the end, `\@preamble` is expanded with `\xdef`, and the group is left: this triggers the expansion of `\tabu@prep@TRIAL` set `\aftergroup` by the first X column encountered in the preamble.

We `\let \@halignto` to `\@empty`: it's a measurement, and put some protections. A group is opened with the famous `{\ifnum0='}\fi` and control is given to `\tabu@arrayleft@measure`.

```

505 \def\tabu@prep@TRIAL{%
506   \begingroup \iffalse{\fi \ifnum0='}\fi
507   \tabu@setup@TRIAL
508   \tabucolX \tabu@target
509   \ifcase 0\iftabu@negcoef \else\iftabu@spread \else 1\fi\fi
510     \advance\tabu@X@cols \@ne
511     \expandafter\let\csname tabu@X\the\tabu@X@cols\endcsname \@undefined
512     \tabu@X@cols \z@
513     \def\tabu@naturalX@max {\z@}%
514     \def\tabu@naturalX@min {\z@}%
515     \let\tabu@startpboxORI \@startpbox
516     \let\@startpbox \tabu@measure@startpbox
517   \fi
518   \def\tabu@lasttry{\m@ne\p@}\tabu@cnt \z@ % number of trials
519   \tabu@arrayleft@measure
520 }% \tabu@prep@TRIAL

```

`\tabu@setup@TRIAL`

```

521 \def\tabu@setup@TRIAL{\tabu@global@X \let\tabu@global@X \@empty
522   \let\@halignto \@empty \tabu@naturalX \z@
523   \tabu@savecounters \let\tabu@savecounters \relax
524   \hbadness \@M \let\hbadness \@tempcnta
525   \hfuzz \maxdimen \let\hfuzz \@tempdima
526   \let\tabu@rewrite@X \tabu@rewrite@Xnested
527   \let\savetabu \@gobble \let\tabudefaulttarget \linewidth
528   \let\write \tabu@nowrite \let\GenericError \tabu@GenericError
529   \long\def\@xfootnote[##1]##2{}\let\@footnotetext \@gobble
530   \ifx\@arrayright\relax \let\@arrayright \@empty\fi
531   \tabu@TRIAL@hook
532 }% \tabu@setup@TRIAL
533 \def\tabu@savecounters{%
534   \def\@elt##1{\global\value{##1}\the\value{##1}\relax}%
535   \edef\tabu@global@X {\cl@ckpt}\let\@elt \relax
536 }% \tabu@savecounters

```

`\tabuDisableCommands`

```

537 \let\tabu@TRIAL@hook \@empty
538 \newcommand\tabuDisableCommands[1]{\g@addto@macro\tabu@TRIAL@hook {#1}}

```

8.8 One trial after another

`\tabu@arrayleftmeasure` Measuring the whole tabular occurs just before `\@arrayleft`. Hence the name of the macro: `\tabu@arrayleft@measure`.

`\tabu@TRIAL` `\halign` is temporarily expanded inside a `\vbox` which is captured in `\tabu@box`.

```

539 \def\tabu@arrayleftmeasure#1\bggroup{%
540   \def\tabu@Xfinish{%
541     \tabu@global@Xnested \ifnum0='{}\fi\endgroup
542     \tabu@global@X #1\bggroup}%
543   \def\tabu@TRIAL {\setbox\tabu@box \hbox\bgroup $#1\bgroup
544     \the\toks@ \tabu@endTRIAL}%
545   \toks@{\tabu@setup@everycr
546     \let\@preamble \tabu@savedpreamble}% <required for multicolumn>
547   \tabu@collect
548 }% \tabu@arrayleftmeasure

```

`\tabu@long@arrayleft@measure` This is the long version for `longtabu`

```

549 \long\def\tabu@long@arrayleft@measure #1\LT@bchunk #2\LT@bchunk{%
550   \def\tabu@Xfinish{%
551     \tabu@global@Xnested \ifnum0='{}\fi\endgroup
552     \tabu@setup@everycr \let\@preamble \tabu@savedpreamble
553     \tabu@global@X #1\LT@bchunk #2\LT@bchunk}%
554   \def\tabu@TRIAL{%
555     \setbox\tabu@box \hbox\bgroup
556     \tabu@setup@everycr \let\@preamble \tabu@savedpreamble
557     \tabu@global@X \LT@chunks\@M
558     #1\LT@bchunk #2\LT@bchunk \the\toks@ \tabu@long@endTRIAL}%
559   \toks@{}%\let\LT@output\relax
560   \tabu@collect
561 }% \tabu@long@arrayleft@measure
562 \def\tabu@long@endTRIAL{%
563   \LT@echunk
564   \global\setbox\@ne \hbox{\unhbox\@ne}\kern\wd\@ne
565   \LT@get@widths
566   \egroup % got \tabu@box

```

```

567 \iftabu@spread \tabu@spreadarith % <only once>
568 \else          \tabu@arith
569 \fi
570 \iftabu@measuring \tabu@measuringfalse
571 \expandafter \tabu@TRIAL % <continue trials>
572 \else
573 \expandafter \tabu@Xfinish
574 \the \expandafter\toks@ \tabu@endenvir % <we are then!>
575 \fi
576 }% \tabu@long@endTRIAL

```

`\tabu@collect` The mechanism is the same as \mathcal{AMS} -`\collect@body` (also defined in `environ.sty`). The content of the tabular is captured inside `\toks@`, expanded by `\tabu@TRIAL`.

```

577 \def\tabu@collect{\catcode'\^^@=13\def\tabu@stack{b}\tabu@collectbody}
578 \long\def\tabu@collectbody#1\end#2{%
579 \edef\tabu@stack{\tabu@pushbegins #1\begin\end\expandafter\@gobble\tabu@stack}%
580 \ifx\tabu@stack\@empty
581 \toks@\expandafter{\the\toks@#1}\def\tabu@endenvir{\end{#2}}%
582 \expandafter \tabu@TRIAL
583 \else
584 \toks@\expandafter{\the\toks@#1\end{#2}}%
585 \expandafter \tabu@collectbody
586 \fi
587 }% \tabu@collectbody
588 \long\def\tabu@pushbegins#1\begin#2{%
589 \ifx\end#2\else b\expandafter\tabu@pushbegins\fi}

```

`\tabu@endTRIAL` At the end of the trial, we call `\tabu@arith` to compute the widths. `\tabu@arith` exits leaving `\iftabu@measuring` equal to `\iftrue`: a further trial is necessary, or equal to `\iffalse`: the target is reached, `\tabu@Xfinish` can print the tabu in a last expansion of `\halign`.

```

590 \def\tabu@endTRIAL{%
591 \tabu@adl@endTRIAL % <arydshln in nested trials>
592 \endarray$\egroup
593 \iftabu@spread \tabu@spreadarith % <only once>
594 \else          \tabu@arith
595 \fi
596 \iftabu@measuring \tabu@measuringfalse
597 \expandafter \tabu@TRIAL % <continue trials>
598 \else
599 \expandafter \tabu@Xfinish
600 \the \expandafter\toks@ \tabu@endenvir % <we are then!>
601 \fi
602 }% \tabu@endTRIAL
603 \def\tabu@adl@endTRIAL{\crrc \noalign{\global\adl@ncol \tabu@nbcolls}}

```

8.9 Algorithms: Measuring the tabu

The arithmetic of X columns

`\tabu@arith` General algorithms for tabu to with X columns.

```

604 \def\tabu@arith{%
605 \advance\tabu@cnt \@ne \tabu@message{\tabu@message@arith}%
606 \ifx \@startpbox\tabu@measure@startpbox
607 \let\@startpbox \tabu@startpboxORI \fi
608 \dimen@ \tabu@X@sum
609 \iftabu@negcoef \tabu@arith@negcoef \fi
610 \ifcase 0\ifdim \dimexpr\wd\tabu@box-\tabu@hfuzz-1sp<\tabu@target
611 \else \ifdim \tabu@lasttry=\wd\tabu@box 1\else

```

```

612         2\fi\fi\relax
613         \ifdim \tabu@X@sum=\dimen@ \tabu@message{\tabu@message@reached}%
614         \else\ifdim\tabu@X@sum=\z@ \tabu@message{\tabu@message@reached}%
615         \else\ifnum \tabu@cnt>\@ne \tabu@message{\tabu@message@reached}%
616         \else \tabu@measuringtrue
617         \tabucolX = \dimexpr\tabucolX*\p@/\tabu@X@sum\relax
618         \fi\fi\fi
619     \or \tabu@message{\tabu@header Reached minimum width. Backing up}%
620         \tabu@warn{\tabu width = \the\wd\tabu@box\space>
621             \the\tabu@target\space+ \tabu@hfuzz}%
622     \or
623         \ifdim \tabu@X@sum=\z@ \tabu@measuringtrue \tabucolX=\@tempdima
624         \else
625             \advance\tabucolX-\dimexpr(\wd\tabu@box-\tabu@target-.5\tabu@X@sum/\p@)
626                                     *\p@/\tabu@X@sum\relax
627             \ifdim \tabucolX<\z@
628                 \tabu@warn{X Columns too narrow (table too wide)}%
629                 \tabucolX=\tabucolX@error
630             \else \tabu@measuringtrue
631                 \ifdim \tabu@X@sum=\dimen@
632                     \edef\tabu@lasttry{\the\wd\tabu@box}%
633                 \else
634                     \ifnum \tabu@cnt>\@ne \edef\tabu@lasttry{\the\wd\tabu@box}\fi
635                     \ifdim \tabucolX<\@tempdima \tabucolX \@tempdima\fi
636                 \fi
637             \fi
638         \fi
639     \fi %(ifcase)
640     \iftabu@measuring
641     \else \edef\tabu@global@X{\tabu@global@X
642         \tabucolX \the\tabucolX\relax
643         \tabu@target \the\tabu@target\relax % < for \savetabu >
644         \ifnum \tabu@nested=\z@
645             \ifx \@arrayright\@empty
646                 \ifdim \tabu@X@sum>\z@
647                     \def\noexpand\@halignto{to\the\tabu@target}\fi\fi\fi}%
648     \fi
649     \tabu@X@sum \dimen@
650 }% \tabu@arith
651 \def\tabu@arith@negcoef{%
652     \tabu@X@cols \@ne \count@ \tabu@X@cols
653     \@whilesw \ifcsname tabu@X\the\tabu@X@cols\endcsname\fi{%
654         \ifdim \tabu@wd\tabu@X@cols>\p@
655             \expandafter\let\expandafter\tabu@temp
656                 \csname tabu@X\the\tabu@X@cols\endcsname
657             \ifdim \tabu@wd\tabu@X@cols <\tabu@temp\tabucolX
658                 \advance\tabu@X@sum -\tabu@temp\p@
659                 \tabu@message{\tabu@message@negcoef{\the\tabu@X@cols}}%
660                 \ifdim \tabu@wd\tabu@X@cols >\tabu@wd\count@ \count@ \tabu@X@cols \fi
661             \fi\fi
662             \advance\tabu@X@cols \@ne
663         }%
664     \@tempdima=\dimexpr\tabu@wd\count@ * \p@/
665         \dimexpr\csname tabu@X\the\count@\endcsname\p@\relax\relax
666 }% \tabu@arith@negcoef

```

The arithmetic of X columns for tabu spread

```
\tabu@spreadarith Algorithm for tabu spread with X columns: only the first pass.

667 \def\tabu@spreadarith{%
668   \tabu@naturalX \tabu@naturalX@max\relax
669   \tabucolX \tabu@naturalX@min\relax
670   \tabu@message{\tabu@message@spread}%
671   \ifdim \dimexpr \wd\tabu@box + \tabu@spreadtarget <\tabu@target
672     \advance\tabu@spreadtarget
673     \dimexpr\wd\tabu@box-\tabu@naturalX+\tabucolX\relax
674   \ifdim \tabu@spreadtarget <\tabu@target
675     \tabu@target =\tabu@spreadtarget
676   \fi
677   \tabu@message{\the\tabu@target}%
678   \tabu@message{\ifdim \tabu@spreadtarget>\tabu@target
679     (tabu) spread reduced to fit default target.\fi}%
680   \else
681     \tabu@message{\expandafter\tabu@msg@align \the\tabu@target { }{}{}{}{}@@
682     \tabu@message{(tabu)\ifdim\wd\tabu@box<\tabu@target spread too large\else
683       tabu spread is useless here\fi,
684       default target used.}%
685   \fi
686   \let\@startpbox \tabu@startpboxORI
687   \xdef\tabu@halignto {to\the\tabu@target}%
688   \tabucolX \tabu@target
689   \tabu@spreadfalse \tabu@measuringtrue
690 }% \tabu@spreadarith
```

Reporting in the .log file (debugshow option)

```
\tabu@message@arith
\tabu@message@spread
\tabu@message@negcoef

691 \def\tabu@message@arith{\tabu@header
692   \expandafter\tabu@msg@align \the\tabucolX { }{ }{ }{ }{}@@
693   \expandafter\tabu@msg@align \the\wd\tabu@box { }{ }{ }{ }{}@@
694   \expandafter\tabu@msg@align \the\tabu@target { }{ }{ }{ }{}@@
695   \expandafter\tabu@msg@align \the\tabu@X@sum {}{}{}{}{}{}@@
696   \ifx\@startpbox\tabu@measure@startpbox\tabu@spaces (coefs<0)\else
697   \ifdim\tabu@X@sum>\z@
698     \expandafter\tabu@msg@align \the\dimexpr(\tabu@target-\wd\tabu@box)
699     *p@/\tabu@X@sum\relax {}{}{}{}{}{}@@\fi\fi^^J}%
700 \def\tabu@message@spread{\tabu@spreadheader
701   \expandafter\tabu@msg@align \the\tabu@spreadtarget {}{}{}{}{}{}@@
702   \expandafter\tabu@msg@align \the\wd\tabu@box {}{}{}{}{}{}@@
703   \expandafter\tabu@msg@align \the\tabucolX {}{}{}{}{}{}@@
704   \expandafter\tabu@msg@align \expandafter-\the\tabu@naturalX {}{}{}{}{}{}@@
705 \def\tabu@message@reached{\tabu@header
706   \expandafter\tabu@msg@align \the\tabucolX { }{ }{ }{ }{}@@
707   \expandafter\tabu@msg@align \the\wd\tabu@box {}{}{}{}{}{}@@
708   <\expandafter\tabu@msg@align \the\tabu@target { }{ }{ }{ }{}@@
709   + \tabu@hfuzz\space= hfuzz *****^^J%
710   (tabu)\tabu@spaces***** Reached Target:
711     hfuzz = \tabu@hfuzz\on@line\space *****^^J%
712     \ifnum\tabu@nested=\z@ \tabu@elapsedtime ^^J\fi}%
713 \def\tabu@message@etime{(tabu)\tabu@spaces
714   Time elapsed for trials:
715   \the\numexpr(\pdfelapsedtime-\tabu@starttime-32767)/65536\relax sec
716   \the\numexpr\numexpr(\pdfelapsedtime-\tabu@starttime)
717   -\numexpr(\pdfelapsedtime-\tabu@starttime-32767)/65536\relax*65536\relax
```

```

718      *1000/65536\relax ms}%
719 \def\tabu@message@negcoef #1{\tabu@spaces\tabu@spaces
720   * #1. X[-\csname tabu@X#1\endcsname]: natural width
721   = \tabu@wd{#1}
722   < \csname tabu@X#1\endcsname X
723   = \the\dimexpr\csname tabu@X#1\endcsname\tabu@colX
724 }% \tabu@message@negcoef
725 \edef\tabu@spaces{\@spaces}
726 \def\do#1{%
727   \def\tabu@msg@align##1.##2##3##4##5##6##7##8##9\@{
728     \ifnum##1<10 #1 #1\else
729     \ifnum##1<100 #1 \else
730     \ifnum##1<\@m #1\fi\fi\fi
731     ##1.##2##3##4##5##6##7##8##9\@}%
732   \def\tabu@header{(tabu) \ifnum\tabu@cnt<10 #1\fi\the\tabu@cnt) #1}%
733   \def\tabu@titles{\ifnum \tabu@nested=\z@
734     (tabu) Try#1 #1 tabu X #1 #1 #1tabu Width #1 #1 Target
735     #1 #1 #1 #1 Coefs #1 #1 #1 Update^^J\fi}%
736   \def\tabu@spreadheader{%
737     (tabu) Try#1 #1 Spread #1 #1 tabu Width #1 #1 Nat. Min. #1 #1 #1 Nat.X
738     #1 #1 #1 New Target^^J%
739     (tabu)spread}%
740   \def\tabu@tracing@save##1{\begingroup
741     \def\col@sep ####1\relax {(tabu) column sep #1 #1 #1 = ####1^^J}%
742     \def\arrayrulewidth ####1\relax{(tabu) arrayrulewidth #1 = ####1^^J}%
743     \def\doublerulesep ####1\relax {(tabu) doublerulesep #1 #1= ####1^^J}%
744     \def\extrarowheight ####1\relax{(tabu) extrarowheight #1 = ####1^^J}%
745     \def\extratabsurround ####1\relax{(tabu) extratabsurround = ####1^^J}%
746     \def\minrowclearance ####1\relax {(tabu) minrowclearance#1 = ####1^^J}%
747     \def\arraystretch ####1{(tabu) arraystretch #1 #1 = ####1^^J}%
748     \iftabu@colortbl
749     \def\CT@arc@####1{(tabu) arrayrulecolor #1 = ####1^^J}%
750     \def\CT@drsc@####1{(tabu) doublerulesepcolor=####1^^J}%
751     \let\color \relax
752     \fi
753     \def\NC@find ####1\@nil{(tabu) tabu preamble#1 #1 = \detokenize{####1}^^J}%
754     \def\tabu@wddef####1####2{(tabu) Natural width ####1 = ####2^^J}%
755     \let\edef \@gobbles\let\def \@empty
756     \tabu@message{%
757       (tabu) \string\savetabu{##1}: \on@line^^J%
758     \ifx \savetabu\tabusaveX
759       (tabu) X columns width #1 = \the\tabu@colX^^J\fi
760       (tabu) target #1 #1 #1 #1 #1 = \the\tabu@target^^J%
761       \tabu@sav@param
762       \tabu@sav@pream \@nil
763       \iftabu@negcoef \tabu@sav@wd\fi}%
764     \endgroup}%
765 } \do{ }

```

Measuring the natural width for tabu spread and negativ width-coefficients

`\tabu@measure@startpbox` When “`tabu spread`” is used with `X` columns, the first trial must measure the natural width of the columns. When `X` columns have negativ coefficient, the natural is computed after the target has been reached, with the absolute coefficients.

Nested trials may occur (`tabu spread` inside a `X` column with negativ coefficient for example). For the furthur trials, the standard scheme for `X` column is used: the natural width is measured only once.

pdfT_EX font expansion is disabled inside the `varwidth` environment (we set `\pdfadjustspacing` to 0).

```

766 \def\tabu@measure@startpbox#1{\bgroup
767   \let\@startpbox \tabu@startpboxORI           % in case of nesting
768   \in@{\tabu@dimexpr}{#1}%
769   \ifin@
770     \global\advance\tabu@X@cols \@ne
771     \ifcsname tabu@X\the\tabu@X@cols\endcsname\else
772       \global\tabu@X@cols \@ne
773       \global\tabu@naturalX \z@
774     \fi
775   \fi
776   \iftabu@spread\else
777     \ifin@ {\let\tabu@dimexpr \@firstoftwo\expandafter}%
778           \ifdim \tabu@wd{#1}<\p@ \in@false\fi\fi\fi
779     \ifin@
780       \beginngroup \let\tabu@dimexpr \@secondoftwo
781       \edef\x{\endgroup \def\noexpand\tabu@temp{\the\dimexpr#1}%
782       }\x
783       \setbox\tabu@box=\hbox\bgroup
784       \let\tabu@rewrite@X \tabu@@rewrite@X
785       \begin{varwidth}{\tabu@temp}\narrowragged\arraybackslash
786         \tabu@noadjustspacing \parfillskip\@flushglue
787       \bgroup \aftergroup\tabu@measure@endpbox
788     \else
789       \expandafter\@gobble\@startpbox{#1}% \@gobble \bgroup
790     \fi
791   }% \tabu@measure@startpbox
792 \ifdefined\pdfadjustspacing
793   \def\tabu@noadjustspacing{\pdfadjustspacing\z@}%
794 \else
795   \let\tabu@noadjustspacing\relax
796 \fi
797 \def\tabu@measure@endpbox{%
798   \@finalstrut\@arstrutbox\end{varwidth}\egroup      % <got my \tabu@box>
799   \global\advance\tabu@naturalX \wd\tabu@box
800   \ifdim \tabu@naturalX@max<\tabu@naturalX
801     \xdef\tabu@naturalX@max{\the\tabu@naturalX}\fi
802   \@tempdima=\dimexpr \wd\tabu@box * \tabu@X@sum /
803     \dimexpr\csname tabu@X\the\tabu@X@cols\endcsname\p@\relax\relax
804   \ifdim \tabu@naturalX@min<\@tempdima
805     \xdef\tabu@naturalX@min{\the\@tempdima}\fi
806   \ifdim \tabu@wd\tabu@X@cols>\z@
807   \ifdim \tabu@wd\tabu@X@cols<\wd\tabu@box
808     \tabu@wddef\tabu@X@cols {\the\wd\tabu@box}%
809     \tabu@debug{\tabu@spaces\tabu@spaces
810       * \the\tabu@X@cols. X[-\csname tabu@X\the\tabu@X@cols\endcsname]:
811       target=\the\dimexpr\tabu@temp\relax\space
812       \expandafter\expandafter\expandafter\string\tabu@wd\tabu@X@cols
813       =\tabu@wd\tabu@X@cols}%

```

```

814 \fi\fi
815 \box\tabu@box \egroup % end of \vtop (measure)
816 }% \tabu@measure@endpbox
817 \def\tabu@wd#1{\csname tabu@W\the\tabu@nested @\number#1\endcsname}
818 \def\tabu@wddef#1{\expandafter\xdef
819 \csname tabu@W\the\tabu@nested @\number#1\endcsname}
820 \def\tabu@wdlet#1{\global\expandafter\let
821 \csname tabu@W\the\tabu@nested @\number#1\endcsname}
822 \def\tabu@naturalX@min{\z@}
823 \def\tabu@naturalX@max{\z@}

```

8.10 \everycr setup and \tabuphantomline

`\tabu@setup@everycr` `\ialign` resets `\everycr` to an empty token. This macro sets `\everycr` for the `tabu` environment.

```

824 \def\tabu@setup@everycr{%
825 \tabu@firstcline true
826 \let\tabu@first@cline \z@
827 \def\tabu@restoreeverycr {\let\everycr \tabu@everycr}%
828 \def\everycr{\afterassignment\tabu@restoreeverycr \@temptokena}% <for ialign>
829 \tabu@everycr\expandafter{\the\tabu@everycr
830 \noalign{%
831 \ifcase\tabu@first@cline
832 \global\let\tabu@first@cline \tw@
833 \or\global\let\tabu@first@cline \tw@
834 \or\global\tabu@firstcline false
835 \global\let\tabu@first@cline \m@ne
836 \fi}%
837 \tabu@rowfontreset
838 }%
839 }% \tabu@setup@everycr

```

`\tabuphantomline` This macro inserts a phantom line in front of a `tabu`. This is necessary when you use `\usetabu` with `tabu X` column, with a single line containing `\multicolumn...`

```

840 \def\tabuphantomline{\crrc\noalign{%
841 \global\let\@arstrutbox \voidb@x
842 \iftabu@colortbl
843 \xdef\tabu@minrowclearance{\the\minrowclearance}%
844 \global\minrowclearance \z@
845 \fi
846 \global\let\vcenter \vbox
847 \ifnum \tabu@nested>\z@ \vspace*{\dimexpr\extrarowheight-\ht\@arstrutbox}\fi
848 \toks@{}\count@\@ne
849 \@whilenum\count@<\tabu@nbc\do{\advance\count@\@ne
850 \toks@\expandafter{\the\toks@&}}%
851 \toks@\expandafter{\the\toks@
852 \cr\noalign{%
853 \global\let\@arstrutbox \tabu@arstrutbox
854 \global\let\vcenter \tabu@vcenter
855 \iftabu@colortbl
856 \global\minrowclearance \tabu@minrowclearance\relax\fi
857 \global\let\tabu@first@cline \@ne}}}%
858 \expandafter}\the\toks@
859 }% \tabuphantomline

```

8.11 Lines inside tabu

Vertical lines

`\tabuvline@rewrite` | is defined as a new column type but only inside the group where `\@mkpream` does its job. This way, the modification is very local to `tabu`.

`\tabu@lines` This feature is disabled with the `light` option.

```
860 \def\tabu@lines{{\endlinechar\m@ne \scantokens{\def\x{}}}%
861   \expandafter}\expandafter\tabu@lines@ \x
862 }% \tabu@lines
863 \def\tabu@lines@#1{%
864   \tabu@newcolumnntype #1{\tabuvline@rewrite}%
865   \NC@list\expandafter{\the\NC@list \NC@do #1}%
866 }% \tabu@lines@
867 \newcommand*\tabuvline@rewrite[1][\]{\tabuvline@arg{#1}%
868   \expandafter \NC@find \tabu@temp}
869 \def\tabu@vline#1{\vrule width#1}
```

`\tabuvline@arg` A tedious (and fastidious) macro to parse the optional argument of | vertical lines...

```
870 \def\tabuvline@arg#1{%
871   \ifx\#1\% <shortcut when no option>
872     \edef\tabu@temp{!}%
873     \iftabu@colortbl\beginngroup\noexpand\CT@arc@ \fi
874     \noexpand\tabu@vline\arrayrulewidth
875     \iftabu@colortbl\endgroup\fi}%
876   \else \futurelet \tabu@temp \tabuvline@argi #1\p@\p@\@nnil \tabuvline@argiii
877   \fi
878 }% \tabuvline@arg
879 \def\tabuvline@argi{%
880   \let\tabu@color \@empty
881   \ifcat A\noexpand\tabu@temp
882     \@tempdima\arrayrulewidth
883     \expandafter\tabu@getlinecolor
884   \else \expandafter\tabuvline@argii
885   \fi
886 }% \tabuvline@argi
887 \def\tabuvline@argii#1\@nnil{%
888   \tabu@maybecolor \@tempdima #1 \@nnil
889 }% \tabuvline@argii
890 \def\tabuvline@argiii{%
891   \edef\tabu@temp{!\beginngroup
892     \ifx\tabu@color\@empty
893       \iftabu@colortbl \noexpand\CT@arc@ \fi
894     \else \tabu@color\fi
895     \noexpand\tabu@vline{\the\@tempdima}\endgroup}}%
896 }% \tabuvline@argiii
```

Horizontal lines: `\tabucline`

`\tabucline` `\tabucline[<style or spec.>]{start-end}`

`\tabucline` appears only at the end of a line: this is the place where we can insert a `\noalign` group. We built a new line to be inserted inside the `tabu`: this new line which contains the rule or leaders, is stored into `\toks@` (inside the `\noalign` group). When leaving the group `\toks@` is expanded.

```
897 \def\tabu@linedash{4pt}% <default value>
898 \def\tabu@dashgap{4pt}% <default value>
899 \def\tabucline{\noalign{\ifnum0='}\fi \@ifstar
```

```

900 {\@tempswatrue\tabu@ccline}
901 {\@tempswafalse\tabu@ccline}}%
902 \newcommand*\tabu@ccline[2][\toks@{}%
903 \tabu@startstop{#2}\tabu@getlinespec{#1}\@multicnt\@ne
904 \ifnum\tabu@start>\tabu@stop
905 \else\ifx\tabu@xleaders\relax
906 \else
907 \toks@{\noalign{\global\let\@arstrutbox \voidb@x}}%
908 %% \iftabu@firstccline\if\tabu@align t\vskip-\ht\@arstrutbox\fi\fi
909 \@whilenum\@multicnt<\tabu@start\do{\advance\@multicnt\@ne
910 \toks@\expandafter{\the\toks@ &}}%
911 \loop
912 \toks@\expandafter{\the\toks@ \omit }%
913 \ifnum \@multicnt=\@ne
914 \expandafter \tabu@add \csname tabu@tok@\the\@multicnt L\endcsname \@nil
915 \fi
916 \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
917 \toks@\expandafter{\the\toks@ \begingroup}%
918 \toks@\expandafter{\the\expandafter\toks@
919 \tabu@xleaders \endgroup}%
920 \fi
921 \ifcase 0\if@tempswa\else\ifnum \@multicnt<\tabu@stop 1\fi\fi\relax
922 \expandafter \tabu@add \csname tabu@tok@\the\@multicnt R\endcsname \@nil
923 \fi
924 \ifnum\@multicnt<\tabu@stop
925 \advance\@multicnt\@ne
926 \if@tempswa \toks@\expandafter{\the\toks@ &}%
927 \else \toks@\expandafter{\the\toks@ \span}\fi
928 \repeat
929 \@whilenum\@multicnt<\tabu@nbccls\do{\advance\@multicnt\@ne
930 \toks@\expandafter{\the\toks@ &}}%
931 \toks@\expandafter{\the\toks@ \cr
932 \noalign{\global\let\@arstrutbox \tabu@arstrutbox}}%
933 %% \tabu@debug{LIGNE AJOUTEE:\the\toks@}%
934 \fi\fi
935 \ifnum0='{ \fi\expandafter}\the\toks@
936 }% \tabu@ccline
937 \def\tabu@add #1\@nil{\toks@\expandafter{\the\toks@
938 \begingroup #1\endgroup\ifmmode$\fi}}

```

`\tabu@startstop` This macro parses the mandatory argument of `\tabucline`: start-column and end-column of the cline.

```

939 \def\tabu@startstop#1{\tabu@start@stop #1\relax 1-\tabu@nbccls\@nnil}
940 \def\tabu@start@stop #1-#2\@nnil{%
941 \@defaultunits \tabu@start \number0#1\relax\@nnil
942 \@defaultunits \tabu@stop \number0#2\relax\@nnil
943 \ifnum\tabu@start>\tabu@nbccls\tabu@start \tabu@nbccls\else
944 \ifnum\tabu@start<\@ne \tabu@start \@ne \fi\fi
945 \ifnum\tabu@stop>\tabu@nbccls \tabu@stop \tabu@nbccls \else
946 \ifnum\tabu@stop<\@ne \tabu@stop \tabu@nbccls\fi\fi
947 }% \tabu@start@stop

```

`\tabu@getlinespec` This macro parses the optional argument of `\tabucline` and check if it's a line specification (then `\tabu@getline` is expanded) or a `\leaders` specification (then `\tabu@leaders` is expanded).

```

948 \def\tabu@getlinespec#1{\let\tabu@xleaders \relax
949   \@defaultunits \let\@tempa=#1 \relax\@nnil
950   \ifx\@tempa\relax \let\tabu@xleaders \tabu@defaultleaders\else
951   \ifx\@tempa\hbox \tabu@defleaders{#1}\else
952   \ifx\@tempa\box \tabu@defleaders{#1}\else
953   \ifx\@tempa\copy \tabu@defleaders{#1}\else
954   \ifcsname tabu@line@style@\detokenize{#1}\endcsname
955     \csname tabu@line@style@\detokenize{#1}\endcsname
956   \else \tabu@getline{#1}\p@ on0pt off0pt}\fi\fi\fi\fi\fi
957 }% \tabu@getlinespec
958 \def\tabu@defleaders#1{%
959   \def\tabu@xleaders{\iftabu@colortbl\CT@arc@\fi
960     \xleaders\hbox{\lower.5\extrarowheight#1}\tabu@leaderfill}}
961 \def\tabucline@warn#1{\PackageWarning{tabu}
962   {Undefined line syle: #1
963   \MessageBreak Using default line style instead}}%
964   \let\tabu@xleaders \tabu@defaultleaders
965 }% \tabucline@warn

```

`\tabu@getline` This macro parses the optional argument of `\tabucline` (or the one of `\tabulinesyle`) and extract the thickness, the dash an gap specified. Default values assignments are done either.

```

966 \def\tabu@getline#1{\tabu@lineon #1 \@nil on\tabu@linedash \p@ \@nil\@nnil{#1}}
967 \def\tabu@maybecolor{\ifx\tabu@color\@empty \afterassignment\tabu@colortest
968   \else \@defaultunits\fi}
969 \def\tabu@lineon #1on#2\@nil#3\@nnil#4{%
970   \let\tabu@color \@empty
971   \@defaultunits \let\@tempa=#1 \relax\@nnil
972   \ifcase 0\ifx o\@tempa 1\else\ifcat A\noexpand\@tempa 2\fi\fi\relax
973     \tabu@maybecolor \@tempdima #1\arrayrulewidth \p@\@nnil
974     \tabu@maybecolor \@tempdimb #2\p@ \@nnil
975     \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil
976   \or\@tempdima \arrayrulewidth
977     \tabu@maybecolor \@tempdimb #2\p@ \@nnil
978     \tabu@lineoff #1 on#2 off\tabu@dashgap \p@ \@nnil
979   \else
980     \tabu@maybecolor \@tempdima \arrayrulewidth #1\p@\@nnil
981     \ifx\tabu@color\@empty \tabucline@warn\@tempa\else
982     \@tempdimb \z@
983     \tabu@lineoff off0pt \p@\@nnil \fi
984   \fi
985 }% \tabu@lineon
986 \def\tabu@lineoff #1off#2\@nnil{%
987   \tabu@maybecolor \@tempdimc #2\p@ \@nnil
988   \iftabu@colortbl \ifx \tabu@color\@empty
989     \def\tabu@color{\noexpand\CT@arc@}%
990   \fi\fi
991   \ifdim \@tempdimb=\z@
992     \ifdim \@tempdimc>\z@ \@tempdimb \tabu@dashgap\relax\fi\fi
993   \ifdim \@tempdimc=\z@
994     \ifdim \@tempdimb>\z@ \@tempdimc \tabu@linedash\relax\fi\fi
995   \ifdim \@tempdima<\z@ \else
996   \ifdim \@tempdimb<\z@ \else
997   \ifdim \@tempdimc<\z@ \else
998     \edef \tabu@xleaders{\tabu@color\xleaders
999     \ifdim \@tempdimc>\z@

```

```

1000      \hbox\bgroup \kern\the\dimexpr\@tempdimc/2\relax\fi
1001      \noexpand\iftabu@firstcline
1002      \vrule depth\dimexpr\the\@tempdima
1003          \ifdim\@tempdimb>\z@ width\the\@tempdimb\fi
1004      \noexpand\else
1005      \vrule height\dimexpr-\extrarowheight+\the\@tempdima
1006          depth\dimexpr\extrarowheight
1007          \ifdim\@tempdimb>\z@ width\the\@tempdimb\fi
1008      \noexpand\fi
1009      \ifdim\@tempdimc>\z@
1010          \kern\the\dimexpr\@tempdimc/2\egroup\fi
1011      \tabu@leaderfill}%
1012  \fi\fi\fi
1013 }% \tabu@lineoff

```

`\tabu@colortest`

```

1014 \def\tabu@colortest{\futurelet\tabu@temp\tabu@linecolor}
1015 \def\tabu@linecolor{%
1016   \ifcase 0\if ,\noexpand\tabu@temp\else
1017       \ifx\relax\tabu@temp\else
1018       \ifx \@sptoken\tabu@temp1\else
1019       \ifcat A\noexpand\tabu@temp2\else
1020       3\fi\fi\fi\fi\relax
1021   \def\tabu@next##1{\futurelet\tabu@temp\tabu@linecolor}%
1022   \or\def\tabu@next{\tabu@gobblespace{\futurelet\tabu@temp\tabu@linecolor}}%
1023   \or\let\tabu@next \tabu@getlinecolor
1024   \else\expandafter\remove@to@nnil
1025   \fi \tabu@next
1026 }% \tabu@linecolor
1027 \def\tabu@getlinecolor#1\p@{%
1028   \edef\@tempa{\zap@space #1 \@empty}%
1029   \ifcsname\string\color@\@tempa\endcsname
1030       \edef\tabu@color{\noexpand\noexpand\noexpand\color{\@tempa}}% \set@color
1031   \fi\remove@to@nnil
1032 }% \tabu@getlinecolor

```

`\tabulinestyle` `\tabulinestyle{style=spec.}`

```

1033 \def\tabulinestyle#1{\@for\@tempa:=#1\do
1034     {\expandafter\tabu@linestyle\@tempa==\@nil}}
1035 \def\tabu@linestyle#1=#2=#3\@nil{%
1036   \begingroup \tabu@getlinespec {#2}\expandafter\gdef
1037       \csname tabu@line@style@\detokenize{#1}\expandafter\endcsname
1038       \expandafter{\expandafter\def\expandafter\tabu@xleaders
1039           \expandafter{\tabu@xleaders}}}%
1040   \endgroup
1041 }% \tabu@linestyle
1042 \expandafter\def \csname tabu@line@style@\endcsname {%
1043     \let\tabu@xleaders \tabu@defaultleaders}%
1044 \def\tabu@defaultleaders{\iftabu@colortbl\CT@arc@\fi
1045     \leaders
1046     \iftabu@firstcline
1047         \vrule depth \arrayrulewidth
1048     \else
1049         \vrule height\dimexpr-\extrarowheight+\arrayrulewidth
1050         depth \extrarowheight
1051     \fi
1052     \tabu@leaderfill}
1053 \let\tabu@leaderfill \hfil

```

8.12 Verbatim inside tabu with X columns

`\tabu@sanitizetext`

```

1054 {\catcode32=13\relax\catcode'\^@=13\relax
1055 \gdef\tabu@verb{\@sanitize\catcode'\^=7\edef\string~\tabu@makeatletter
1056 \catcode32=13\let =\ \catcode'\^@=13\def^^@{\hskip\parfillskip\null\linebreak}%
1057 \endlinechar\m@ne}}
1058 {\catcode64=13\edef @{\string @}\expandafter\xdef
1059 \csname tabu@makeatletter\endcsname{\catcode64=13\def\noexpand @{\unskip @}}
1060 \newcommand\tabu@sanitizetext[1][\ttfamily]{\begingroup
1061 \tabu@verb #1\tabu@sanitizetext}
1062 \long\def\tabu@s@nitizetext#1{\@makeother\{\@makeother\}%
1063 \everyeof{\noexpand}\scantokens{#1}\endgroup}

```

8.13 Numbers in tabu

`\tabudecimal`

`\tabudecimal` `\tabu@tabudecimal` is `\tabudecimal` inside `tabu`.

```

1064 \def\tabu@tabudecimal#1{%
1065 \def\tabu@decimal{#1}\@temptokena{}%
1066 \let\tabu@getdecimal@ \tabu@getdecimal@ignorespaces
1067 \tabu@scandecimal
1068 }% \tabu@tabudecimal
1069 \def\tabu@scandecimal{\futurelet \tabu@temp \tabu@getdecimal@}
1070 \def\tabu@skipdecimal#1{#1\tabu@scandecimal}
1071 \def\tabu@getdecimal@ignorespaces{%
1072 \ifcase 0\ifx\tabu@temp\ignorespaces\else
1073 \ifx\tabu@temp\@sptoken1\else
1074 2\fi\fi\relax
1075 \let\tabu@getdecimal@ \tabu@getdecimal
1076 \expandafter\tabu@skipdecimal
1077 \or \expandafter\tabu@gobblespace\expandafter\tabu@scandecimal
1078 \else \expandafter\tabu@skipdecimal
1079 \fi
1080 }% \tabu@getdecimal@ignorespaces
1081 \def\tabu@get@decimal#1{\@temptokena\expandafter{\the\@temptokena #1}%
1082 \tabu@scandecimal}
1083 \def\do#1{%
1084 \def\tabu@get@decimalspace#1{%
1085 \@temptokena\expandafter{\the\@temptokena #1}\tabu@scandecimal}%
1086 }\do{ }
1087 \let\tabu@@tabudecimal \tabu@tabudecimal

```

`\tabu@getdecimal`

```

1088 \def\tabu@getdecimal{%
1089 \ifcase 0\ifx 0\tabu@temp\else
1090 \ifx 1\tabu@temp\else
1091 \ifx 2\tabu@temp\else
1092 \ifx 3\tabu@temp\else
1093 \ifx 4\tabu@temp\else
1094 \ifx 5\tabu@temp\else
1095 \ifx 6\tabu@temp\else
1096 \ifx 7\tabu@temp\else
1097 \ifx 8\tabu@temp\else
1098 \ifx 9\tabu@temp\else
1099 \ifx .\tabu@temp\else
1100 \ifx ,\tabu@temp\else

```

```

1101             \ifx -\tabu@temp\else
1102             \ifx +\tabu@temp\else
1103             \ifx e\tabu@temp\else
1104             \ifx E\tabu@temp\else
1105             \ifx\tabu@celleft\tabu@temp1\else
1106             \ifx\ignorespaces\tabu@temp1\else
1107             \ifx\@sptoken\tabu@temp2\else
1108             3\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\relax
1109     \expandafter\tabu@get@decimal
1110 \or \expandafter\tabu@skipdecimal
1111 \or \expandafter\tabu@get@decimalspace
1112 \else\expandafter\tabu@printdecimal
1113 \fi
1114 }% \tabu@getdecimal
1115 \def\tabu@printdecimal{%
1116   \edef\tabu@temp{\the\@temptokena}%
1117   \ifx\tabu@temp\@empty\else
1118   \ifx\tabu@temp\space\else
1119     \expandafter\tabu@decimal\expandafter{\the\@temptokena}%
1120   \fi\fi
1121 }% \tabu@printdecimal

```

8.14 \savetabu

\savetabu When this command is called by the user, the tabu preamble and target are globally stored into a macro `\tabu@saved@{user-name}`.

```

1122 \newcommand*\savetabu[1]{\noalign{%
1123   \ifx\#1\\ \tabu@savewarn{}\else
1124     \@ifundefined{tabu@saved@\string#1}{}{\tabu@savewarn{#1}{Overwriting}}%
1125   {%
1126     \toks0\expandafter{\tabu@savedpreamble}%
1127     \toks1\expandafter{\tabu@savedparam}%
1128     \toks2\expandafter{\tabu@savedpream}%
1129     \toks3{}}%
1130   \iftabu@negcoef
1131     \edef\tabu@savewd{\tabu@savewd{\number\numexpr1}}%
1132     \toks3\expandafter{\tabu@savewd}\fi
1133   \expandafter\xdef\csname tabu@saved@\string#1\endcsname##1{%
1134     \noexpand\ifcase##1\relax
1135       \noexpand\ifdim\tabu@target>\z@
1136         \noexpand\tabu@warn@usetabu
1137       \noexpand\fi
1138       \tabu@target \the\tabu@target\relax
1139       \tabucolX \the\tabucolX\relax
1140       \the\toks1%
1141       \gdef\noexpand\tabu@usetabu{%
1142         \the\toks1%
1143         \def\noexpand\@halignto{\tabu@halignto}%
1144         \tabucolX \the\tabucolX\relax
1145         \def\noexpand\@preamble{\the\toks0}%
1146         \noexpand\tabu@make@arstrutbox
1147         \the\toks3
1148         \noexpand\ifx \noexpand\tabu@align\noexpand\tabu@aligndefault@text
1149           \def\noexpand\tabu@align{\tabu@align}%
1150         \noexpand\fi}%
1151       \aftergroup\noexpand\tabu@usetabu
1152     \noexpand\else
1153       \noexpand\ifdim\tabu@target=\z@

```

```

1154         \tabu@target \the\tabu@target\relax
1155         \noexpand\fi
1156         \gdef\noexpand\tabu@preamble{%
1157             \noexpand\ifx \noexpand\tabu@align\noexpand\tabu@aligndefault@text
1158                 \def\noexpand\tabu@align{\tabu@align}%
1159             \noexpand\fi}%
1160         \aftergroup\noexpand\tabu@preamble
1161         \noexpand\fi
1162         \the\toks2}%
1163     \tabu@message@save{#1}}%
1164 \fi}%
1165 }% \savetabu
1166 \def\tabu@aligndefault@text{\tabu@aligndefault}%
1167 \def\tabu@warn@usetabu{\PackageWarning\tabu{
1168     {Specifying a target with \string\usetabu\space is useless
1169     \MessageBreak The target cannot be changed!}}
1170 \def\tabu@savewd#1{%
1171     \ifnum#1>\tabu@nbcolls \expandafter \@gobble
1172     \else \ifdim\tabu@wd{#1}>\z@
1173         \noexpand\tabu@wddef{#1}{\tabu@wd{#1}}\fi
1174         \expandafter\tabu@savewd % loop
1175     \fi{#1+1}}%
1176 }% \tabu@savewd
1177 \def\tabu@make@arstrutbox{%
1178     \@tempdima \ht \strutbox
1179     \advance \@tempdima by\extrarowheight
1180     \setbox \@arstrutbox \hbox{\vrule
1181         \@height \arraystretch \@tempdima
1182         \@depth \arraystretch \dp \strutbox
1183         \@width \z@}%
1184 }% \tabu@make@arstrutbox

```

`\tabu@savewarn` Info for overwriting when `\savetabu` is used.

`\tabu@saveerr` Error if `\usetabu` is called with an unknown argument.

```

1185 \def\tabu@savewarn#1#2{\PackageInfo\tabu{
1186     {User-name ‘#1’ already used for \string\savetabu
1187     \MessageBreak #2}}%
1188 \def\tabu@saveerr#1{\PackageError\tabu{
1189     {User-name ‘#1’ is unknown for \string\usetabu
1190     \MessageBreak I cannot restore an unknown preamble!}\@ehd}

```

8.15 `\rowfont`

Setting font and alignment specification

`\rowfont` `\rowfont` uses the control sequences `\tabu@celllalign`, `\tabu@celleft`, `\tabu@cellright` and `\tabu@cellralign` which have been placed on purpose into the user-defined tokens inserted in any preamble by the `array` package.

`\tabu@celllalign` and `\tabu@cellralign` are used to modify the alignment. If the optional [*alignment*] parameter of `\rowfont` is not specified, then those control sequence expand to `\@empty`.

`\tabu@celleft` contains the font-modification information.

Placement of those control sequences into the user-tokens that are inserted in the preamble by the `array` package is explained below under the macro `\tabu@prepnext@tok`.

```

1191 \def\tabu@rowfont{\noalign{\ifnum0=‘}\fi\tabu@row@font}
1192 \newcommand*\tabu@row@font[2][]{%

```

```

1193 \global\let\tabu@rowfontreset \tabu@rowfont@reset
1194 \global\let\tabu@cellleft \tabu@cellleft
1195 \global\let\tabu@cellright \tabu@cellright
1196 \gdef\tabu@cellfont{\let\color \tabu@colorleavevmode#2\let\color \tabu@colorORI}%
1197 \ifcsname tabu@cell@#1\endcsname % row alignment
1198 \csname tabu@cell@#1\endcsname \fi
1199 \toks\@expandafter {\tabu@cellleft\tabu@cellfont}% inside \noalign group ok
1200 \xdef\tabu@cellleft {\the\toks}%
1201 \ifnum0='{ \fi}% end of noalign group
1202 }% \rowfont
1203 \def\tabu@colorleavevmode{\leavevmode\tabu@colorORI}
1204 \AtBeginDocument{\let\tabu@colorORI \color}

```

`\tabu@rowfont@reset` This macro resets `\tabu@cellalign`, `\tabu@cellleft`, `\tabu@cellright`, `\tabu@cellralign` and `\everycr` to the value they had before the expansion of `\rowfont`.

It expands when a new row is inserted into the tabular or array.

```

1205 \def\tabu@rowfont@reset{%
1206   \noalign{%
1207     \global\let\tabu@rowfontreset \@empty
1208     \global\let\tabu@cellleft \tabu@cellleft
1209     \global\let\tabu@cellright \tabu@cellright
1210     \global\let\tabu@cellfont \@empty
1211     \global\let\tabu@celllalign \@empty
1212     \global\let\tabu@cellralign \@empty
1213   }%
1214 }% \tabu@rowfont@reset
1215 \let\tabu@rowfontreset \@empty

```

Preparing stuff to be able to use `\rowfont`

`\tabu@prepnext@tok` `\tabu@prepnext@tok` will replace `\prepnext@tok` (in `array.sty`): its purpose is to add the control sequences `\tabu@cellalign`, `\tabu@cellleft`, `\tabu@cellright` and `\tabu@cellralign` at the right position in the “preamble” for `\halign`. Those control sequences are not inserted directly into the preamble, but by the means of the user-tokens placed there by the `array` package.

The package `array` defines a macro `\prenext@tok` to initialize each user-token inserted at both side of each “normal” column. For “special” `@` and `!` columns, there is only one token.

$$\underbrace{>\{\backslash\bfseries\color{red}\}}_{\backslash\text{toks}<i>} \text{ r } \underbrace{<\{\backslash\color{black}\},\backslash\$}_{\backslash\text{toks}<i+1>}$$

When a column is inserted in the tabular preamble (`\@preamble`), the T_EX counter `\count@` is equal to $i + 1$ (*ie.* the right token) and the counter `\@tempcnta` is equal to i (*ie.* the left token). If the column is special (*ie.* `@` or `!`) `\@tempcnta` is not updated.

Thus, when a new token is “prepared” by `\prepnext@tok`:

either: $i = \text{count@} = \text{@tempcnta}$: the token to prepare (*ie.* $\text{toks}<i+1>$) is the right one of a “normal” column. The switch `\iftabu@cellright` is set to `true`.

The *previous* token ($\text{toks}<i> = \text{toks}\text{count@}$) is necessarily the left one of this “normal” column: we prepend `\tabu@cellalign` and append `\tabu@cellleft` to this token ($\text{toks}<i>$). This token is finished and will not change afterwards.

or: $i = \text{count@} = \text{@tempcnta} + 1$: the token to prepare ($\text{toks}<i+1>$) is either the left one of a normal column, or the single one of a special `@` or `!` column.

If the switch `\iftabu@cellright` is `true`, then the *previous* token $\text{toks}<i>$ is the right one of the last inserted column (which was a “normal” column, thus);, `\tabu@cellright\tabu@cellralign` is appended to it, and the switch `\ittabu@cellright` is reset to `false`. May be `\prepnext@tok` will be expanded again (by `\save@decl`): if it happens, then again $\text{count@} = \text{@tempcnta} + 1$ (same case) but `\iftabu@cellright` is `false` and nothing is changed.

else: The token to prepare (which is `\toks< i + 1 >=\toks\count@+1`), cannot be the right one of a “normal” column: `\iftabu@cellright` is set to false.
 The fact that `|\count@-\@tempcnta| > 1` tells us that the previous token `\toks< i >` is necessarily the single one of a “special” `@` or `!` column. We don’t modify this token, as long as *special columns are always inserted as is*: `\rowcolor` has no effect on special columns, nor `\rowfont`.

Thereafter, the original initialisation sequence occurs: `\advance\count@` by `\@one` and initialize the token to prepare (`\toks\count@=\toks< i + 1 >`) to an empty one.

```

1216 \newif\iftabu@cellright
1217 \AtBeginDocument{\let\tabu@prepnext@tokORI \prepnext@tok}% original definition
1218 \def\tabu@prepnext@tok{%
1219   \ifnum \count@<\z@ % <first initialisation>
1220     \@tempcnta \@M % <not initialized by array.sty>
1221     \tabu@nbcoks\z@
1222     \xdef\tabu@global@temp{\tabu@global@temp \tabu@nbcoks\z@}%
1223     \expandafter\let\csname tabu@tok@1\endcsname \relax
1224     \tabu@cellrightfalse
1225   \else
1226     \ifcase \numexpr \count@-\@tempcnta \relax % (case 0): prev. token is left
1227       \advance \tabu@nbcoks\@one
1228       \expandafter\let\csname tabu@tok@\the\tabu@nbcoks R\endcsname \relax
1229       \expandafter\gdef\expandafter\tabu@global@temp\expandafter{%
1230         \tabu@global@temp \advance\tabu@nbcoks\@one}%
1231       \iftabu@cellright % before-previous token is right and is finished
1232         \tabu@cellrightfalse % <only once>
1233         \tabu@savetok R\tabu@preptokenright
1234       \fi
1235       \ifnum \tabu@nbcoks=\@one \tabu@savetok L\fi
1236       \tabu@preptokenleft
1237     \or % (case 1) previous token is right
1238       \tabu@savetok R\tabu@cellrighttrue
1239     \else % special column: do not change the token
1240       \ifnum \tabu@nbcoks>\z@ %special column: always on the right of normal one
1241         \tabu@savetok R%
1242       \else % unless this is the very first column (\tabu@nbcoks=0)
1243         \advance\tabu@nbcoks\@one \tabu@savetok L\advance\tabu@nbcoks\m@ne
1244       \fi
1245       \iftabu@cellright % before-previous token is right
1246         \tabu@cellrightfalse
1247         \tabu@preptokenright
1248       \fi
1249     \fi % \ifcase
1250   \fi
1251   \tabu@prepnext@tokORI
1252 }% \tabu@prepnext@tok
1253 \def\tabu@preptokenright{%
1254   \advance \count@ \m@ne
1255   \toks\count@\expandafter {\the\toks\count@ \tabu@cellright \tabu@celllalign}%
1256   \advance \count@ \@one
1257 }% \tabu@preptokenright
1258 \def\tabu@preptokenleft{\toks\count@\expandafter {\expandafter\tabu@celllalign
1259   \the\toks\count@ \tabu@cellleft}%
1260 }% \tabu@preptokenleft
1261 \def\tabu@savetok#1{\begingroup
1262   \expandafter\tabu@savetok\csname tabu@tok@\the\tabu@nbcoks #1\endcsname
1263 }% \tabu@savetok
1264 \def\tabu@savetok#1{%

```

```

1265 \ifnum \@chclass=\@ne \toks@\toks\count@
1266 \else\ifnum \@chclass=5 \toks@\toks\count@
1267 \else \toks@{}\fi\fi
1268 \unless\ifx#1\relax
1269 \toks@\expandafter\expandafter\expandafter{%
1270 \expandafter#1\the\toks@}\fi
1271 \toks@\expandafter{\expandafter\def\expandafter#1\expandafter{%
1272 \the\toks@}}}%
1273 \@temptokena\expandafter{\tabu@global@temp}%
1274 \xdef\tabu@global@temp {\the\@temptokena \the\toks@}%
1275 \expandafter\endgroup \the\toks@
1276 }% \tabu@savetok

```

Neutralisation of glues and alignment modification

<code>\tabu@cellleft</code> <code>\tabu@celllalign</code> <code>\tabu@cellright</code> <code>\tabu@cellralign</code> <code>\tabu@cellfont</code>	<p>First initialisation to \@empty.</p> <pre> 1277 \let\tabu@cellleft\@empty 1278 \let\tabu@cellright\@empty 1279 \def\tabu@celllalign{\tabu@cellleft}% row font spec. applies to pre-column material 1280 \let\tabu@cellralign\@empty 1281 \let\tabu@cellfont\@empty </pre>
<code>\tabu@cell@l</code> <code>\tabu@cell@c</code> <code>\tabu@cell@r</code> <code>\tabu@cell@j</code>	<p>Setup macros to modify the alignment. The skips inserted to make the standard alignment specified in the tabular preamble are not the same with standard array tabulars and colortbl tabulars, hence the switch \iftabu@colortbl.</p> <pre> 1282 \def\tabu@cell@l{% force alignment to left 1283 \gdef\tabu@celllalign{\tabu@removehfil 1284 \raggedright\arraybackslash 1285 \tabu@cellleft}% 1286 \gdef\tabu@cellralign{\tabu@flush1\tabu@ignorehfil}% 1287 \toks@\expandafter{\tabu@cellleft\raggedright\arraybackslash}% local 1288 \xdef\tabu@cellleft{\the\toks@}% 1289 }% \tabu@cell@l 1290 \def\tabu@cell@c{% force alignment to center 1291 \gdef\tabu@celllalign{\tabu@removehfil 1292 \centering\arraybackslash 1293 \tabu@flush{.5}\tabu@cellleft}% 1294 \gdef\tabu@cellralign{\tabu@flush{.5}\tabu@ignorehfil}% 1295 \toks@\expandafter{\tabu@cellleft\centering\arraybackslash}% local 1296 \xdef\tabu@cellleft{\the\toks@}% 1297 }% \tabu@cell@c 1298 \def\tabu@cell@r{% force alignment to right 1299 \gdef\tabu@celllalign{\tabu@removehfil 1300 \raggedleft\arraybackslash 1301 \tabu@flush1\tabu@cellleft}% 1302 \gdef\tabu@cellralign{\tabu@ignorehfil}% 1303 \toks@\expandafter{\tabu@cellleft\raggedleft\arraybackslash}% local 1304 \xdef\tabu@cellleft{\the\toks@}% 1305 }% \tabu@cell@r 1306 \def\tabu@cell@j{% force justification (for p, m, b columns) 1307 \gdef\tabu@celllalign{\tabu@justify\tabu@cellleft}% 1308 \global\let\tabu@cellralign\@empty 1309 \toks@\expandafter{\tabu@cellleft\tabu@justify}% local (noalign grp) 1310 \xdef\tabu@cellleft{\the\toks@}% 1311 }% \tabu@cell@j 1312 \def\tabu@justify{% 1313 \leftskip\z@skip \@rightskip\leftskip \rightskip\@rightskip </pre>

```

1314 \parfillskip\@flushglue
1315 }% \tabu@justify
1316 %% ragged2e settings
1317 \def\tabu@cell@L{% force alignment to left (ragged2e)
1318 \gdef\tabu@celllalign{\tabu@removehfil
1319 \RaggedRight\arraybackslash
1320 \tabu@cellleft}%
1321 \gdef\tabu@celllralign{\tabu@flush1\tabu@ignorehfil}%
1322 \toks@{\expandafter{\tabu@cellleft\RaggedRight\arraybackslash}}%
1323 \xdef\tabu@cellleft{\the\toks@}%
1324 }% \tabu@cell@L
1325 \def\tabu@cell@C{% force alignment to center (ragged2e)
1326 \gdef\tabu@celllalign{\tabu@removehfil
1327 \Centering\arraybackslash
1328 \tabu@flush{.5}\tabu@cellleft}%
1329 \gdef\tabu@celllralign{\tabu@flush{.5}\tabu@ignorehfil}%
1330 \toks@{\expandafter{\tabu@cellleft\Centering\arraybackslash}}%
1331 \xdef\tabu@cellleft{\the\toks@}%
1332 }% \tabu@cell@C
1333 \def\tabu@cell@R{% force alignment to right (ragged2e)
1334 \gdef\tabu@celllalign{\tabu@removehfil
1335 \RaggedLeft\arraybackslash
1336 \tabu@flush1\tabu@cellleft}%
1337 \gdef\tabu@celllralign{\tabu@ignorehfil}%
1338 \toks@{\expandafter{\tabu@cellleft\RaggedLeft\arraybackslash}}%
1339 \xdef\tabu@cellleft{\the\toks@}%
1340 }% \tabu@cell@R
1341 \def\tabu@cell@J{% force justification (ragged2e)
1342 \gdef\tabu@celllalign{\justifying\arraybackslash\tabu@cellleft}%
1343 \global\let\tabu@celllralign\@empty
1344 \toks@{\expandafter{\tabu@cellleft\justifying\arraybackslash}}%
1345 \xdef\tabu@cellleft{\the\toks@}%
1346 }% \tabu@cell@J
1347 \def\tabu@flush#1{%
1348 \iftabu@colortbl % colortbl uses \hfill rather than \hfil
1349 \hskip \ifnum\currentgrouptype>13 \stretch{#1}%
1350 \else\ifdim#1pt<1pt \tabu@cellskip
1351 \else \stretch{#1}
1352 \fi\fi \relax
1353 \else % array.sty
1354 \ifnum \currentgrouptype>13\relax
1355 \hfil \hskip1sp
1356 \fi
1357 \fi
1358 }% \tabu@flush
1359 \AtBeginDocument{%
1360 \@ifpackageloaded{ragged2e}
1361 {}
1362 {\let\tabu@cell@L \tabu@cell@l
1363 \let\tabu@cell@R \tabu@cell@r
1364 \let\tabu@cell@C \tabu@cell@c
1365 \let\tabu@cell@J \tabu@cell@j
1366 }%
1367 }% \AtBeginDocument

```

`\tabu@removehfil` `\tabu@removehfil` removes (eventually) the infinite stretchable glue inserted *before* the cell (in the preamble of `\halign`) to make the column alignment.

```

1368 \newskip\tabu@cellskip
1369 \let\tabu@hfil\hfil
1370 \let\tabu@hfill\hfill
1371 \let\tabu@hskip\hskip
1372 \def\tabu@removehfil{%
1373   \iftabu@colortbl
1374     \unkern \tabu@cellskip = \lastskip
1375     \ifnum\gluestretchorder\tabu@cellskip = \tw@ \hskip-\tabu@cellskip
1376     \else \tabu@cellskip = \z@skip
1377     \fi
1378   \else
1379     \ifdim\lastskip=1sp\unskip\fi
1380     \ifnum\gluestretchorder\lastskip = \@ne
1381       \hfilneg % \hfilneg for array.sty but not for colortbl...
1382     \fi
1383   \fi
1384 }% \tabu@removehfil

\tabu@ignorehfil \tabu@ignorehfil removes (eventually) the infinite stretchable glue inserted after the cell (in the preamble of \halign) to make the column alignment.

1385 \def\tabu@ignorehfil{%
1386   \aftergroup\tabu@nohfil
1387 }% \tabu@ignorehfil
1388 \def\tabu@nohfil{% \hfil -> do nothing + restore original \hfil
1389   \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1390 }% \tabu@nohfil
1391 \AtBeginDocument{%
1392   \@ifpackageloaded{colortbl}
1393   {%
1394     \def\tabu@nohfil{%
1395       \def\hfil{\let\hfil\tabu@hfil}% local to (alignment template) group
1396       \def\hfill{\let\hfill\tabu@hfill}% (colortbl uses \hfill) pfff...
1397       \def\hskip##1\relax{\let\hskip\tabu@hskip}}% local
1398     }% \@ifpackageloaded colortbl
1399   }%
1400 }% AtBeginDocument

```

8.16 Utilities

`tabu \fbox`

`\tabu@fbox` works exactly like L^AT_EX `\fbox` but allows the syntax: `\fbox\bgroup...\egroup` suitable for use inside tabular columns. `\fbox` is `\let` to `\tabu@fbox` at the entry inside a `tabu` environment.

```

1401 \def\tabu@fbox{%
1402   \leavevmode
1403   \let\color@bgroup\bgroup
1404   \def\color@egroup{\endgraf\egroup}%
1405   \afterassignment\tabu@begin@fbox
1406   \setbox\@tempboxa \hbox
1407 }% \tabu@fbox
1408 \def\tabu@begin@fbox{\color@bgroup\kern\fboxsep\aftergroup\tabu@end@fbox}
1409 \def\tabu@end@fbox{\kern\fboxsep \color@egroup \@frameb@x\relax}

```

tabu \fcolorbox

`\tabu@fcolorbox` works exactly like `xcolor \fcolorbox` but allows the syntax:

`\fcolorbox{frame color}{background color}\bgroup...\egroup`

suitable for use inside tabular columns. `\fcolorbox` is `\let` to `\tabu@fcolorbox` at the entry inside a `tabu` environment.

```

1410 \def\tabu@color@b@x#1#2{\leavevmode
1411   \def\tabu@do@color@b@x{#1{#2\color@block{\wd\z@}{\ht\z@}{\dp\z@}\box\z@}}%
1412   \afterassignment\tabu@begin@color@b@x
1413   \setbox\z@ \hbox
1414 }% \tabu@color@b@x
1415 \def\tabu@begin@color@b@x{%
1416   \bgroup \kern\fboxsep \set@color
1417   \aftergroup\tabu@end@color@b@x
1418 }% \tabu@begin@color@b@x
1419 \def\tabu@end@color@b@x{\kern\fboxsep \egroup
1420   \dimen@ \ht\z@ \advance\dimen@\fboxsep \ht\z@ \dimen@
1421   \dimen@ \dp\z@ \advance\dimen@\fboxsep \dp\z@ \dimen@
1422   \tabu@do@color@b@x
1423 }% \tabu@end@color@b@x

```

\centering, \raggedright, \raggedleft

Inside `tabu` environment, no need to add `\arraybackslash` after these commands.

```

1424 \def\tabu@temp#1#2{{\toks@\expandafter{#2\arraybackslash}\xdef#1{\the\toks@}}}
1425 \tabu@temp \tabu@centering \centering
1426 \tabu@temp \tabu@raggedleft \raggedleft
1427 \tabu@temp \tabu@raggedright \raggedright
1428 \def\tabu@trivlist{\expandafter\def\expandafter\@trivlist\expandafter{%
1429   \expandafter\let\expandafter\\ \expandafter\@normalcr\@trivlist}}%

```

8.17 Corrections

delarray compatibility fix for colortbl and arydshln

Both `colortbl` and `arydshln` forgot the control sequence `\@arrayright` which must be expanded by `\endarray`. Originally defined for `delarray`, this control sequence is used by `tabu` environments when `tabu X` columns are present in the preamble.

Here is the fix. We test if `\endarray` contains `\@arrayright` before modifying the control sequence, in case `colortbl` and/or `arydshln` modify their implementation.

```

1430 \def\tabu@fix@arrayright{%
1431   \@ifpackageloaded{arydshln}
1432   {%
1433     \@ifpackageloaded{colortbl}
1434     {% colortbl + arydshln
1435       \def\tabu@endarray{%
1436         \adl@endarray \egroup \adl@arrayrestore \CT@end \egroup %<original>
1437         \@arrayright % <FC>
1438         \gdef\@preamble{}% <FC>
1439       }% \endarray
1440     }% arydshln / no colortbl
1441   \def\tabu@endarray{%
1442     \adl@endarray \egroup \adl@arrayrestore \egroup %<original>
1443     \@arrayright % <FC>
1444     \gdef\@preamble{}% <FC>
1445   }% \endarray

```

```

1446 }%
1447 {%
1448 \@ifpackageloaded{colortbl}
1449   {% colortbl / no arydshln
1450   \def\tabu@endarray{%
1451     \crrc \egroup \egroup
1452     \@arrayright % <FC>
1453     \gdef\@preamble{}\CT@end
1454   }%
1455   {\PackageWarning{tabu}
1456     {\string\@arrayright\space is missing from the
1457     \MessageBreak definition of \string@endarray.
1458     \MessageBreak Comptability with delarray.sty is broken.}}%
1459 }%
1460}% \tabu@fix@arrayright

```

arydshln @ columns

```

1461 \def\tabu@adl@xarraydashrule#1#2#3{%
1462   \ifnum\@lastchclass=\adl@class@start\else
1463   \ifnum\@lastchclass=\@ne\else
1464   \ifnum\@lastchclass=5 \else % <FC> @-arg (class 5) and !-arg (class 1)
1465     \adl@leftrulefalse \fi\fi % must be treated the same
1466   \fi
1467   \ifadl@zwvrule\else \ifadl@inactive\else
1468     \@addtopreamble{\vrule\@width\arrayrulewidth
1469       \@height\z@ \@depth\z@}\fi \fi
1470   \ifadl@leftrule
1471     \@addtopreamble{\adl@vlineL{\CT@arc@}{\adl@dashgapcolor}%
1472       {\number#1}#3}%
1473   \else \adl@vlineR{\CT@arc@}{\adl@dashgapcolor}%
1474     {\number#2}#3}
1475   \fi
1476}% \tabu@adl@xarraydashrule

```

arydshln, colors without colortbl and empty p columns

arydshln redefines \@endpbox for p columns. The definition is stored in \adl@act@endpbox. Here it is:

```

\unskip \ifhmode \nobreak
  \vrule\@width\z@\@height\z@\@depth\dp\@arstrutbox
\fi
\egroup \adl@colhtdp \box\adl@box \hfil

```

The \vrule inserted is exactly what package array calls: \@finalstrut \@arstrutbox.

However, just like in array.sty, this array-strut should be inserted inconditionnally, and \ifhmode applies only to \nobreak (misplaced \fi in arydshln definition).

Finally, arydshln is not compatible with colors in columns, such that: >{\color{red}}p3in, Unless colortbl is also loaded, the color group is missing.

Fixed inside tabu.

```

1477 \def\tabu@adl@act@endpbox{%
1478   \unskip \ifhmode \nobreak \fi \adl@finalstrut \@arstrutbox
1479   \egroup \egroup
1480   \adl@colhtdp \box\adl@box \hfil
1481 }% \tabu@adl@act@endpbox
1482 \def\tabu@adl@fix{%
1483   \let\adl@xarraydashrule \tabu@adl@xarraydashrule % <fix> arydshln
1484   \let\adl@act@endpbox \tabu@adl@act@endpbox % <fix> arydshln

```


```

1485 \let\adl@act@@endpbox \tabu@adl@act@endpbox % <fix> arydshln
1486 \let\@preamerror \@preamerr % <fix> arydshln
1487 }% \tabu@adl@fix

1488 \end{package}

```

9 References

- [1] *A new implementation of L^AT_EX's tabular and array environments* by Frank Mittelbach
2008/09/09 v2.4c – Tabular extension package (FMi)
CTAN:help/Catalogue/entries/array.html
- [2] *The varwidth package* by Donald Arseneau
2009/03/30 ver 0.92 – Variable-width minipages
CTAN:help/Catalogue/entries/varwidth.html
- [3] *The enumitem-zref package* by 
2010/11/28 ver 1.1 – Extended references for enumitem pkg
CTAN:help/Catalogue/entries/enumitem-zref.html

10 History

[2011/01/15 v1.9]

- Bug in `\savetabu` when used inside `longtabu...`
- Bug when `tabu` with `X` column is nested inside `lontabu`.
- Documentation (`\rowfont` was missing in the `summary`).

[2010/12/28 v1.8]

- `\tracingtabu` / `debugshow` package option:
reporting of the time elapsed during trials (if `\pdfelapsedtime` and thus pdfT_EX is available)
Slight modifications for better reporting on the `.log` file.
- Fix a bug when `\savetabu` is used after `\multicolumn` (`\multicolumn` globally redefines `\@preamble`).
- Fix a bug with `\tabucline` and `\CT@arc@` (`colortbl`).
- Better privacy of columns types specifically defined for `tabu`.
- Improvement in the rewriting process (but only very few people should notice...)
- Documentation.

[2010/12/18 v1.7]

- Code optimisation
- Modification in the columns rewriting process (bug with some new column types defined by the user).

[2010/12/07 v1.5]

- Implementation of negativ width coefficients for `X` columns (cf. [tabu X columns point 2](#)).
- Columns natural widths computation (for `tabu spread` with `X` columns and negativ coefficients) is based on the code of the `varwidth` package by Donald Arseneau.
- `longtabu` is now provided, based on the `longtable` package by David Carlisle.
`longtabu` can be used just like `tabu`.
- Vertical lines can be used whatever the catcode of `|` is.
- `\savetabu` reports saved informations in the `.log` (`debugshow` option).
- `\savetabu ... \usetabu` now restores the `\halign` preamble rather than the `tabu` preamble!
`\preamble` can be use in the `tabu` preamble to restore a `tabu` preamble.
- `\tabucline` is more robust with “special” preambles containing `>` or `<` tokens.
`\tabucline` now takes care of `\arrayrulecolor` (package `colortbl`).
- `enumitem-zref` package has been added to the documentation (see the link [point 1](#))
- Optimisation of some parts of the code.

[2010/11/22 v1.4]

- Compatibility improvement with linegoal for the syntax: `\begin{tabu} to\linegoal{...}`
- Hyper footnotes now work correctly.
- Fix a bug when using colored vertical lines in `tabu` in math mode.
- Fix a bug with vertical lines and `colortbl \arrayrulecolor` specification.
- Fix a compatibility bug with `arydshln`: when nesting a tabular that use vertical dashed lines (`arydshln`) inside `tabu spread` with `X` columns.

[2010/11/18 v1.3]

- Fix a bug that may appear in `\tabucline` depending on the preamble due to arbitrary `\countdef`.
- Improvement in the use of `\everycr`: no `\global` stuff. Thus bug fixed when nesting `tabu` inside \mathcal{S} -align environment for example. Same issue with `\rowfont` which now works without global modification of `\everycr`.
- No phantom line is added to `tabu` but a command `\tabuphantomline` is provided for this purpose (required with `\multicolumn` in some cases).
- Improvement on vertical alignment.
- To do: an example file to test a wide range of possibilities...
- Documentation.

[2010/11/15 v1.2]

- Improvement in parameters parsing for optional parameters (`|` and `\tabucline`).
- Modification / optimization in `\tabu@prepnext@tok`.
- Modification of `\tabucline` to get better results with `m` columns (`X[m]`) and also when `\minrowclearance > 0` (package `colortbl`).

[2010/10/28 v1.1]

- First version.

11 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

Symbols	
<code>\@arrayright</code>	57, 530, 645, 1437, 1443, 1452, 1456
<code>\@arstrutbox</code>	82, 798, 841, 847, 853, 907, 908, 932, 1180, 1478
<code>\@chclass</code>	1265, 1266
<code>\@currentHref</code>	369
<code>\@currentlabelname</code>	370
<code>\@defaultunits</code>	941, 942, 949, 968, 971
<code>\@elt</code>	534, 535
<code>\@finalstrut</code>	798, 1478
<code>\@firstoftwo</code>	292, 777
<code>\@flushglue</code>	193, 196, 197, 786, 1314
<code>\@footnotemark</code>	342
<code>\@footnotetext</code>	251, 529
<code>\@gobbletwo</code>	755
<code>\@halignto</code>	114, 170, 183, 522, 647, 1143
<code>\@ifnextchar</code>	29, 85
<code>\@let@token</code>	271, 273, 274, 275, 276, 277, 426, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443
<code>\@m</code>	730
<code>\@makeother</code>	1062
<code>\@mpfn</code>	330, 339, 348
<code>\@multicnt</code>	903, 909, 913, 914, 916, 921, 922, 924, 925, 929
<code>\@normalcr</code>	254, 1429
<code>\@preamble</code>	151, 546, 552, 556, 1145, 1438, 1444, 1453
<code>\@preamerr</code>	1486
<code>\@preamerror</code>	1486
<code>\@rightskip</code>	1313
<code>\@sanitize</code>	1055
<code>\@secondoftwo</code>	780
<code>\@spaces</code>	725

<code>\@sptoken</code> . . .	188, 275, 440, 1018, 1073, 1107	<code>\color@bgroup</code>	1403, 1408
<code>\@tempcnta</code>	524, 1220, 1226	<code>\color@block</code>	1411
<code>\@tempdimb</code>	974, 977, 982, 991, 992, 994, 996, 1003, 1007	<code>\color@egroup</code>	1404, 1409
<code>\@tempdimc</code>	987, 992, 993, 994, 997, 999, 1000, 1009, 1010	<code>\copy</code>	953
<code>\@temptokena</code>	305, 318, 324, 828, 1065, 1081, 1085, 1116, 1119, 1273, 1274	<code>\crrr</code>	603, 840, 1451
<code>\@thefnmark</code>	340, 350	<code>\CT@drsc@</code>	307, 750
<code>\@trivlist</code>	1428, 1429	<code>\currentgrouptype</code>	1349, 1354
<code>\@undefined</code>	6, 511	D	
<code>\@whilenum</code>	849, 909, 929	<code>\d@llarend</code>	144
<code>\@whilesw</code>	653	<code>\delarray_[(package_)]option)</code>	40
<code>\@xfootnote</code>	252, 335, 529	<code>\detokenize</code>	753, 954, 955, 1037
<code>\@xfootnotetext</code>	251	<code>\dimen@</code>	608, 613, 631, 649, 1420, 1421
<code>\{</code>	1062	<code>\doublerulesep</code>	313, 743
<code>\}</code>	1062	<code>\dp</code>	1182, 1411, 1421
<code>\^</code>	577, 1054, 1055, 1056	E	
<code>_</code>	1056	<code>\endarray</code> 57, 58, 219, 233, 592, 1439, 1445, 1457	
A		<code>\endlinechar</code>	191, 194, 860, 1057
<code>\adl@act@@endpbox</code>	1485	<code>\endlongtable</code>	226
<code>\adl@act@endpbox</code>	1484	<code>\endlongtabu</code>	226
<code>\adl@arrayrestore</code>	1436, 1442	<code>\endtabu</code>	218
<code>\adl@box</code>	1480	<code>\endtabular</code>	220
<code>\adl@class@start</code>	1462	<code>\everycr</code>	259, 827, 828
<code>\adl@colhtdp</code>	1480	<code>\everyeof</code>	1063
<code>\adl@dashgapcolor</code>	1471, 1473	<code>\everypar</code>	243, 244
<code>\adl@endarray</code>	1436, 1442	<code>\extracolsep</code>	181, 193, 196, 197
<code>\adl@leftrulefalse</code>	1465	<code>\extrarowheight</code>	314, 744, 847, 960, 1005, 1006, 1049, 1050, 1179
<code>\adl@ncol</code>	603	<code>\extratabsurround</code>	315, 745
<code>\adl@vlineL</code>	1471	F	
<code>\adl@vlineR</code>	1473	<code>\footnotetext</code>	330, 334
<code>\adl@xarraydashrule</code>	1483	<code>\futurelet</code>	186, 271, 289, 426, 876, 1014, 1021, 1022, 1069
<code>\afterassignment</code>	30, 208, 288, 473, 828, 967, 1405, 1412	G	
<code>\aftergroup</code>	137, 174, 175, 176, 179, 205, 212, 216, 241, 269, 787, 1151, 1160, 1386, 1408, 1417	<code>\GenericError</code>	214, 528
<code>\array</code>	219	<code>\gluestretchorder</code>	1375, 1380
<code>\arraybackslash</code>	785, 1284, 1287, 1292, 1295, 1300, 1303, 1319, 1322, 1327, 1330, 1335, 1338, 1342, 1344, 1424	H	
<code>\arraycolsep</code>	310	<code>\H@@@footnotetext</code>	362
<code>\arrayrulewidth</code>	312, 742, 874, 882, 973, 976, 980, 1047, 1049, 1468	<code>\hbadness</code>	524
<code>\arraystretch</code>	316, 747, 1181, 1182	<code>\hfilneg</code>	1381
<code>\AtEndOfPackage</code>	6, 33, 36, 41, 44	<code>\hfuzz</code>	240, 525
B		<code>\ht</code>	847, 908, 1178, 1411, 1420
<code>\begin</code>	4, 4	<code>\Hy@footnote@currentHref</code>	353
C		<code>\Hy@raisedlink</code>	366
<code>\c@footnote</code>	336	<code>\hyper@@anchor</code>	364, 367
<code>\c@page</code>	397	I	
<code>\catcode</code> 11, 13, 577, 1054, 1055, 1056, 1058, 1059		<code>\ifadl@inactive</code>	1467
<code>\Centering</code>	453, 1327, 1330	<code>\ifadl@leftrule</code>	1470
<code>\centering</code>	252, 450, 1292, 1295, 1425	<code>\ifadl@zwvrule</code>	1467
<code>\chardef</code>	82	<code>\ifcase</code>	273, 429, 509, 610, 831, 916, 921, 972, 1016, 1072, 1089, 1134, 1226, 1249
<code>\cl@@ckpt</code>	535	<code>\ifcat</code>	144, 443, 881, 972, 1019
<code>\color</code>	751, 1030, 1196, 1204	<code>\ifcsname</code>	376, 653, 771, 954, 1029, 1197
<code>\color@</code>	1029	<code>\ifdim</code> 180, 242, 246, 296, 392, 481, 484, 502, 503, 610, 611, 613, 614, 623, 627, 631, 635, 646, 654, 657, 660, 671, 674, 678, 682, 697, 778, 800, 804, 806, 807, 991, 992, 993, 994, 995, 996, 997, 999, 1003, 1007, 1009, 1135, 1153, 1172, 1350, 1379	
<code>\color@b@x</code>	256	<code>\iffalse</code>	506
		<code>\ifHy@nesting</code>	363

`\ifn@` 58, 408, 769, 777, 779
`\ifnum` 31, 506,
 541, 551, 615, 634, 644, 712, 728, 729,
 730, 732, 733, 847, 899, 904, 913, 916,
 921, 924, 935, 943, 944, 945, 946, 1171,
 1191, 1201, 1219, 1235, 1240, 1265, 1266,
 1349, 1354, 1375, 1380, 1462, 1463, 1464
`\iftabu@cellright` 1216, 1231, 1245
`\iftabu@colortbl` [207](#), 305, 317, 748, 842, 855,
 873, 875, 893, 959, 988, 1044, 1348, 1373
`\iftabu@firstcline` [77](#), 908, 1001, 1046
`\iftabu@measuring` [77](#), 384, 570, 596, 640
`\iftabu@negcoef` .. [77](#), 174, 509, 609, 763, 1130
`\iftabu@spread` [77](#), 390, 509, 567, 593, 776
`\ignorespaces` 1072, 1106
`\immediate` 210, 212, 216
`\in@` 57, 408, 768
`\in@false` 778

J

`\justifying` 1342, 1344

L

`\lastbox` 245
`\lastskip` 1374, 1379, 1380
`\leaders` 1045
`\leftskip` 1313
`\light_(package_option)` [43](#)
`\linebreak` 1056
`\linegoal` 38, 396
`\linegoal_(package_option)` [34](#)
`\linewidth` [34](#), 38, 238, 242, 246, 395, 527
`\LNGL@setlinegoal` 291, 292
`\longtable` 226
`\longtabu` 63, 224, 229
`\loop` 911
`\lower` 960
`\LT@bchunk` 549, 553, 558
`\LT@echunk` 563
`\LT@get@widths` 565
`\LT@output` 559
`\LT@chunksize` 557

M

`\maxdimen` 525
`\message` 24, 32
`\minrowclearance` 317, 746, 843, 844, 856

N

`\narrowragged` 785
`\NC@find` 115, 146,
 162, 166, 169, 193, 196, 197, 320, 753, 868
`\NC@rewrite@X` 135
`\newcount` 65, 66, 67, 68, 69, 70
`\newdimen` 71, 72, 73, 74, 75, 76
`\newsavebox` 81
`\newskip` 1368
`\newtoks` 327
`\noalign` 603, 830, 840,
 852, 899, 907, 932, 1122, 1191, 1199, 1206
`\null` 1056
`\numexpr` 715, 716, 717, 1131, 1226

O

`\omit` 912
`\on@line` 397, 711, 757

P

`\PackageError` 51, 53, 225, 230, 1188
`\PackageInfo` 1185
`\PackageWarning` 217, 463, 468, 961, 1167, 1455
`\parfillskip` 786, 1056, 1314
`\pdfadjustspacing` 792, 793
`\pdfelapsedtime` 26, 27, 715, 716, 717
`\preamble` [13](#), 124, 130, 140
`\preamble_(private_column_type)` [124](#)
`\prepnext@tok` 45, 258, 1217
`\ProcessOptions` 55
`\protected` 211, 215
`\protected@xdef` 340, 349

R

`\RaggedLeft` 454, 1335, 1338
`\raggedleft` 253, 451, 1300, 1303, 1426
`\RaggedRight` 452, 1319, 1322
`\raggedright` 253, 449, 1284, 1287, 1427
`\remove@to@nnil` 445, 1024, 1031
`\repeat` 928
`\rightskip` 1313
`\rowfont` [13](#), 52, 257, 258, [1191](#)

S

`\save@decl` 45, 145
`\savetabu` [12](#), 300, 527, 643, 757, 758, [1122](#), 1186
`\scantokens` 191, 194, 860, 1063
`\set@color` 1030, 1416
`\setbox` 245, 543, 555, 564, 783, 1180, 1406, 1413
`\setlength` 393
`\span` 927
`\stretch` 1349, 1351
`\strip@pt` 386, 400, 488
`\strutbox` 1178, 1182

T

`\tabcolsep` 311
`\tabu` [218](#)
`\tabu@` 219, 220, 226, 302
`\tabu@@begin` 297, 299, 303
`\tabu@@celleft` 1194, 1208
`\tabu@@cellright` 1195, 1209
`\tabu@@rewrite@X` 407, 784
`\tabu@@tabudecimal` 1087
`\tabu@add` 914, 922, 937
`\tabu@adl@act@endpbox` 1477, 1481, 1484, 1485
`\tabu@adl@endTRIAL` 62, 591, 603
`\tabu@adl@fix` 62, 233, 1482, 1487
`\tabu@adl@xarraydashrule` .. 1461, 1476, 1483
`\tabu@align` 301, 302, 908, 1148, 1149, 1157, 1158
`\tabu@aligndefault` 238, 240, 297, 1166
`\tabu@aligndefault@text` ... 1148, 1157, 1166
`\tabu@arith` 568, 594, [604](#)
`\tabu@arith@negcoef` 609, 651, 666
`\tabu@arrayleft@measure` 221, 227, 519
`\tabu@arrayleftmeasure` 221, [539](#)
`\tabu@arstrutbox` 82, 853, 932
`\tabu@begin` [271](#)
`\tabu@begin@color@b@x` ... 1412, 1415, 1418
`\tabu@begin@fbox` 1405, 1408
`\tabu@box` [81](#), 543, 555, 566, 610, 611, 620,
 625, 632, 634, 671, 673, 682, 693, 698,
 702, 707, 783, 798, 799, 802, 807, 808, 815
`\tabu@cell@C` 1325, 1332, 1364
`\tabu@cell@c` [1282](#)

<code>\tabu@cell@J</code>	1341, 1346, 1365	<code>\tabu@GenericErrorORI</code>	213, 214
<code>\tabu@cell@j</code>	<u>1282</u>	<code>\tabu@get@decimal</code>	1081, 1109
<code>\tabu@cell@L</code>	1317, 1324, 1362	<code>\tabu@get@decimalspace</code>	1084, 1111
<code>\tabu@cell@l</code>	<u>1282</u>	<code>\tabu@getdecimal</code>	1075, <u>1088</u>
<code>\tabu@cell@R</code>	1333, 1340, 1363	<code>\tabu@getdecimal@</code>	1066, 1069, 1075
<code>\tabu@cell@r</code>	<u>1282</u>	<code>\tabu@getdecimal@ignorespaces</code>	1066, 1071, 1080
<code>\tabu@cellfont</code>	1196, 1199, 1210, <u>1277</u>	<code>\tabu@getline</code>	956, <u>966</u>
<code>\tabu@celllalign</code>	1211, 1258, <u>1277</u> , 1283, 1291, 1299, 1307, 1318, 1326, 1334	<code>\tabu@getlinecolor</code>	883, 1023, 1027, 1032
<code>\tabu@celllalign</code>	1342	<code>\tabu@getlinespec</code>	903, <u>948</u> , 1036
<code>\tabu@cellleft</code>	1105, 1194, 1199, 1200, 1208, 1259, <u>1277</u> , 1285, 1287, 1288, 1293, 1295, 1296, 1301, 1303, 1304, 1307, 1309, 1310, 1320, 1322, 1323, 1328, 1330, 1331, 1336, 1338, 1339, 1342, 1344, 1345	<code>\tabu@gettarget</code>	286, 287, 288
<code>\tabu@cellralign</code> ...	1212, 1255, <u>1277</u> , 1286, 1294, 1302, 1308, 1321, 1329, 1337, 1343	<code>\tabu@global@temp</code>	136, 137, 149, 1222, 1229, 1230, 1273, 1274
<code>\tabu@cellright</code>	1195, 1209, 1255, <u>1277</u>	<code>\tabu@global@X</code>	171, 204, 205, 234, 385, 399, 521, 535, 542, 553, 557, 641
<code>\tabu@cellrightfalse</code>	1224, 1232, 1246	<code>\tabu@global@X@nested</code>	201, 203
<code>\tabu@cellrighttrue</code>	1238	<code>\tabu@global@Xnested</code>	173, 201, 206, 541, 551
<code>\tabu@cellskip</code> ..	1350, 1368, 1374, 1375, 1376	<code>\tabu@gobble@pt</code>	85, 86
<code>\tabu@centering</code>	252, 1425	<code>\tabu@gobbleopt</code>	<u>84</u> , 411
<code>\tabu@checklast</code>	155, 157, 165, 167	<code>\tabu@gobblespace</code> ..	<u>84</u> , 189, 280, 1022, 1077
<code>\tabu@cline</code>	900, 901, 902, 936	<code>\tabu@halignto</code>	172, 183, 286, 287, 296, 320, 404, 687, 1143
<code>\tabu@cnt</code> <u>65</u> , 473, 476, 518, 605, 615, 634, 732		<code>\tabu@header</code>	619, 691, 705, 732
<code>\tabu@collect</code>	547, 560, <u>577</u>	<code>\tabu@hfil</code>	1369, 1389, 1395
<code>\tabu@collectbody</code>	577, 578, 585, 587	<code>\tabu@hfill</code>	1370, 1396
<code>\tabu@color</code>	880, 892, 894, 967, 970, 981, 988, 989, 998, 1030	<code>\tabu@hfuzz</code>	240, 610, 621, 709, 711
<code>\tabu@color@b@x</code>	256, 1410, 1414	<code>\tabu@hskip</code>	1371, 1397
<code>\tabu@colorleavevmode</code>	1196, 1203	<code>\tabu@Hy@footnotetext</code>	351, 361, 374
<code>\tabu@colorORI</code>	1196, 1203, 1204	<code>\tabu@Hy@ftntext</code>	<u>336</u>
<code>\tabu@colortblfalse</code>	60	<code>\tabu@Hy@ftntxt</code>	336, 342, 344, 360
<code>\tabu@colortbltrue</code>	60	<code>\tabu@Hy@xfootnote</code>	<u>336</u>
<code>\tabu@colortest</code>	967, <u>1014</u>	<code>\tabu@ignorehfil</code>	1286, 1294, 1302, 1321, 1329, 1337, <u>1385</u>
<code>\tabu@dashgap</code>	898, 975, 978, 992	<code>\tabu@immediate</code>	210, 213, 217
<code>\tabu@debug</code>	21, 32, 809, 933	<code>\tabu@islast</code>	138, 159, 161
<code>\tabu@decimal</code>	1065, 1119	<code>\tabu@justify</code>	1307, 1309, 1312, 1315
<code>\tabu@defaultleaders</code> ...	950, 964, 1043, 1044	<code>\tabu@lastrewrite</code>	160, 168, 185
<code>\tabu@defleaders</code>	951, 952, 953, 958	<code>\tabu@lasttry</code>	518, 611, 632, 634
<code>\tabu@dimexpr</code>	133, 417, 498, 501, 504, 768, 777, 780	<code>\tabu@leaderfill</code>	960, 1011, 1052, 1053
<code>\tabu@do@color@b@x</code>	1411, 1422	<code>\tabu@linecolor</code> .	1014, 1015, 1021, 1022, 1026
<code>\tabu@elapsedtime</code>	28, 712	<code>\tabu@linedash</code>	897, 966, 994
<code>\tabu@end@color@b@x</code>	1417, 1419, 1423	<code>\tabu@linegoal@target</code>	289, 290, 294
<code>\tabu@end@fbox</code>	1408, 1409	<code>\tabu@linegoaltarget</code>	288, 289
<code>\tabu@endarray</code>	58, 233, 1435, 1441, 1450	<code>\tabu@lineoff</code>	975, 978, 983, 986, 1013
<code>\tabu@endenvir</code>	574, 581, 600	<code>\tabu@lineon</code>	966, 969, 985
<code>\tabu@endrewrite</code>	154, 166, 167	<code>\tabu@lines</code>	47, 141, <u>860</u>
<code>\tabu@endTRIAL</code>	544, <u>590</u>	<code>\tabu@lines@</code>	861, 863, 866
<code>\tabu@everycr</code>	259, 827, 829	<code>\tabu@linestyle</code>	1034, 1035, 1041
<code>\tabu@extracol@sep</code>	186, 187, 189, 200	<code>\tabu@long@arrayleft@measure</code>	227, <u>549</u>
<code>\tabu@extracolsep</code>	181, 186	<code>\tabu@long@endTRIAL</code>	558, 562, 576
<code>\tabu@fbox</code>	256, 1401, 1407	<code>\tabu@make@arstrutbox</code> ...	1146, 1177, 1184
<code>\tabu@first@cline</code> .	826, 831, 832, 833, 835, 857	<code>\tabu@makeatletter</code>	1055
<code>\tabu@firstcline</code>	47	<code>\tabu@maybeicolor</code>	888, 967, 973, 974, 977, 980, 987
<code>\tabu@firstclinefalse</code>	834	<code>\tabu@measure@endpbox</code>	<u>766</u>
<code>\tabu@firstclinettrue</code>	825	<code>\tabu@measure@startpbox</code> .	516, 606, 696, <u>766</u>
<code>\tabu@fix@arrayright</code>	59, 1430, 1460	<code>\tabu@measuringfalse</code>	260, 570, 596
<code>\tabu@flush</code>	1286, 1293, 1294, 1301, 1321, 1328, 1329, 1336, 1347, 1358	<code>\tabu@measuringtrue</code> .	389, 616, 623, 630, 689
<code>\tabu@footnotetext</code>	<u>328</u>	<code>\tabu@message</code> 19, 24, 394, 398, 605, 613, 614, 615, 619, 659, 670, 677, 678, 681, 682, 756	
<code>\tabu@footnotes</code> 241, <u>327</u> , 329, 331, 333, 346, 356		<code>\tabu@message@arith</code>	605, <u>691</u>
<code>\tabu@footnotetext</code>	251, 328, 378	<code>\tabu@message@etime</code>	28, 713
<code>\tabu@GenericError</code>	211, 528	<code>\tabu@message@negcoef</code>	659, <u>691</u>
		<code>\tabu@message@reached</code> ..	613, 614, 615, 705
		<code>\tabu@message@save</code>	20, 25, 1163
		<code>\tabu@message@spread</code>	670, <u>691</u>
		<code>\tabu@minrowclearance</code>	843, 856

`\tabu@msg@align` . . . 681, 692, 693, 694, 695,
 698, 701, 702, 703, 704, 706, 707, 708, 727
`\tabu@naturalX` 71,
 266, 522, 668, 673, 704, 773, 799, 800, 801
`\tabu@naturalX@max` 268, 513, 668, 800, 801, 823
`\tabu@naturalX@min` 267, 514, 669, 804, 805, 822
`\tabu@nbcols` 65, 603, 849, 929, 939,
 943, 945, 946, 1171, 1221, 1222, 1227,
 1228, 1230, 1235, 1240, 1242, 1243, 1262
`\tabu@NC@list` 143, 150, 154, 261
`\tabu@NC@rewrite@X` 135
`\tabu@negcoefficient` 260
`\tabu@negcoeftrue` 174, 481
`\tabu@nested` 65, 202, 237,
 264, 269, 644, 712, 733, 817, 819, 821, 847
`\tabu@new@columntype` 91, 96, 99, 110
`\tabu@newcolumntype` 90, 131, 153, 166, 167, 864
`\tabu@next` 169, 181, 184,
 193, 196, 197, 198, 1021, 1022, 1023, 1025
`\tabu@noadjustspacing` 786, 793, 795
`\tabu@noline` 46, 53
`\tabu@nohfil` 1386, 1388, 1390, 1394
`\tabu@nolongtabu` 63, 230
`\tabu@norowfont` 46, 51
`\tabu@nowrite` 208, 528
`\tabu@preamble` 1156, 1160
`\tabu@prep@TRIAL` 175, 505
`\tabu@prepare@save` 306, 307, 322, 326
`\tabu@preparesave` 300, 304
`\tabu@prepnext@tok` 45, 258, 1216
`\tabu@prepnext@tokORI` 1217, 1251
`\tabu@preptokenleft` 1236, 1258, 1260
`\tabu@preptokenright` . 1233, 1247, 1253, 1257
`\tabu@printdecimal` 1112, 1115, 1121
`\tabu@private@columntype` . . . 101, 108, 111
`\tabu@privatecolumns` 109, 112, 139
`\tabu@privatecolumntype` . . 90, 113, 116, 124
`\tabu@pushbegins` 579, 588, 589
`\tabu@raggedleft` 253, 1426
`\tabu@raggedright` 253, 1427
`\tabu@removehfil`
 1283, 1291, 1299, 1318, 1326, 1334, 1368
`\tabu@restoreeverycr` 827, 828
`\tabu@rewrite@X` 113, 120, 382, 526, 784
`\tabu@rewrite@Xnested` 408, 526
`\tabu@rewrite@Xrestore` 120, 408
`\tabu@rewritefirst` 131, 147, 261, 302
`\tabu@rewritefirst_(new_columntype)` . 131
`\tabu@rewritelast` 154, 162, 166
`\tabu@rewritelast_(new_columntype)` . . 153
`\tabu@rewritemiddle` 143, 146, 153, 156
`\tabu@row@font` 1191, 1192
`\tabu@rowfont` 46, 257, 1191
`\tabu@rowfont@reset` 1193, 1205
`\tabu@rowfontreset` . . . 837, 1193, 1207, 1215
`\tabu@sanitizetext` 235, 255, 1054
`\tabu@save@decl` 45, 87, 145
`\tabu@savecounters` 523, 533, 536
`\tabu@saveditparam` 309, 761, 1127
`\tabu@saveditpream` 320, 762, 1128
`\tabu@saveditpreamble` 151, 546, 552, 556, 1126
`\tabu@saveerr` 117, 119, 125, 127, 1185
`\tabu@savepreamble` 134, 136, 148, 152
`\tabu@savetok` 1262, 1264
`\tabu@savetok` 1233,
 1235, 1238, 1241, 1243, 1261, 1263, 1276
`\tabu@savewarn` 1123, 1124, 1185
`\tabu@savewd` 763, 1131, 1132, 1170, 1174, 1176
`\tabu@scandecimal`
 1067, 1069, 1070, 1077, 1082, 1085
`\tabu@settarget` 222, 228, 271
`\tabu@setup` 222, 228, 232
`\tabu@setup@everycr`
 48, 176, 179, 545, 552, 556, 824
`\tabu@setup@saveglobal` 236, 263, 270
`\tabu@setup@TRIAL` 507, 521
`\tabu@skipdecimal` 1070, 1076, 1078, 1110
`\tabu@spaces` 696, 710, 713, 719, 725, 809
`\tabu@spread` 282, 287
`\tabu@spreadarith` 567, 593, 667
`\tabu@spreadfalse` 260, 689
`\tabu@spreadheader` 700, 736
`\tabu@spreadtarget` 71,
 390, 391, 403, 671, 672, 674, 675, 678, 701
`\tabu@spreadtrue` 287
`\tabu@start` 65, 904, 909, 941, 943, 944
`\tabu@start@stop` 939, 940, 947
`\tabu@startpboxORI` 515, 607, 686, 767
`\tabu@startstop` 903, 939
`\tabu@starttime` 27, 715, 716, 717
`\tabu@stop` 65, 904, 916, 921, 924, 942, 945, 946
`\tabu@tabudecimal` . . . 255, 1064, 1068, 1087
`\tabu@target` 71, 180,
 272, 288, 296, 320, 390, 392, 393, 397,
 402, 508, 610, 621, 625, 643, 647, 671,
 674, 675, 677, 678, 681, 682, 687, 688,
 694, 698, 708, 760, 1135, 1138, 1153, 1154
`\tabu@temp` 158, 159, 161, 186,
 188, 192, 195, 289, 291, 418, 422, 472,
 476, 488, 496, 498, 655, 657, 658, 781,
 785, 811, 868, 872, 876, 881, 891, 1014,
 1016, 1017, 1018, 1019, 1021, 1022, 1069,
 1072, 1073, 1089, 1090, 1091, 1092, 1093,
 1094, 1095, 1096, 1097, 1098, 1099, 1100,
 1101, 1102, 1103, 1104, 1105, 1106, 1107,
 1116, 1117, 1118, 1424, 1425, 1426, 1427
`\tabu@timerstart` 22, 27, 239
`\tabu@titles` 398, 733
`\tabu@to` 281, 286
`\tabu@tr@cing` 30, 31
`\tabu@tracing` 29, 30
`\tabu@tracing@save` 25, 740
`\tabu@TRIAL` 539, 554, 571, 582, 597
`\tabu@TRIAL@hook` 531, 537, 538
`\tabu@trivlist` 254, 1428
`\tabu@usetabu` 1141, 1151
`\tabu@vcenter` 83, 854
`\tabu@verb` 1055, 1061
`\tabu@vline` 869, 874, 895
`\tabu@warn` 215, 620, 628
`\tabu@warn@usetabu` 1136, 1167
`\tabu@wd` . . . 502, 503, 654, 657, 660, 664,
 721, 778, 806, 807, 812, 813, 817, 1172, 1173
`\tabu@wddef` 754, 808, 818, 1173
`\tabu@wdlet` 414, 483, 820
`\tabu@write` 209, 213, 217
`\tabu@X@cols` 65, 132, 171, 265, 383, 386, 413,
 414, 417, 483, 498, 510, 511, 512, 652,

653, 654, 656, 657, 659, 660, 662, 770, 771, 772, 803, 806, 807, 808, 810, 812, 813	<code>\tabuvline@arg</code> 867, 870	
<code>\tabu@X@sum</code> 74, 386, 387, 400, 401, 416, 420, 476, 481, 482, 484, 488, 608, 613, 614, 617, 623, 625, 626, 631, 646, 649, 658, 695, 697, 699, 802	<code>\tabuvline@argi</code> 876, 879, 886	
<code>\tabu@Xalign</code> 449, 450, 451, 452, 453, 454, 462, 466	<code>\tabuvline@argii</code> 884, 887, 889	
<code>\tabu@Xarg</code> 382, 412	<code>\tabuvline@argiii</code> 876, 890, 896	
<code>\tabu@Xc@ef</code> 473, 475, 478	<code>\tabuvline@rewrite</code> 860	
<code>\tabu@Xcoef</code> 456, 458, 472, 474	<code>\themppfn</code> 340, 350	
<code>\tabu@Xfinish</code> 540, 550, 573, 599	<code>\toks</code> 88, 1126, 1127, 1128, 1129, 1132, 1140, 1142, 1145, 1147, 1162, 1255, 1258, 1259, 1265, 1266	
<code>\tabu@xfootnote</code> 252, 335, 379	<code>\tracingtabu</code> 16	
<code>\tabu@xfootnotetext</code> 251, 332	<code>\tracingtabu_(debugshow_option)</code> 19	
<code>\tabu@xleaders</code> 905, 919, 948, 950, 959, 964, 998, 1038, 1039, 1043	<code>\ttfamily</code> 1060	
<code>\tabu@Xmath</code> 422, 457, 490, 494	U	
<code>\tabu@Xparse</code> 412	<code>\unhbox</code> 564	
<code>\tabu@Xparsespace</code> 427, 455	<code>\unkern</code> 1374	
<code>\tabu@Xtest</code> 426, 428, 461	<code>\unless</code> 1268	
<code>\tabu@Xtype</code> 446, 447, 448, 467, 471	<code>\unskip</code> 1059, 1379, 1478	
<code>\tabucline</code> 8, 46, 54, 258, 897	<code>\usetabu</code> 12, 116, 123, 140, 418, 1168, 1189	
<code>\tabucline*</code> 9	<code>\usetabu_(private_column_type)</code> 116	
<code>\tabucline@warn</code> 961, 965, 981	V	
<code>\tabucolX</code> 74, 409, 410, 417, 418, 488, 508, 617, 623, 625, 627, 629, 635, 642, 657, 669, 673, 688, 692, 703, 706, 723, 759, 1139, 1144	<code>\value</code> 339, 348, 534	
<code>\tabucolX@align</code> 421, 463, 465, 487, 491	<code>\vbox</code> 846	
<code>\tabucolX@do</code> 412	<code>\vcenter</code> 83, 846, 854	
<code>\tabucolX@error</code> 74, 629	<code>\verbatim</code> 10, 235, 255	
<code>\tabucolX@rw</code> 115, 409, 410, 411, 417, 421, 468, 470, 480, 489, 493, 496, 497, 498	<code>\vskip</code> 908	
<code>\tabudecimal</code> 11, 255, 1064	<code>\vspace</code> 847	
<code>\tabudefaultrighttarget</code> 34, 38, 238, 242, 393, 395, 396, 527	<code>\vtop</code> 815	
<code>\tabuDisableCommands</code> 537	W	
<code>\tabular</code> 220	<code>\write</code> 209, 212, 216, 528	
<code>\tabulinestyle</code> 9, 1033	X	
<code>\tabuphantomline</code> 7, 840	<code>\X_(private_column_type)</code> 113	
<code>\tabusaveX</code> 758	<code>\xleaders</code> 960, 998	
	Z	
	<code>\zap@space</code> 1028	