The package cascade*

F. Pantigny
fpantigny@wanadoo.fr

February 8, 2023

Abstract

The LaTeX package cascade provides a command \Cascade to do constructions to present mathematical demonstrations with successive braces for the deductions. The package cascade provides also a command \Edacsac which creates similar structures but with braces going backwards.

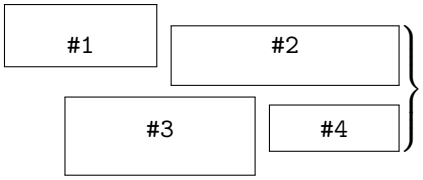
1 The command \Cascade

The package cascade provides a command \Cascade which allows constructions like the following where the size of the right brace is computed on only a part of the LaTeX elements composed on the left.

$$\left.\begin{array}{r}\det(A) = \begin{vmatrix}3 & 4\\ -1 & 7\end{vmatrix} \neq 0 \text{ and, therefore, } A \text{ is inversible}\\ \text{yet } AX = Y\end{array}\right\} \text{hence, } X = A^{-1}Y$$

```
\Cascade{$\det(A) = \begin{vmatrix}3&4\\ -1&7\end{vmatrix}\neq 0$}
        {and, therefore, $A$ is inversible}
        {}
        {yet $AX=Y$}
hence, $X = A^{-1}Y$
```

The command \Cascade takes its four arguments as follow :



The commands \Cascade can be nested as in the following example :

$$\left.\begin{array}{l}\left.\begin{array}{l}(BH) \perp (AC)\\ (OC) \perp (AC)\end{array}\right\} \text{ hence } (BH) \parallel (OC)\\ \left.\begin{array}{l}(CH) \perp (AB)\\ (OB) \perp (AB)\end{array}\right\} \text{ hence } (CH) \parallel (OB)\end{array}\right\} \text{hence } (OBHC) \text{ is a parallelogram}$$

*This document corresponds to the version 1.2a of cascade, at the date of 2023/02/08.

For the legibility of such constructions, a simplified version of `\Cascade` is available, named `\ShortCascade`.
The code `\ShortCascade{X}{Y}` is merely a shortcut for the code `\Cascade{}{X}{}{Y}`.

The preceding example can be coded with two commands `\ShortCascade` and an encompassing command `\Cascade`:

```
\Cascade{\ShortCascade{$(BH) \perp (AC)$}
                      {$(OC) \perp (AC)$}}
        {hence\enskip $(BH) \parallel (OC)$}
        {\ShortCascade{$(CH) \perp (AB)$}
                      {$(OB) \perp (AB)$}}
        {hence\enskip $(CH) \parallel (OB)$}
hence $(OBHC)$ is a parallelogram
```

2 The option t

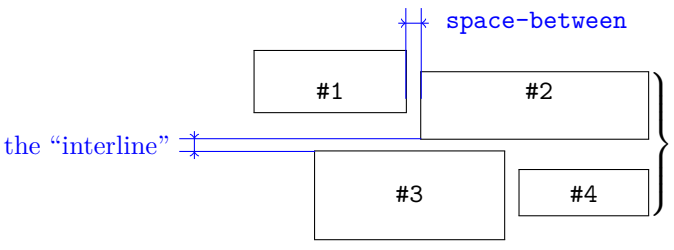
With the option `t` in the encompassing command `\Cascade`, a whole strucutre of nested commands `\Cascade` is aligned on the top line.
When the key `t` is used, if we wish to add some text after the structure, we have to put that text between angle brackets in order to have that text aligned with the last brace.

```
\begin{enumerate}
\item \Cascade[t]{\ShortCascade{$(BH) \perp (AC)$}{$(OC) \perp (AC)$}}
        {hence\enskip $(BH) \parallel (OC)$}
        {\Cascade{}{$(CH) \perp (AB)$}{}{$(OB) \perp (AB)$}}
        {hence\enskip $(CH) \parallel (OB)$}
        <hence $(OBHC)$ is a parallelogram>
\end{enumerate}
```

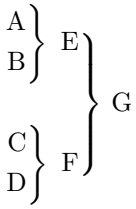
1. $\left.\begin{array}{l}(BH) \perp (AC)\\(OC) \perp (AC)\end{array}\right\}$ hence $(BH) \parallel (OC)$
   $\left.\begin{array}{l}(CH) \perp (AB)\\(OB) \perp (AB)\end{array}\right\}$ hence $(CH) \parallel (OB)$
   $\Big\}$ hence $(OBHC)$ is a parallelogram

3 Other options

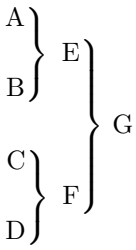
- The option `space-between` is a TeX dimension described on the following figure. Its initial value is 0.5 em. It applies to the current command `\Cascade` but also to the possible nested commands.
- The option `interline` can be used to *increase* the "interline" showed in the following picture. The initial value of `interline` is 0 pt and applies only to the current command `\Cascade`.
- The option `interline-all` changes the default value of `interline` used by the current command `\Cascade` and all the possible nested commands `\Cascade`.



```
\Cascade[interline=4mm]{\ShortCascade{A}{B}}{E}{\ShortCascade{C}{D}}{F} G
```




```
\Cascade[interline-all=4mm]{\ShortCascade{A}{B}}{E}{\ShortCascade{C}{D}}{F} G
```

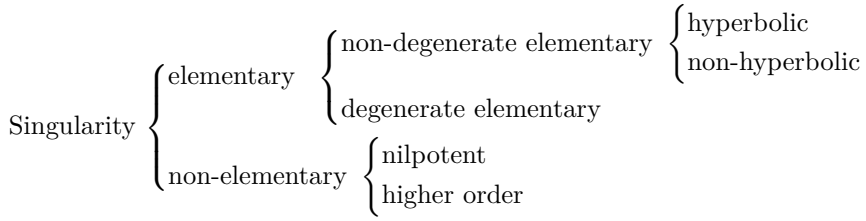


The options can also be given at the document level with the command `\CascadeOptions`. In this case, the scope of the declarations is the current TeX group (these declarations are "semi-global").

4 The command \Edacsac

The command `\Edacsac` (*cascade* written in reverse) is similar to the command `\Cascade` but with braces going backwards. The key `t` is not available in that command.

```
Singularity
\Edacsac
  {elementary}
    {
      \Edacsac
        {non-degenerate elementary}
          {\ShortEdacsac{hyperbolic}{non-hyperbolic}}
        {degenerate elementary}
          {}
      }
  {non-elementary}
    {\ShortEdacsac{Nilpotent}{Higher order}}
```



5 Technical remark

The package cascade is designed to provide by default results similar to the those given by the environments of amsmath — and mathtools — especially `{aligned}`.

```
\[\left.\begin{aligned}
& A = \sqrt{a^2+b^2} \\
& B = \frac{ax+b}{cx+d}
\end{aligned}\right\}\]
```

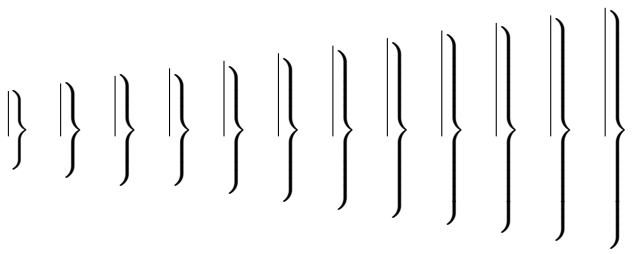
$$\left.\begin{aligned} A &= \sqrt{a^2+b^2} \\ B &= \frac{ax+b}{cx+d} \end{aligned}\right\}$$

```
\ShortCascade{$\displaystyle A = \sqrt{a^2+b^2}$}
             {$B = \dfrac{ax+b}{cx+d}$}
```

$$\left.\begin{aligned} A &= \sqrt{a^2+b^2} \\ B &= \dfrac{ax+b}{cx+d} \end{aligned}\right\}$$

The package cascade constructs the braces with the classical pair `\left-\right` of TeX. However, the extensible delimiters, in TeX, cannot take all sizes. We give, in the following example, the braces obtained when surrounding vertical rules from 6 mm to 17 mm (the code uses the L3 programming layer).

```
\int_step_inline:nnnn 6 1 {17} { $\left.\hbox{\vrule height #1 mm}\right\}$\quad }
```



6 Implementation

```
\RequirePackage{l3keys2e}
\ProvidesExplPackage
  {cascade}
  {\myfiledate}
  {\myfileversion}
  {Easy presentation of demonstrations in cascades}
```

`\spread@equation` We will use the command `\spread@equation` of amsmath to increase the interline in the commands `\Cascade`. When used, this command becomes no-op (in the current TeX group).
Nevertheless, we want the extension cascade available without amsmath. That's why we give a definition of `\spread@equation` (this definition will be loaded only if amsmath — or mathtools — has not been loaded yet).

```
\cs_if_free:NT \spread@equation
  {
    \cs_set_protected:Npn \spread@equation
      {
        \openup \jot
        \cs_set_protected:Npn \spread@equation { }
      }
  }
```

Don't put `\cs_set_eq:NN \spread@equation \prog_do_nothing:` in the last line because this would raise errors with nested environments.

The dimension `\l_@@_interline_dim` will be the value of the vertical space added between the two boxes connected by the brace.

```
\dim_new:N \l_@@_interline_dim
```

The dimension `\l_@@_interline_all_dim` is the default value of `\l_@@_interline_dim`. This default value can be modified with the option `interline-all`. Therefore, when modified in the options of a command `\Cascade`, this value will affect all the possible nested commands.

```
\dim_new:N \l_@@_interline_all_dim
```

The dimension `\l_@@_space_between_dim` is the horizontal space inserted between the two elements of the same row of the construction.

```
\dim_new:N \l_@@_space_between_dim
\dim_set:Nn \l_@@_space_between_dim { 0.5 em }
```

```
\bool_new:N \l_@@_t_bool
\bool_new:N \l_@@_main_command_bool
\bool_new:N \l_@@_nested_command_bool
\bool_new:N \l_@@_first_argument_bool
```

The set of keys `cascade/command` will be used by the command `\Cascade`.

```
\keys_define:nn { cascade / command }
  {
```

The key `t` means that the command `\Cascade` will be aligned upwards.

```
    t .code:n =
      \bool_if:NTF \l_@@_t_bool
        { \msg_error:nn { cascade } { t~option~already~set } }
        { \bool_set_true:N \l_@@_t_bool } ,
    t .value_forbidden:n = true ,
```

The option `interline` is the vertical space added between the two items connected by a brace.

```
    interline .dim_set:N = \l_@@_interline_dim,
    interline .value_required:n = true ,
```

The option `interline-all` will change the value of `interline` for all the commands `\Cascade`, even the nested commands.

```
    interline-all .code:n =
     {
       \dim_set:Nn \l_@@_interline_all_dim { #1 }
       \dim_set:Nn \l_@@_interline_dim { #1 }
     } ,
    interline-all .value_required:n = true ,
```

The option `space-between` is the horizontal space inserted between the two elements of the same row of the construction.

```
    space-between .dim_set:N = \l_@@_space_between_dim ,
    space-between .value_required:n = true
  }
```

The set of keys `cascade/global` will be used for the command `\CascadeOptions` (which fixes the options at a "global" level).

```
\keys_define:nn { cascade / global }
  {
    interline-all .dim_set:N = \l_@@_interline_all_dim ,
    interline-all .value_required:n = true ,
    space-between .dim_set:N = \l_@@_space_between_dim ,
    space-between .value_required:n = true
  }

\cs_new_protected:Npn \@@_initialisation:
  {
    \box_clear_new:N \l_@@_box_one
    \box_clear_new:N \l_@@_box_two
    \box_clear_new:N \l_@@_box_three
    \box_clear_new:N \l_@@_box_four
    \dim_zero_new:N \l_@@_top_dim
    \dim_zero_new:N \l_@@_bottom_dim
  }
```

`\CascadeOptions` The command `\CascadeOptions` is the command to set the options of the cascade at the document level (these options are set in a local way in the sense of the TeX groups).

```
\NewDocumentCommand \CascadeOptions { m }
  { \keys_set:nn { cascade / global } { #1 } }
```

`\Cascade` The command `\Cascade` is the main command of this package.

```
\NewDocumentCommand \Cascade { O { } m m m m D < > { } }
  {
    \if_mode_math:
      \msg_error:nn { cascade } { math~mode }
    \fi:
    \mode_leave_vertical:
```

The dimension `\g_@@_yoffset_dim` will be used by the option `t`.

```
    \bool_if:NF \l_@@_nested_command_bool
      {
        \dim_gzero_new:N \g_@@_yoffset_dim
        \bool_set_true:N \l_@@_first_argument_bool
      }
    \group_begin:

    \spread@equation
    \dim_set_eq:NN \l_@@_interline_dim \l_@@_interline_all_dim
    \keys_set:nn { cascade / command } { #1 }
    \tl_if_empty:nF { #6 }
      {
        \bool_if:NF \l_@@_t_bool
          { \msg_error:nn { cascade } { angular~argument~without~t } }
      }
    \@@_initialisation:
    \hbox_set:Nn \l_@@_box_one
      {
        \bool_set_true:N \l_@@_first_argument_bool
        \bool_set_true:N \l_@@_nested_command_bool
        #2
      }
    \hbox_set:Nn \l_@@_box_two { #3 }
    \hbox_set:Nn \l_@@_box_three
      {
        \bool_set_false:N \l_@@_first_argument_bool
        \bool_set_true:N \l_@@_nested_command_bool
        #4
      }
    \hbox_set:Nn \l_@@_box_four { #5 }
```

The dimension `\l_@@_top_dim` is the space that we will have to add before the main construction to make up for the "`\smash[t]`" of the box `#1`.

```
    \dim_set:Nn \l_@@_top_dim
      {
        \dim_max:nn
          \c_zero_dim
          { \box_ht:N \l_@@_box_one - \box_ht:N \l_@@_box_two }
      }
```

The dimension `\l_@@_bottom_dim` is the space that we will have to add after the main construction to make up for the "`\smash[b]`" of the box `#3`.

```
    \dim_set:Nn \l_@@_bottom_dim
      {
        \dim_max:nn
          \c_zero_dim
          { \box_dp:N \l_@@_box_three - \box_dp:N \l_@@_box_four }
```

```
      }
```

We do the "`\smash[t]`" of box `#1` and the "`\smash[b]`" of box `#3`.

```
    \box_set_ht:Nn \l_@@_box_one \c_zero_dim
    \box_set_dp:Nn \l_@@_box_three \c_zero_dim
```

We can now construct the box.

```
    \vbox_set:Nn \l_tmpa_box
      {
        \skip_vertical:N \l_@@_top_dim
        \vbox_top:n
          {
            \@@_the_vcenter:nn { #2 } { #4 }
```

We update `\g_@@_yoffset_dim`.

```
            \bool_if:NT \l_@@_first_argument_bool
              {
                \dim_set:Nn \l_tmpa_dim { \box_ht_plus_dp:N \l_tmpb_box }
                \l_tmpa_dim = 0.5\l_tmpa_dim
                \dim_add:Nn \l_tmpa_dim { \the \fontdimen 22 \textfont2 }
                \dim_sub:Nn \l_tmpa_dim
                  { \dim_max:nn { \box_ht:N \l_@@_box_two } { \box_ht:N \strutbox } }
                \dim_gadd:Nn \g_@@_yoffset_dim \l_tmpa_dim
              }
            \hbox
              {
                \c_math_toggle_token
                \left .
                \box_use_drop:N \l_tmpb_box
                \right \}
                \c_math_toggle_token
                \bool_if:NT \l_@@_t_bool
                  {
                     \bool_if:NF \l_@@_nested_command_bool
                       {
                         \tl_if_empty:nF { #6 }
                           {
                             \skip_horizontal:n \l_@@_space_between_dim
                             #6
                           }
                       }
                  }
              }
            \skip_vertical:N \l_@@_bottom_dim
          }
      }
    \bool_if:NTF \l_@@_nested_command_bool
      { \box_use_drop:N \l_tmpa_box }
      {
```

We are in the main command `\Cascade` and, if the option `t` is in force, we have now to take into account that key.

```
        \bool_if:NTF \l_@@_t_bool
          { \box_move_down:nn \g_@@_yoffset_dim { \box_use:N \l_tmpa_box } }
          { \box_use_drop:N \l_tmpa_box }
```

```
      }
    \group_end:
  }
```

The following macro is only for the legibility of the code.

```
\cs_new_protected:Npn \@@_the_vcenter:nn #1 #2
  {
    \hbox_set:Nn \l_tmpb_box
      {
        \c_math_toggle_token
        \vcenter
          {
            \halign
              {
                \hfil ## \cr
                \hbox
                  {
                    \tl_if_empty:nF { #1 }
                      {
                        \box_use_drop:N \l_@@_box_one
                        \skip_horizontal:n \l_@@_space_between_dim
                      }
                    \box_use:N \l_@@_box_two
                    \strut
                  }
                \cr
                \noalign { \skip_vertical:n \l_@@_interline_dim }
                \hbox
                  {
                    \tl_if_empty:nF { #2 }
                      {
                        \box_use_drop:N \l_@@_box_three
                        \skip_horizontal:n \l_@@_space_between_dim
                      }
                    \box_use_drop:N \l_@@_box_four
                    \strut
                  }
                \cr
              }
          }
        \c_math_toggle_token
      }
  }
```

The command `\Edacsac`. The code is simpler because we don't need the `\halign` and we don't have the key `t`.

```
\NewDocumentCommand \Edacsac { O { } m m m m }
  {
    \if_mode_math:
      \msg_error:nn { cascade } { math~mode }
    \fi:
    \mode_leave_vertical:
    \group_begin:
```

```
\spread@equation
\dim_set_eq:NN \l_@@_interline_dim \l_@@_interline_all_dim
\keys_set:nn { cascade / command } { #1 }
\@@_initialisation:
\hbox_set:Nn \l_@@_box_one { #2 }
\hbox_set:Nn \l_@@_box_two { #3 }
\hbox_set:Nn \l_@@_box_three { #4 }
\hbox_set:Nn \l_@@_box_four { #5 }
\dim_set:Nn \l_@@_top_dim
  {
    \dim_max:nn
      \c_zero_dim
      { \box_ht:N \l_@@_box_two - \box_ht:N \l_@@_box_one }
  }
\dim_set:Nn \l_@@_bottom_dim
  {
    \dim_max:nn
      \c_zero_dim
      { \box_dp:N \l_@@_box_four - \box_dp:N \l_@@_box_three }
  }
\box_set_ht:Nn \l_@@_box_two \c_zero_dim
\box_set_dp:Nn \l_@@_box_four \c_zero_dim
\vbox
   {
     \skip_vertical:N \l_@@_top_dim
     \vtop
       {
         \hbox
           {
             \c_math_toggle_token
             \left \{
             \vcenter
               {
                 \hbox
                   {
                     \tl_if_empty:nF { #2 }
                       {
                         \box_use_drop:N \l_@@_box_one
                         \skip_horizontal:n \l_@@_space_between_dim
                       }
                     \box_use_drop:N \l_@@_box_two
                     \strut
                   }
                 \skip_vertical:N \l_@@_interline_dim
                 \hbox
                   {
                     \tl_if_empty:nF { #4 }
                       {
                         \box_use_drop:N \l_@@_box_three
                         \skip_horizontal:n \l_@@_space_between_dim
                       }
                     \box_use_drop:N \l_@@_box_four
                     \strut
```

```
                        }
                    }
                  \right .
                  \c_math_toggle_token
                }
              \skip_vertical:N \l_@@_bottom_dim
            }
        }
    \group_end:
}

\msg_new:nnn
  { cascade }
  { math~mode }
  {
    The~commands~of~the~extension~'cascade'~
    should~be~used~in~text~mode~only.~However,~you~can~
    go~on~for~this~time.
  }
\msg_new:nnn
  { cascade }
  { t~option~already~set }
  {
    You~can't~use~the~key~'t'~here~because~it~has~been~set~
    in~an~encompassing~command.~If~you~go~on,~this~key~will~be~
    ignored.
  }
\msg_new:nnn { cascade } { angular~argument~without~t }
  {
    You~can't~use~the~argument~between~angular~brackets~because~
    you~have~not~used~the~key~'t'. The~argument~between~angular~brackets~
    will~be~ignored.
  }
```

`\ShortCascade` The command `\ShortCascade` is a simplified version of `\Cascade` with only two arguments.

```
\NewDocumentCommand \ShortCascade { O { } m m }
  { \Cascade [ #1 ] { } { #2 } { } { #3 } }
```

`\ShortEdacsac` Idem for `\ShortEdacsac`

```
\NewDocumentCommand \ShortEdacsac { O { } m m }
  { \Edacsac [ #1 ] { #2 } { } { #3 } { } }
```

7 History

Changes between versions 1.1 and 1.2

New commands `\Edacsac` and `\ShortEdacsac`.

Changes between versions 1.0 and 1.1

New option `t`.

Contents