schemata — Generic package to aid construction of topical categories*

Charles P. Schaum†

Released 2025/08/27

Abstract

The schemata package helps the creation of topical outlines that illustrate the breakdown of concepts and categories in academic texts from the late medieval to early modern periods.

Contents

1 Introduction

This package uses boxes and math mode to typeset *schemata* (plural of τό σχῆμα or *schema*, meaning *form*, *shape*, *appearance*, etc.). One sees them in academic literature from at least the seventeenth through the nineteenth centuries.[1]

Packages like *TikZ*, PSTricks, METAPOST, or other solutions have advantages over this one, especially for those seeking a top-to-bottom diagram.[2] Yet these packages may present challenges if one has to implement both open *and* closed braces in a schema, which math mode allows.

*This file describes version 1.5, last revised 2025/08/27.

†E-mail: charles[dot]schaum@comcast.net

[1]Books that use this package include: Löhe, *The Pastor* [*Der evangelische Geistliche*] (St. Louis, 2015) and Schaum and Collver, *Breath of God, Yet Work of Man* (St. Louis, 2019).

[2]For example: H. Dembowski, *Einführung in die Christologie* (Darmstadt, 1993), 146.

2 Usage

2.1 Package Loading and Options

The schemata package is a minimal "wrapper" for math mode. It can be used with LaTeX or with "generic" formats, including Plain TeX, eplain, and Lollipop. Currently, schemata will operate with ConTeXt without halting, but the way that it sizes and aligns delimiters does not work well with ConTeXt.

For LaTeX invoke:	`\usepackage[`⟨*options*⟩`]{schemata}`
For generic use:	`\input␣schemata.sty`

`\schemataLaTeX` Normally, schemata uses generic TeX macros if the format is not LaTeX 2_ε. When using a LaTeX-like format with a different name than `LaTeX2e`, one could insert the following before `\usepackage{schemata}`:

```
\edef\schemataLaTeX{\fmtname}
```

Yet this is usually unneeded. Normally we want `\schemataLaTeX` to be undefined before `schemata.sty` is loaded to get the default value `LaTeX2e`. We recommend not using this macro unless you know what you are doing.

options LaTeX users can choose one among four package options: `braces`, `brackets`, `parens`, and `groups`. These set the defaults for the delimiters used with `\schema` and `\Schema`. If no options are chosen, the default is `braces`.

2.2 Macro Overview

One can describe schemata as a grouping of boxes that contain content, whose relationships are demonstrated by delimiters. We start with the boxes and their content. Subsequently, we deal with the delimiters, then later, the manner of grouping and arrangement, as well as tweaks and tutorials.

One need not enclose schemata within any math mode delimiters but one can use, for example, display math when constructing large schemata. The `\schema` and `\Schema` macros will ensure that they use math mode. Only the large examples in this manual are contained in a math environment.

2.2.1 Containers: `\schemabox`

`\schemabox` Schemata contain vertically-centered lists of material in inner vertical mode. By design, it is *not* a `\long` macro. When in a `\schema` or a `\Schema` (see below), a `\schemabox` stacks one or more lines of `\hbox`-enclosed text in a `\vbox`. It redefines the macro `\\` to close the current `\hbox` and begin a new one. A `\schemabox` can add kerns before and after each line of text to aid horizontal spacing (Section 2.3).

`\schemabox[`⟨*width*⟩`]{`⟨*text*⟩`}`

The ⟨*width*⟩ of a `\schemabox` is a dimension, e.g., `3cm`. No text wrapping (as in a `\parbox`) takes place. If there is more than one line of text, each line of ⟨*text*⟩ must be terminated explicitly by `\\`, except the final line. Usually, the first line of a `\schemabox` inserts a `\strut`, but that can be modified (Section 2.3).

When not in internal vertical mode, `\schemabox` ignores ⟨*width*⟩, does not redefine the macro `\\`, and prints its argument as text to mitigate errors:

```
Some text has \schemabox{line 1\\ line 2}.
```

Some text has line 1
line 2.

2.2.2 Delimiters

`\DoBraces` `\DoBrackets` `\DoParens` `\DoGroups`

Both generic TEX and LATEX users can use these four macros to set or change the type of delimiters. In both generic TEX and LATEX, the default delimiter is braces. `\DoBraces`, `\DoBrackets`, `\DoParens`, and `\DoGroups` do the same thing as the respective package options, except they also can change the delimiters within `\schema` and `\Schema`. They remain in force until the end of a given scope:

Default assumptions include using braces, at least one uppercase letter on each side of a delimiter, and that delimiters open from left to right.

```
\schema
  {\schemabox{A}}
  {\schemabox{B\\C}}
```

$$A \left\{ \begin{matrix} B \\ C \end{matrix} \right.$$

Now we change to brackets (`\DoBrackets`). We add a "nudge" before B and C for better spacing (Section 2.3):

```
\DoBrackets
\schema
  {\schemabox{A}}
  {\NudgeSB[left]\schemabox{B\\C}}
```

$$A \left[\begin{matrix} B \\ C \end{matrix} \right.$$

Change to parentheses. We use `\schema` with a closing delimiter and add a "nudge" after A and B for better spacing:

```
\DoParens
\schema[close]
  {\NudgeSB\schemabox{A\\B}}
  {\schemabox{C}}
```

$$\left. \begin{matrix} A \\ B \end{matrix} \right) C$$

As we change to groups we use the more complex macro `\Schema` with a nested closing delimiter on the right-hand side, as well as a "nudge" after B and C:

```
\DoGroups
\Schema{0ex}{3ex}
  {\smallskip\schemabox{A}}
  {\Schema[close]{0ex}{3ex}
    {\NudgeSB\schemabox{B\\C}}
    {\smallskip\schemabox{D}}
}
```

$$A \left(\begin{matrix} B \\ C \end{matrix} \right) D$$

We change delimiters within the schema below. We insert "nudges" both before and after B and C, and before D. See also Section 2.5.

```
\DoBrackets
\Schema{0ex}{2.4ex}
  {\schemabox{A}}
  {\DoParens\Schema[close]{0ex}{2.4ex}
    {\NudgeSB[left]\NudgeSB[right]
     \schemabox{B\\C}}
    {\NudgeSB[left]\schemabox{D}}
  }
```

$$A \left[\begin{matrix} B \\ C \end{matrix} \right) D$$

One can add new delimiter types.[3] In PLAIN TEX one might see the following example. Both `\SwitchSB` and `\NudgeSB` are described in Section 2.3:

```
\catcode`@=11\relax
\def\DoVerts{%
  \let\@schemata@LD\Vert%
  \let\@schemata@RD\Vert}
\catcode`@=12\relax
\DoVerts
\schema{\SwitchSB\schemabox{A}}
  {\NudgeSB[left]\schemabox{B\\C\\D}}
```

$$\mathrm{A} \Vert \begin{matrix}\mathrm{B}\\ \mathrm{C}\\ \mathrm{D}\end{matrix}$$

2.2.3 Leaf Nodes: `\schema`

`\schema` A "simple" schema has a left-hand side with vertically-centered vertical material, a brace, and a right-hand side with vertically-centered vertical material:

`\schema[`⟨*type*⟩`]{`⟨*left side*⟩`}{`⟨*right side*⟩`}`

The ⟨*left side*⟩ and ⟨*right side*⟩ are vertical material in order to allow a `\smallskip` or other vertical adjustment as needed. The ⟨*type*⟩ of a schema is `open` by default. The delimiter opens toward the right:

```
\schema
  {\schemabox{A}}
  {\schemabox{B\\C}}
```

$$\mathrm{A} \left\{ \begin{matrix}\mathrm{B}\\ \mathrm{C}\end{matrix} \right.$$

Any value of ⟨*type*⟩ other than the string `open` makes a "closed" schema (the delimiter opens to the left). Especially with closed braces, one often adds a "nudge" to the left-hand side to negates a kern of `-0.2em` added by default.

```
\schema[closed]
  {\NudgeSB\schemabox{A}}
  {\schemabox{B\\C}}
```

$$\mathrm{A} \right\} \begin{matrix}\mathrm{B}\\ \mathrm{C}\end{matrix}$$

In practice, `\schema` does not nest, so it is only useful for the rightmost elements or "leaves" of a large schema to make formatting the leaves more efficient. Using boxes, below we show where one would use `\schema` to populate the leaves:

```
\Schema{-0.4ex}{4.2ex}
  {\schemabox{A}}
  {\vbox{\noindent%
    \fbox{\schema
      {\schemabox{B}}
      {\schemabox{C\\D}}}\\[1ex]
    \fbox{\schema
      {\schemabox{E}}
      {\schemabox{F\\G}}}}}
```

$$\mathrm{A} \left\{ \begin{array}{|l} \hline \mathrm{B} \left\{ \begin{matrix}\mathrm{C}\\ \mathrm{D}\end{matrix} \right. \\ \hline \end{array} \right.$$
$$\left. \begin{array}{|l} \hline \mathrm{E} \left\{ \begin{matrix}\mathrm{F}\\ \mathrm{G}\end{matrix} \right. \\ \hline \end{array} \right.$$

The sizing of a `\schema` changes automatically, depending on the height, depth, and context of the letters. This can look ugly if uniform delimiter size is desired. Use `\Schema` (next section) to enforce such uniformity.

[3] See *The TEXbook*, page 146. Avoid using `\langle`, `\rangle`, `/`, and `\textbackslash`. They do not scale like the other delimiters and will not produce expected results in some cases.

2.2.4 Branches and Root: \Schema

\Schema The "complex" form of a schema also has a left-hand side with vertically-centered vertical material, a brace, and a right-hand side of vertically-centered vertical material, along with two arguments that adjust the layout:

\Schema[⟨*type*⟩]{⟨*adjust*⟩}{⟨*size*⟩}{⟨*left side*⟩}{⟨*right side*⟩}

The ⟨*type*⟩ is `open` by default. As above, any other ⟨*type*⟩ except the exact string `open` will make it a "closed" schema. Both ⟨*adjust*⟩ and ⟨*size*⟩ are dimensions. We recommend expressing them as `ex`. This allows for easier scaling of the schema when changing the font size. Here is how to set ⟨*adjust*⟩:[4]

negative	left side and delimiter up	right side down
positive	left side and delimiter down	right side up

Set the delimiter ⟨*size*⟩ to be a scaled value of ex just a bit larger than the number of lines of text that the delimiter spans.

Adjustments must be done from right to left. Once a leaf or sub-tree is balanced, any attempt to re-balance the schema from left to right will throw everything off. We will return to this point especially in Section 2.4.2 and thereafter.

By using `\Schema` to adjust the delimiter height and centering, one can bypass the shortcomings of `\schema`, but at the cost of time. One has to traverse the schema at least twice to get the desired layout.

`\Schema` lets one produce multiple schemata with the same look, i.e., the same brace heights and general appearance. One can nest and combine these similar schemata into complex layouts. See Section 2.4.5.

Both `\schema` and `\Schema` can be in running text:

Here is one A $\left\{\begin{array}{l}\text{B}\\\text{C}\end{array}\right.$ and the other D $\left\{\begin{array}{l}\text{E}\\\text{F}\end{array}\right.$

```
1  Both \cmd{\schema} and \cmd{\Schema} can be in running text:\\[1ex]
2  \indent Here is one \schema{\schemabox{A}}{\schemabox{B\\C}}
3  and the other \Schema{0ex}{2.5ex}{\schemabox{D}}{\schemabox{E\\F}}
```

Both `\schema` and `\Schema` will stack vertically if set sequentially. It does not matter which type of schema precedes or follows; for example:

```
1  \schema
2    {\schemabox{A}}
3    {\schemabox{B\\C}}
4
5  \Schema{0ex}{2.5ex}
6    {\schemabox{D}}
7    {\schemabox{E\\F}}
```

$$\text{A}\left\{\begin{array}{l}\text{B}\\\text{C}\end{array}\right.$$

$$\text{D}\left\{\begin{array}{l}\text{E}\\\text{F}\end{array}\right.$$

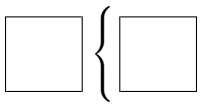
[4] Instead of setting ⟨*adjust*⟩, one could put vertical skips either before or after `\schemabox`, `\schema`, or `\Schema`. Yet using braces as delimiters tends to draw material toward the center cusp (if applicable), where ⟨*adjust*⟩ keeps that centered look while allowing some adjustments.

Certainly, one need not use a \schemabox in either \schema or \Schema. For example, we make a macro \Box below to create one square centimeter of content:

```
\def\Box{%
  \hbox{%
    \vrule%
    \vbox to 1cm{\hrule\hbox to 1cm{\hfil}\vfil\hrule}%
    \vrule%
  }%
}
```

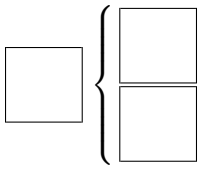
Now we begin with the trivial example of one \Box on each side of the delimiter:

```
\schema{\Box}{\Box}
```



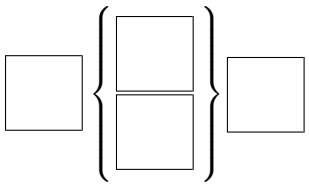
This example is more complex, showing how each side stacks boxes vertically:

```
\schema{\Box}{\Box\Box}
```



Finally we use \Schema to get a schema that is both open and closed:

```
\Schema{-0.2ex}{0.9cm}
{\Box}
{
  \Schema[close]
    {-0.2ex}{0.9cm}
  {\Box\hbox{\Box\kern0.2em}}
  {\Box}
}
```



A kern of 0.2em was added above to compensate for the automatic kern of -0.2em. This correction mimics the behavior of \NudgeSB.

If not expressed in terms of ex height, which is quite useful for correlating the number of lines of text and white space in a schema to the actual height of a brace that is needed, then ⟨*size*⟩ should be expressed as above: slightly less than half the height of the contents, e.g., 0.9cm for a height of 2cm.

2.3 Romancing the \schema

\LCschema
\UCschema

By default, a \schemabox adds a \strut to the first line because the topics in a schema often start with a capital letter. The \strut causes the delimiter of a \schema to have the proper size in order to span that letter.

If the first letter is not a capital letter, or if the text seems a little off-center, you can turn off this default feature of \schemabox by placing \LCschema before the \schemabox where the change is to occur. \LCschema prevents all subsequent uses of \schemabox from adding \strut. One restores the default behavior with \UCschema, also best placed before the intended \schemabox.

Here is an example where an entire schema is in lowercase. Instead of putting `\LCschema` before a particular `\schemabox`, we change the look of the whole thing and then restore the default via `\UCschema`:[5]

```
\LCschema
\Schema{0.1ex}{4.8ex}
{\schemabox{sensus literalis}}
{\schema
  {\schemabox{sensus\\literalis\\(improprie)}}
  {\schemabox{e parallelismo clarior\\
     ex analogia fidei\\ex evidentia rei}}
  \smallskip
  \schemabox{sensus literae}
}
\UCschema
```

$$\text{sensus literalis} \begin{cases} \text{sensus} \\ \text{literalis} \\ \text{(improprie)} \end{cases} \begin{cases} \text{e parallelismo clarior} \\ \text{ex analogia fidei} \\ \text{ex evidentia rei} \end{cases} \\ \text{sensus literae}$$

`\SwitchSB` The macro `\SwitchSB` toggles the insertion of a `\strut` to the opposite of what is in effect for just one `\schemabox`. Thus:

Current State	Using `\SwitchSB`
`\UCschema`	no `\strut`
`\LCschema`	insert a `\strut`

`\SwitchSB` should be placed immediately before the `\schemabox` to be affected. Its effects are reset when that particular `\schemabox` terminates.

Note, however, that mixing lowercase and uppercase-styles of `\schemabox` may put parts of a schema slightly off-center, meaning that one must ⟨*adjust*⟩ a `\Schema` by a tenth of an ex, give or take. Also remember that one can add `\strut` as needed to make manual adjustments.

`\NudgeSB` The macro `\NudgeSB` is another "per-use" macro that, by default, causes a particular `\schemabox` to add either one or two `0.2em` kerns to the end(s) of every line of text. This behavior resets outside of that `\schemabox`.

`\NudgeSB[`⟨*direction*⟩`]`

The default argument of ⟨*direction*⟩ is `right`; the kern is added to the right side of the text. Any other value for ⟨*direction*⟩ causes a `\schemabox` to add a kern to the left side. One can add kerns to both sides of every line of text, e.g.:

```
\NudgeSB[left]\NudgeSB[right]\schemabox{text}
```

Both `\schema` and `\Schema` insert a kern of `-0.2em` between the left-hand side and the delimiter to correct spacing when a delimiter opens to the right, especially braces. In some cases, we bypass the default with a nudge.

[5] Based on axioms in August Pfeiffer, *Thesaurus Hermeneuticus* (Frankfurt am Main, 1698).

`\SBNudgeFactor` This macro is the kern used by `\NudgeSB` to make its corrective. Sometimes you feel like a nudge, sometimes you don't, and sometimes you want to change the size of a nudge. Next we show how to do that within a local scope:

```
\bgroup
  \DoBrackets
  \renewcommand\SBNudgeFactor{\kern 1em}
  \Schema{0ex}{2.3ex}
  {
    \NudgeSB
    \schemabox{a}
  }
  {
    \Schema[close]{0ex}{2.4ex}
    {
      \NudgeSB[right]
      \NudgeSB[left]
      \schemabox{b\\c}
    }
    {
      \NudgeSB[left]
      \schemabox{d}
    }
  }
\egroup
```

$$a \left[\begin{array}{c} b \\ c \end{array} \right] d$$

2.4 Tutorial

Now that we have explained what the macros do, let's take a journey together to discover a methodology for creating general forms of schemata.

2.4.1 Starting Off Basic

Let's ignore most of what we have learned so far and naively try `\schema`:

```
\schema{A}{B\\C}
```

$$A \left\{ \begin{array}{l} B \\ C \end{array} \right.$$

Oh dear, that went badly. Oh, wait! Schemata hold internal vertical lists. That weird `\schemabox` is designed for internal vertical mode:

```
\schema
  {\schemabox{A}}
  {\schemabox{B\\C}}
```

$$A \left\{ \begin{array}{l} B \\ C \end{array} \right.$$

Now we are getting somewhere! But we really need a "big" side of a schema, or else we get:

```
\schema
  {\schemabox{A}}
  {\schemabox{B}}
```

$$A \left\{ B \right.$$

2.4.2 *Loci* 101

We move on from trivial examples to several real-world examples based on published material. Let's try a few examples from *Loci Theologici* by Martin Chemnitz. We begin by using only `\schema`:

```
\schema
{
  \schemabox{Subjectum theo-\\
    logi\ae{} est Notitia\\
    Dei. Considerat\\
    ergo, Dei, vel}
}
{
  \schema
  {
    \schemabox{\textsc{Essentiam},}
  }
  {
    \schemabox{Unitate natur\ae{}.\\
    Trinitate personarum.\\
    Operibus ad intra.}
  }
  \schema
  {
    \schemabox{\textsc{Voluntatem},\\
    manifestatam in\\
    operibus ad extra;\\
    ut in}
  }
  {
    \schemabox{Creatione.\\
    Sustentatione natur\ae{} laps\ae{}.\\
    Reparatione.\\
    Conversione.\\
    Justificatione.\\
    Sanctificatione \&\\
    Glorificatione ejusdem.}
  }
}
```

Subjectum theologiæ est Notitia Dei. Considerat ergo, Dei, vel

- Essentiam,
  - Unitate naturæ.
  - Trinitate personarum.
  - Operibus ad intra.
- Voluntatem, manifestatam in operibus ad extra; ut in
  - Creatione.
  - Sustentatione naturæ lapsæ.
  - Reparatione.
  - Conversione.
  - Justificatione.
  - Sanctificatione &
  - Glorificatione ejusdem.

This is not what we want to see; `\schema` works for the leaves on the right, but not for the "root" on the left. The brace adjusts to the entire right-hand side. Below, we demonstrate what we want and then show how we got there:

Subjectum theologiæ est Notitia Dei. Considerat ergo, Dei, vel
- Essentiam,
  - Unitate naturæ.
  - Trinitate personarum.
  - Operibus ad intra.
- Voluntatem, manifestatam in operibus ad extra; ut in
  - Creatione.
  - Sustentatione naturæ lapsæ.
  - Reparatione.
  - Conversione.
  - Justificatione.
  - Sanctificatione &
  - Glorificatione ejusdem.

1. We first adjust the spacing, starting from the leaves at right, going to the root on the left. We add a `\smallskip`, shown in **red boldface**:

```
13    {
14      \schemabox{Unitate natur\ae{}.\\
15      Trinitate personarum.\\
16      Operibus ad intra.}
17    }\smallskip
```

   One should be attentive to format-dependent macro behavior.[6] In some cases, putting vertical space in the first or last lines of a `\schemabox`, regardless of format, will affect centering negatively.

2. We have two `\schema` leaves and one root, so we only change one `\schema` into a `\Schema`. We count the lines of text, estimate, then revise.

   Below we have 8–9 lines of text from "Essentiam" to "ut in." We estimate ⟨*size*⟩ at `8.5ex` and ⟨*adjust*⟩ at `0ex`. The large brace is too low, so we ⟨*adjust*⟩ to `-1ex`, raising the left side and the delimiter, while lowering the right. We then refine ⟨*size*⟩ to `8.7ex`.[7]

```
1   \Schema{-1ex}{8.7ex}
```

After those two changes, we have the finished schema:

```
1   \Schema{-1ex}{8.7ex}
2   {
3     \schemabox{Subjectum theo-\\
4     logi\ae{} est Notitia\\
5     Dei. Considerat\\
6     ergo, Dei, vel}
7   }
```

[6] Using `\vskip` in Plain TeX starts a new paragraph, so `\smallskip` cannot be used within the horizontal mode of a `\schemabox` when using Plain TeX. LaTeX allows more flexibility and permits one to place vertical skips more freely.

[7] Changes in TeX distributions can change font metrics and thus, the metrics of your schemata.

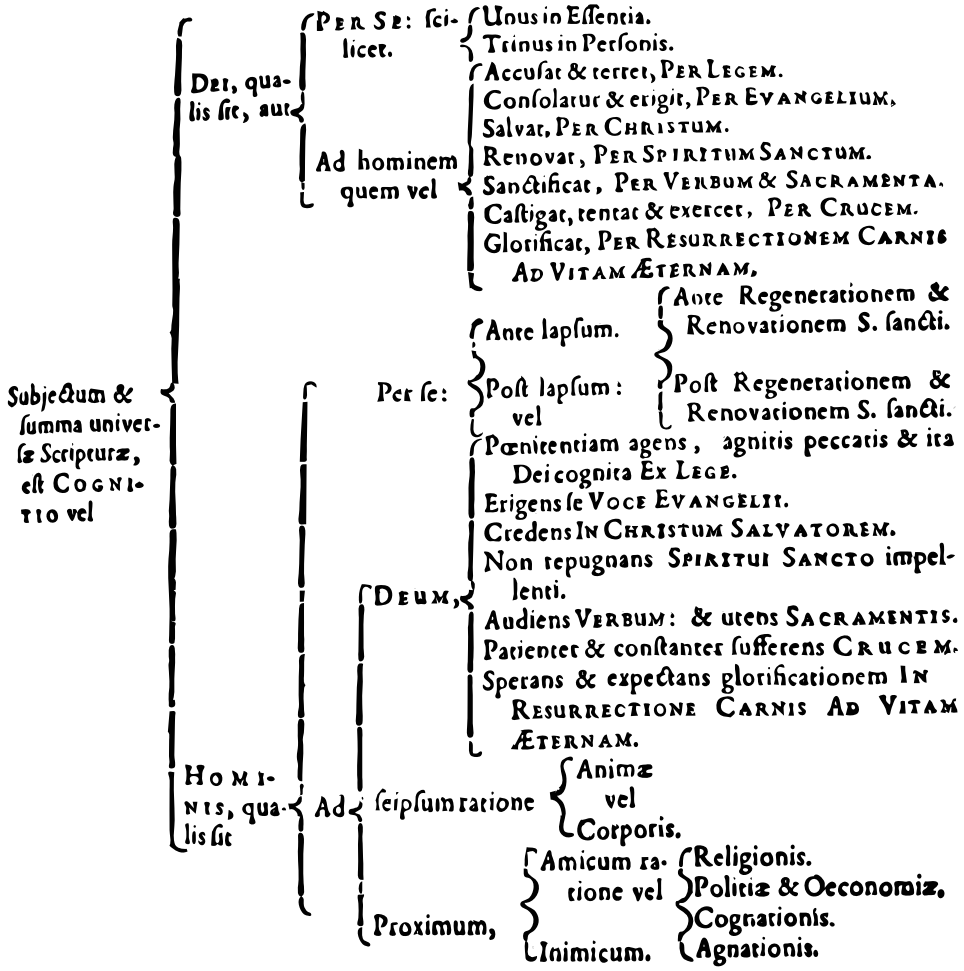
```
{
\schema
  {
    \schemabox{\textsc{Essentiam},}
  }
  {
    \schemabox{Unitate natur\ae{}.\\
    Trinitate personarum.\\
    Operibus ad intra.}
  }\smallskip
  \schema
  {
    \schemabox{\textsc{Voluntatem},\\
    manifestatam in\\
    operibus ad extra;\\
    ut in}
  }
  {
    \schemabox{Creatione.\\
    Sustentatione natur\ae{} %
    laps\ae{}.\\
    Reparatione.\\
    Conversione.\\
    Justificatione.\\
    Sanctificatione \&\\
    Glorificatione ejusdem.}
  }
}
```

2.4.3 Going Big

Thus far, we have dealt with trivial examples while gaining knowledge:

1. We usually use `\schemabox` for the contents of a schema.
2. Schemata usually open from left to right, from root to leaves.
3. We usually typeset leaves with `\schema` to save time.
4. We typeset other parts with `\Schema`.
5. We adjust spacing and delimiters by working from the leaves to the root.
6. We may need to consider differences among formats when using `\vskip`, `\smallskip`, etc. Also, `\newbox` is an `\outer` macro in PLAIN TEX.
7. Sometimes, we need to use `\UCschema`, `\LCschema`, `\SwitchSB`, and `\NudgeSB`.

With this information, we will reproduce a schema found on page 13 of Martin Chemnitz, *Loci Theologici* (Frankfurt, 1653). The image of the schema was obtained by the present author.



- The braces are composed of vertical rules and other type sorts. We will replace them with the analogous standard TEX brace delimiters.
- The Latin uses roman, italic and small caps. We will use *s-finalis* where the original uses *s-medialis*. We will retain some old-style ligatures.
- We will improve spacing between elements and we will not aim for an exact reproduction of line breaks.

We begin by looking at the leaves, the rightmost bits of text enclosed by braces. We can use `\schema` in these cases. That results in the following individual snippets:

The first snippet starts at the top right of the original schema:

```
\schema
{\schemabox{\textsc{Per se}:\\ scilicet.}}
{
  \schemabox{Unus in essentia.}
  \schemabox{Trinus in personis.}
}
```

PER SE: scilicet. { Unus in essentia. / Trinus in personis.

Next is the first large `\schema`, easy to confuse with the other large one:

```
\schema
{\schemabox{Ad hominem\\ quem vel}}
{
  \schemabox{Accusat \& terret, \textsc{Per Legem},\\
  Consolatur \& erigit, \textsc{Per Evangelium}.\\
  Salvat, \textsc{Per Christum}.\\
  Renovat, \textsc{Per Spiritum Sanctum}.\\
  Sanctificat, \textsc{Per Verbum} \& \textsc{Sacramenta}.\\
  Castigat, tentat \& exercet, \textsc{Per Crucem}.\\
  Glorificat \textsc{Per Resurrectionem Carnis}\\
  \textsc{\quad Ad Vitam \AE{}ternam}.}
}
```

Ad hominem quem vel

- Accusat & terret, Per Legem,
- Consolatur & erigit, Per Evangelium.
- Salvat, Per Christum.
- Renovat, Per Spiritum Sanctum.
- Sanctificat, Per Verbum & Sacramenta.
- Castigat, tentat & exercet, Per Crucem.
- Glorificat Per Resurrectionem Carnis Ad Vitam Æternam.

Next we have just one line of text, which is almost tucked in front of the leaf that comes after it, but it still is a leaf in its own right:

```
\schemabox{Ante lapsum.}
```

Ante lapsum.

This next example looks bad, but structure precedes aesthetics. Later we will add vertical skips to get two groups of two lines each:

```
\schema
{\schemabox{Post lapsum:}}
{
  \schemabox{Ante Regenerationem \&\\
  Renovationem S. Sancti.}
  \schemabox{Post Regenerationem \&\\
  Renovationem S. Sancti.}
}
```

Post lapsum:

- Ante Regenerationem &
- Renovationem S. Sancti.
- Post Regenerationem &
- Renovationem S. Sancti.

Here is the second large `\schema`, easy to confuse with the first:

```
\schema
{\schemabox{\textsc{Deum},}}
{
  \schemabox{P\oe{}nitentia agens, agnitis peccatis \&\\
  ira Dei cognita \textsc{Ex Lege}.\\
  Erigens se \textsc{Voce Evangelii}.\\
  Credens \textsc{In Christum Salvatorem}.\\
  Non repugnans \textsc{Spiritui Sancto} impellenti.\\
  Audiens \textsc{Verbum}: \& utens \textsc{Sacramentis}.\\
  Patienter \& constanter sufferens \textsc{Crucem}.\\
  Sperans \& expectans glorificationem\\
  \textsc{\quad In Resurrectione Carnis}\\
  \textsc{\quad Ad Vitam \AE{}ternam}.}
}
```

Deum, { Pœnitentia agens, agnitis peccatis &
ira Dei cognita Ex Lege.
Erigens se Voce Evangelii.
Credens In Christum Salvatorem.
Non repugnans Spiritui Sancto impellenti.
Audiens Verbum: & utens Sacramentis.
Patienter & constanter sufferens Crucem.
Sperans & expectans glorificationem
In Resurrectione Carnis
Ad Vitam Æternam.

Using one `\schemabox` on the right can do a decent job with spacing:

```
\schema
{\schemabox{seipsum ratione}}
{\schemabox{Anim\ae{}\\ vel\\ Corporis}}
```

seipsum ratione { Animæ
vel
Corporis

Here we have a large leaf, later followed by another "one-liner":

```
\schema
{\schemabox{Amicum ra-\\ tione vel}}
 {
  \schemabox{Religionis.\\
  Politic\ae{} \& \OE{}conomic\ae{}.\\
  Cognationis.\\
  Agnationis.}
}
```

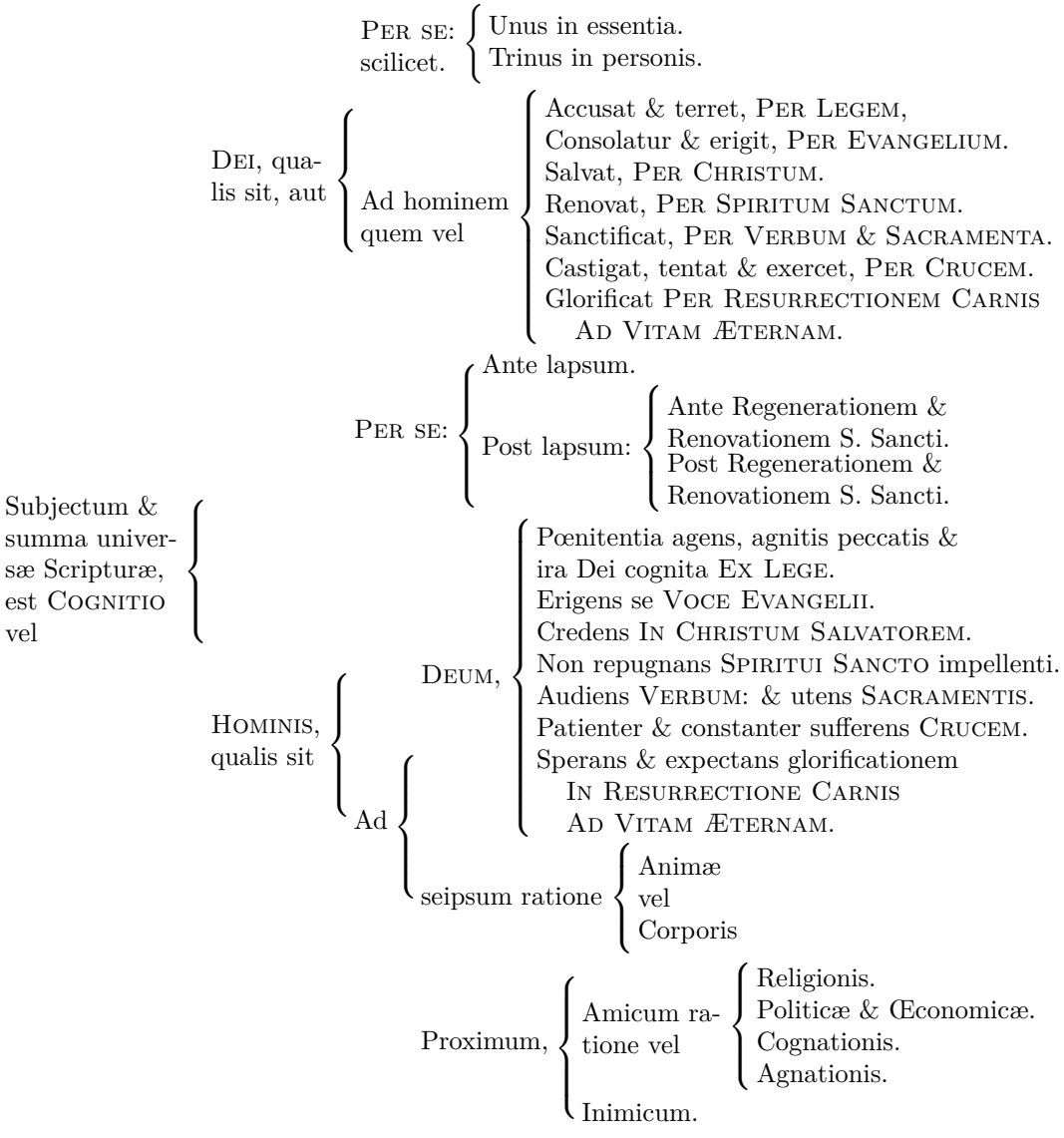
Amicum ratione vel { Religionis.
Politicæ & Œconomicæ.
Cognationis.
Agnationis.

And here is the second “one-liner” in the larger schema:

```
\schemabox{Inimicum.}
```

Inimicum.

We build the leaves into the larger schema using `\Schema` with dummy values for both ⟨*adjust*⟩ and ⟨*size*⟩ in order to verify that the basic design is accurate:



Although the initial result is not pleasing to the eye, the following code listing illustrates the structure of the schema above and serves as a basis for later revisions.

```
\Schema{0ex}{5ex}
{
  \schemabox{Subjectum \&\\
  summa univer-\\
  s\ae{} Scriptur\ae{},\\
  est \textsc{Cognitio}\\
  vel}
}
```

```
{
  \Schema{0ex}{5ex}
  {
    \schemabox{\textsc{Dei}, qua-\\lis sit, aut}
  }
  {
    \schema
    {\schemabox{\textsc{Per se}:\\ scilicet.}}
    {
      \schemabox{Unus in essentia.}
      \schemabox{Trinus in personis.}
    }
    \schema
    {\schemabox{Ad hominem\\ quem vel}}
    {
      \schemabox{Accusat \& terret, \textsc{Per Legem},\\
      Consolatur \& erigit, \textsc{Per Evangelium}.\\
      Salvat, \textsc{Per Christum}.\\
      Renovat, \textsc{Per Spiritum Sanctum}.\\
      Sanctificat, \textsc{Per Verbum} \& \textsc{Sacramenta}.\\
      Castigat, tentat \& exercet, \textsc{Per Crucem}.\\
      Glorificat \textsc{Per Resurrectionem Carnis}\\
      \textsc{\quad Ad Vitam \AE{}ternam}.}
    }
  }
  \Schema{0ex}{5ex}
  {
    \schemabox{\textsc{Hominis},\\ qualis sit}
  }
  {
    \Schema{0ex}{5ex}
    {\schemabox{\textsc{Per se}:}}
    {
      \schemabox{Ante lapsum.}
      \schema
      {\schemabox{Post lapsum:}}
      {
        \schemabox{Ante Regenerationem \&\\
        Renovationem S. Sancti.}
        \schemabox{Post Regenerationem \&\\
        Renovationem S. Sancti.}
      }
    }
    \Schema{0ex}{5ex}
    {\schemabox{Ad}}
    {
      \schema
      {\schemabox{\textsc{Deum},}}
      {
        \schemabox{P\oe{}nitentia agens, agnitis peccatis \&\\
        ira Dei cognita \textsc{Ex Lege}.\\
        Erigens se \textsc{Voce Evangelii}.\\
        Credens \textsc{In Christum Salvatorem}.\\
        Non repugnans \textsc{Spiritui Sancto} impellenti.\\
        Audiens \textsc{Verbum}: \& utens \textsc{Sacramentis}.\\
        Patienter \& constanter sufferens \textsc{Crucem}.\\
```

```
      Sperans \& expectans glorificationem\\
      \textsc{\quad In Resurrectione Carnis}\\
      \textsc{\quad Ad Vitam \AE{}ternam}.}
    }
    \schema
      {\schemabox{seipsum ratione}}
      {\schemabox{Anim\ae{}\\ vel\\ Corporis}}
    \Schema{0ex}{5ex}
    {\schemabox{Proximum,}}
    {
      \schema
      {\schemabox{Amicum ra-\\ tione vel}}
      {
        \schemabox{Religionis.\\
        Politic\ae{} \& \OE{}conomic\ae{}.\\
        Cognationis.\\
        Agnationis.}
      }
      \schemabox{Inimicum.}
    }
  }
}
```

Now we show how to correct the spacing within and among elements. **Here we must work from leaves to root.** Any other approach would waste time repeatedly adjusting the leaves and branches. We show those lines that were changed in **red boldface**, with some surrounding context. Remember that you can add a `\smallskip` within a `\schemabox` in LaTeX, but not in Plain TeX.

We have split the text below into two boxes to make it format-agnostic. The first `\smallskip` adds space between two instances of `\schemabox`. The second `\smallskip` follows the closing brace of the right-hand side, not the `\schemabox`, in order to adjust the entire `\schema`.

```
    \schema
    {\schemabox{\textsc{Per se}:\\ scilicet.}}
    {
      \schemabox{Unus in essentia.}\smallskip
      \schemabox{Trinus in personis.}
    }\smallskip
```

Again, the skip comes after the close of a `\schema` to adjust the whole thing.

```
      Castigat, tentat \& exercet, \textsc{Per Crucem}.\\
      Glorificat \textsc{Per Resurrectionem Carnis}\\
      \textsc{\quad Ad Vitam \AE{}ternam}.}
    }\medskip
```

Below, the first `\smallskip` helps to separate the lone `\schemabox` from the `\schema` beneath it. The result allows us to "tuck" that small element as its own leaf in front of and above the larger leaf that follows.

This also illustrates how the internal vertical lists contained by `\Schema` and `\schema` can hold heterogeneous material, which one can arrange as needed.

A \medskip is placed between two instances of \schemabox, which slightly throws off the way that the brace spans the boxes. A small skip is put at the end of the last \schemabox to correct that.

In simple cases like this particular leaf, we can offset one change in spacing with another change to avoid using \Schema. With anything more complex than that, however, using \Schema becomes inevitable.

Finally, a \smallskip is added after the entire \schema on the right-hand side.

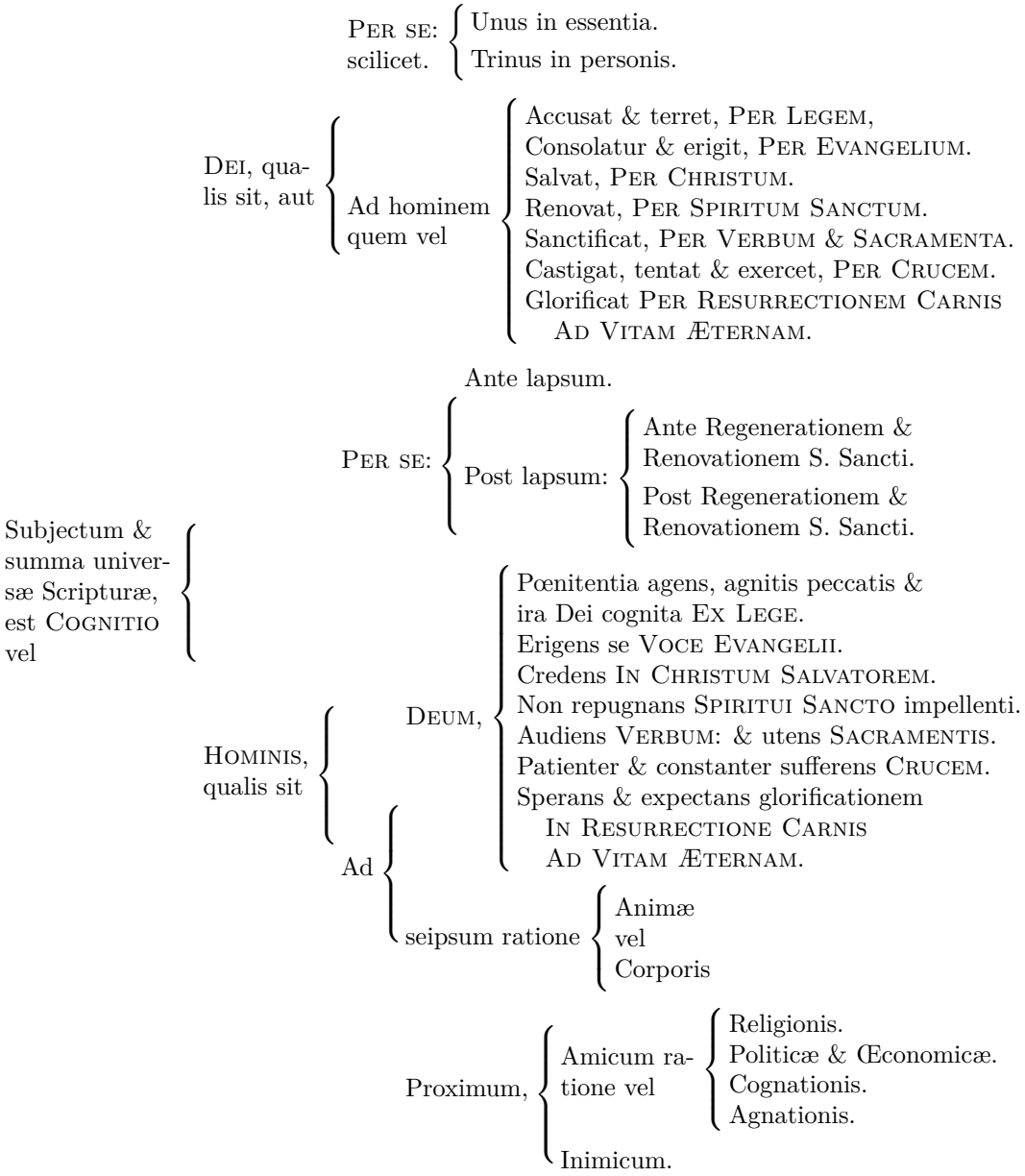
```
39      \Schema{0ex}{5ex}
40      {\schemabox{\textsc{Per se}:}}
41      {
42        \schemabox{Ante lapsum.}\smallskip
43        \schema
44        {\schemabox{Post lapsum:}}
45        {
46          \schemabox{Ante Regenerationem \&\\
47          Renovationem S. Sancti.}\medskip
48          \schemabox{Post Regenerationem \&\\
49          Renovationem S. Sancti.}\smallskip
50        }\smallskip
51      }
```

The skips below follow predictably at the end of each \schema:

```
65          Sperans \& expectans glorificationem\\
66          \textsc{\quad In Resurrectione Carnis}\\
67          \textsc{\quad Ad Vitam \AE{}ternam}.}
68        }\smallskip
69        \schema
70          {\schemabox{seipsum ratione}}
71          {\schemabox{Anim\ae{}\\ vel\\ Corporis}}\smallskip
72        \Schema{0ex}{5ex}
73        {\schemabox{Proximum,}}
74        {
75          \schema
76          {\schemabox{Amicum ra-\\ tione vel}}
77          {
78            \schemabox{Religionis.\\
79            Politic\ae{} \& \OE{}conomic\ae{}.\\
80            Cognationis.\\
81            Agnationis.}
82          }\smallskip
```

Notice in the following schema how the addition of vertical space among the various elements helps make the entire schema easier to read. This is a big improvement over the original version done with a letterpress!

Thus we have come to a rough halfway point. We have identified the various smaller elements of the schema, put them together in a larger structure, and given them the spacing needed to be aesthetically pleasing.



Next we estimate lines of text and blank lines between the top and bottom elements on the right-hand side of each `\Schema`. We put that number as ex height in ⟨*size*⟩. We can use the listing itself for estimating.

- From "Dei, qualis sit, aut" ("Salvat" at right) to "Hominis, qualis sit" ("Sperans" at right), put in the `\Schema` with "Subjectum & summa…".

```
\Schema{0ex}{23ex}
```

- From "Per se: scilicet" ("Unus" at right) to "Ad hominem quem vel" ("Sanctificat" at right), put in the `\Schema` with "Dei, qualis sit, aut".

```
  \Schema{0ex}{8ex}
```

- From “Per se” (“Renovationem” at right) to “Ad” (“ad Vitam” at right), put in the `\Schema` with “Ad hominem quem vel”.

```
34    \Schema{0ex}{16ex}
```

- From “Ante lapsum” to “Post lapsum”, put in the `\Schema` with “Per se”.

```
39      \Schema{0ex}{5ex}
```

- From “Deum” (“Non repugnans” at right) to “Proximum” (“Cognationis” at right), put in the `\Schema` with “Ad”.

```
52      \Schema{0ex}{16ex}
```

- From “Amicum ratione vel” (“Politicæ” at right) to “Inimicum”, put in the `\Schema` with “Proximum”.

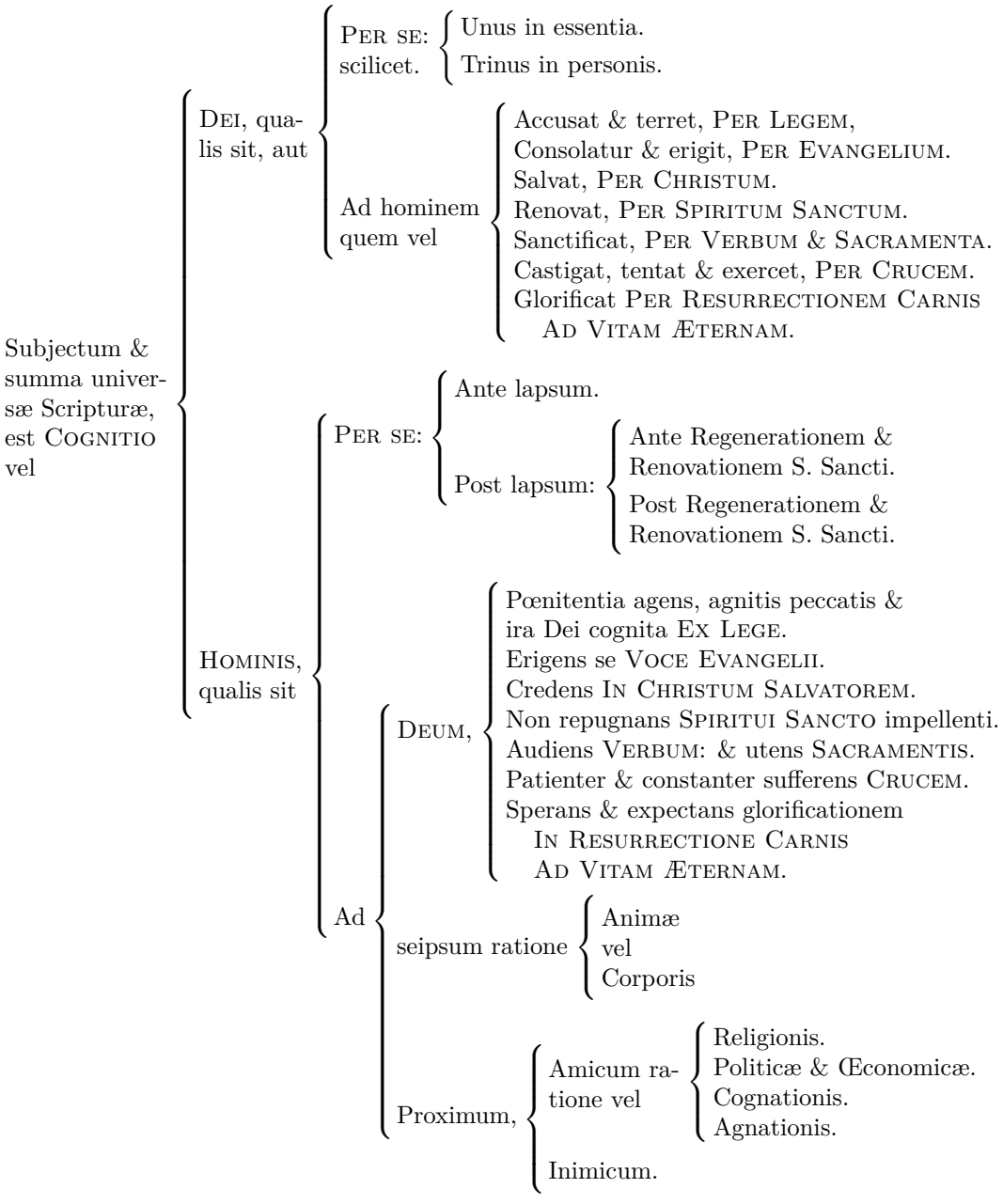
```
72        \Schema{0ex}{5ex}
```

The following schema illustrates our “ball park” figures:

- Subjectum & summa universæ Scripturæ, est Cognitio vel
  - Dei, qualis sit, aut
    - Per se: scilicet.
      - Unus in essentia.
      - Trinus in personis.
    - Ad hominem quem vel
      - Accusat & terret, Per Legem,
      - Consolatur & erigit, Per Evangelium.
      - Salvat, Per Christum.
      - Renovat, Per Spiritum Sanctum.
      - Sanctificat, Per Verbum & Sacramenta.
      - Castigat, tentat & exercet, Per Crucem.
      - Glorificat Per Resurrectionem Carnis Ad Vitam Æternam.
  - Hominis, qualis sit
    - Per se:
      - Ante lapsum.
      - Post lapsum:
        - Ante Regenerationem & Renovationem S. Sancti.
        - Post Regenerationem & Renovationem S. Sancti.
    - Ad
      - Deum,
        - Pœnitentia agens, agnitis peccatis & ira Dei cognita Ex Lege.
        - Erigens se Voce Evangelii.
        - Credens In Christum Salvatorem.
        - Non repugnans Spiritui Sancto impellenti.
        - Audiens Verbum: & utens Sacramentis.
        - Patienter & constanter sufferens Crucem.
        - Sperans & expectans glorificationem In Resurrectione Carnis Ad Vitam Æternam.
      - seipsum ratione
        - Animæ
        - vel
        - Corporis
      - Proximum,
        - Amicum ratione vel
          - Religionis.
          - Politicæ & Œconomicæ.
          - Cognationis.
          - Agnationis.
        - Inimicum.

Finally we tweak ⟨*adjust*⟩ values by counting the lines (`ex`) in the direction the left side needs to move relative to the right, multiply the result by two, and make it negative for up and positive for down. We also adjust the final ⟨*size*⟩ of the braces.

```
\Schema{-25ex}{20.6ex}
  \Schema{-6.4ex}{8.5ex}
  \Schema{-13.4ex}{17.4ex}
    \Schema{-4.4ex}{5ex}
    \Schema{4.2ex}{14.4ex}
      \Schema{2ex}{5.1ex}
```



Here is the final listing for our schema:

```
\Schema{-25ex}{20.6ex}
{
  \schemabox{Subjectum \&\\
  summa univer-\\
  s\ae{} Scriptur\ae{},\\
  est \textsc{Cognitio}\\
  vel}
}
{
  \Schema{-6.4ex}{8.5ex}
  {
    \schemabox{\textsc{Dei}, qua-\\lis sit, aut}
  }
  {
    \schema
    {\schemabox{\textsc{Per se}:\\ scilicet.}}
    {
      \schemabox{Unus in essentia.}\smallskip
      \schemabox{Trinus in personis.}
    }\smallskip
    \schema
    {\schemabox{Ad hominem\\ quem vel}}
    {
      \schemabox{Accusat \& terret, \textsc{Per Legem},\\
      Consolatur \& erigit, \textsc{Per Evangelium}.\\
      Salvat, \textsc{Per Christum}.\\
      Renovat, \textsc{Per Spiritum Sanctum}.\\
      Sanctificat, \textsc{Per Verbum} \& \textsc{Sacramenta}.\\
      Castigat, tentat \& exercet, \textsc{Per Crucem}.\\
      Glorificat \textsc{Per Resurrectionem Carnis}\\
      \textsc{\quad Ad Vitam \AE{}ternam}.}
    }\medskip
  }
  \Schema{-13.4ex}{17.4ex}
  {
    \schemabox{\textsc{Hominis},\\ qualis sit}
  }
  {
    \Schema{-4.4ex}{5ex}
    {\schemabox{\textsc{Per se}:}}
    {
      \schemabox{Ante lapsum.}\smallskip
      \schema
      {\schemabox{Post lapsum:}}
      {
        \schemabox{Ante Regenerationem \&\\
        Renovationem S. Sancti.}\medskip
        \schemabox{Post Regenerationem \&\\
        Renovationem S. Sancti.}\smallskip
      }\smallskip
    }
```

```
    \Schema{4.2ex}{14.4ex}
    {\schemabox{Ad}}
    {
      \schema
      {\schemabox{\textsc{Deum},}}
      {
        \schemabox{P\oe{}nitentia agens, agnitis peccatis \&\\
        ira Dei cognita \textsc{Ex Lege}.\\
        Erigens se \textsc{Voce Evangelii}.\\
        Credens \textsc{In Christum Salvatorem}.\\
        Non repugnans \textsc{Spiritui Sancto} impellenti.\\
        Audiens \textsc{Verbum}: \& utens \textsc{Sacramentis}.\\
        Patienter \& constanter sufferens \textsc{Crucem}.\\
        Sperans \& expectans glorificationem\\
        \textsc{\quad In Resurrectione Carnis}\\
        \textsc{\quad Ad Vitam \AE{}ternam}.}
      }\smallskip
      \schema
        {\schemabox{seipsum ratione}}
        {\schemabox{Anim\ae{}\\ vel\\ Corporis}}\smallskip
      \Schema{2ex}{5.1ex}
      {\schemabox{Proximum,}}
      {
        \schema
        {\schemabox{Amicum ra-\\ tione vel}}
        {
          \schemabox{Religionis.\\
          Politic\ae{} \& \OE{}conomic\ae{}.\\
          Cognationis.\\
          Agnationis.}
        }\smallskip
        \schemabox{Inimicum.}
      }
    }
  }
}
```

2.4.4 Another example

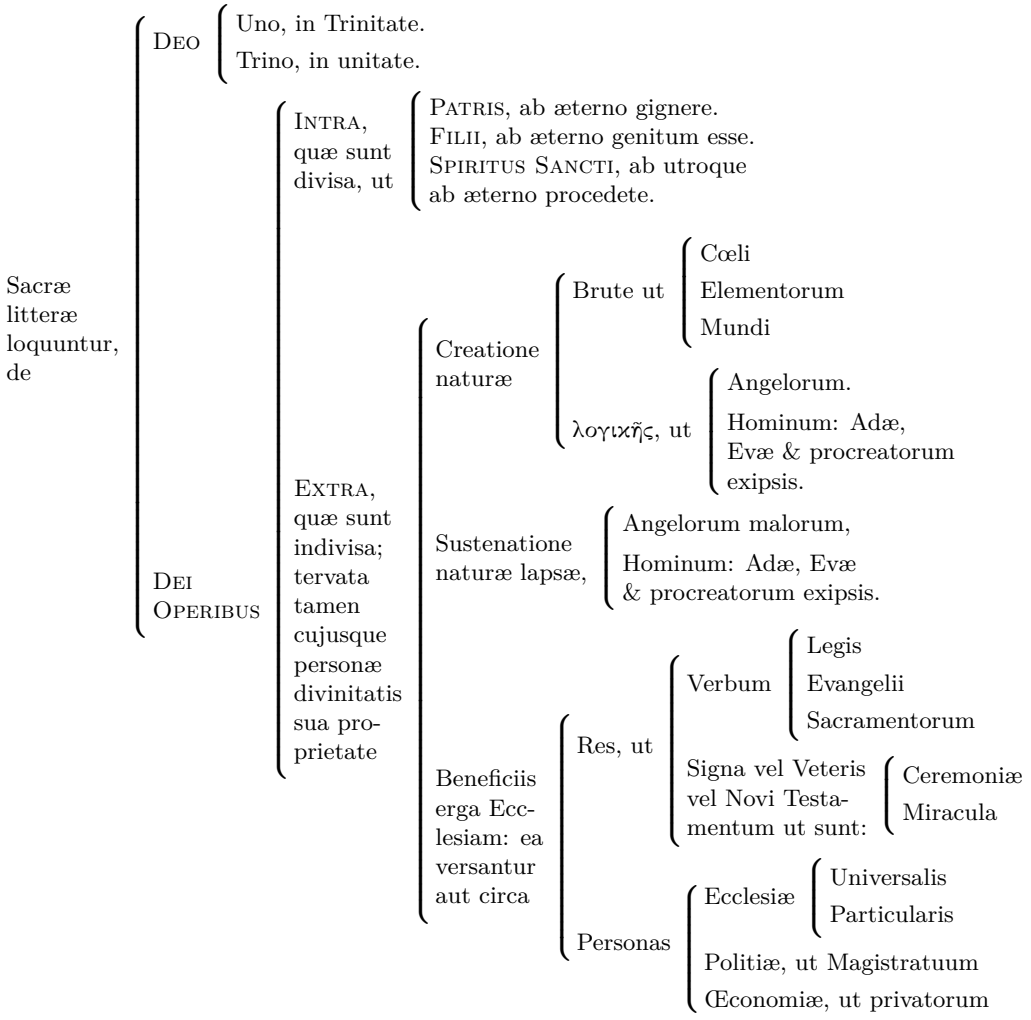
We use `\DoGroups` in a local scope for this next schema. The listing illustrates a more extensive use of spacing. The macro `\gk` uses babel to create Greek text. The version of `\gk` in `schematest.tex` can use either babel or polyglossia to create Greek text, or it can just show a transliteration. In this listing we do not break each line following the macro `\\` in a `\schemabox`.

```
\DoGroups
\Schema{-26ex}{21.5ex}
{
  \schemabox{Sacr\ae{}\\ litter\ae{}\\ loquuntur,\\ de}
}
```

```
{
  \schema
  {
    \schemabox{\textsc{Deo}}
  }
  {
    \schemabox{Uno, in Trinitate.}\smallskip
    \schemabox{Trino, in unitate.}
  }
  \Schema{-16.5ex}{23.4ex}
  {
    \vskip30ex
    \schemabox{\textsc{Dei}\\ \textsc{Operibus}}
  }
  {
    \schema
    {
      \schemabox{\textsc{Intra},\\ qu\ae{} sunt\\ divisa, ut}
    }
    {
      \schemabox{\textsc{Patris}, ab \ae{}terno gignere.\\
      \textsc{Filii}, ab \ae{}terno genitum esse.\\
      \textsc{Spiritus Sancti}, ab utroque\\
      ab \ae{}terno procedete.}
    }\medskip
    \Schema{-2ex}{21ex}
    {
      \schemabox{\textsc{Extra},\\ qu\ae{} sunt\\ indivisa;\\
      tervata\\ tamen\\ cujusque\\ person\ae{}\\
      divinitatis\\ sua pro-\\ prietate}
    }
    {
      \Schema{-0.8ex}{6.4ex}
      {
        \schemabox{Creatione\\ natur\ae{}}
      }
      {
        \schema
        {
          \schemabox{Brute ut}
        }
        {
          \schemabox{C\oe{}li} \smallskip
          \schemabox{Elementorum} \smallskip
          \schemabox{Mundi}
        }\smallskip
        \schema
        {
          \schemabox{\gk{logik~hs}, ut}
        }
        {
          \schemabox{Angelorum.} \smallskip
          \schemabox{Hominum: Ad\ae{},\\
          Ev\ae{} \& procreatorum\\ exipsis.}
        }
      }
```

```
\schema
{
  \schemabox{Sustenatione\\ natur\ae{} laps\ae{},}
}
{
  \schemabox{Angelorum malorum,} \smallskip
  \schemabox{Hominum: Ad\ae{}, Ev\ae{}\\
  \& procreatorum exipsis.}
}
\Schema{2.6ex}{8.8ex}
{
  \schemabox{Beneficiis\\ erga Ecc-\\ lesiam: ea\\
  versantur\\ aut circa}
}
{
  \Schema{2.2ex}{7ex}
  {
    \schemabox{Res, ut}
  }
  {
    \schema
    {
      \schemabox{Verbum}
    }
    {
      \schemabox{Legis} \smallskip
      \schemabox{Evangelii} \smallskip
      \schemabox{Sacramentorum}
    }\smallskip
    \schema
    {
      \schemabox{Signa vel Veteris\\
      vel Novi Testa-\\ mentum ut sunt:}
    }
    {
      \schemabox{Ceremoni\ae{}} \smallskip
      \schemabox{Miracula}
    }
  }
  \smallskip
  \Schema{1ex}{5ex}
  {
    \schemabox{Personas}
  }
  {
    \schema
    {
      \schemabox{Ecclesi\ae{}}
    }
    {
      \schemabox{Universalis} \smallskip
      \schemabox{Particularis}
    }
```

```
115             \smallskip
116             \schemabox{Politi\ae, ut Magistratuum}
117             \smallskip
118             \schemabox{\OE{}conomi\ae, ut privatorum}
119           }
120         }\vskip 2ex
121       }
122     }
123   }
```

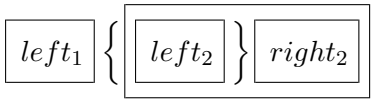


2.4.5 Open and Closed Schemata

Now we look at schemata that have both open and closed braces. One must use `\Schema` to get delimiters to be the same height. These schemata take the form:

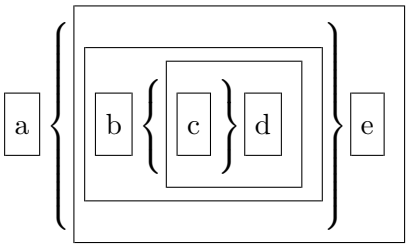
```
\Schema{⟨adjust⟩}{⟨height⟩}
{⟨left1⟩}
{
  \Schema[close]{⟨adjust⟩}{⟨height⟩}
  {⟨left2⟩}
  {⟨right2⟩}
}
```

Perhaps we can better illustrate this nesting by using a modified version of the `\Box` macro from above:



Here is another, more complex example:

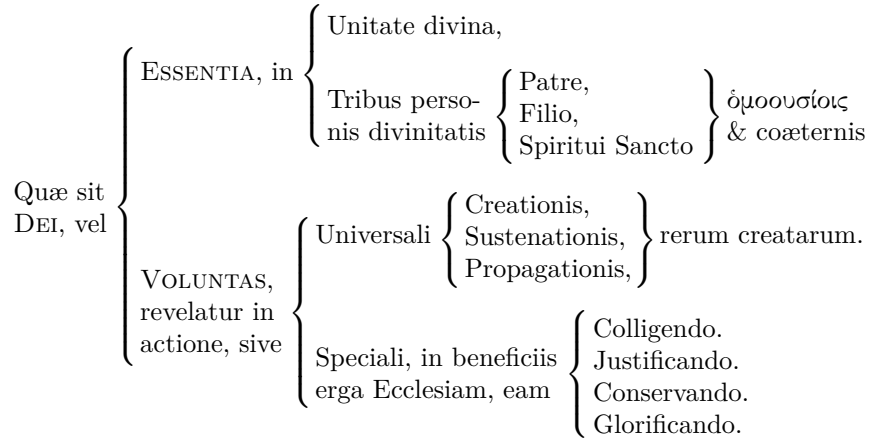
```
\Schema{0ex}{6ex}
{\Box{a}}
{%
  \Box{%
    \Schema[close]{0ex}{6ex}
    {%
      \Box{%
        \Schema{0ex}{3ex}
        {\Box{b}}
        {%
          \Box{%
            \Schema[close]{0ex}{3ex}
            {\hbox{\Box{c}\kern0.2em}}
            {\Box{d}}
          }
        }
      }
    }
    {\Box{e}}
  }
}
```



Now we move on to another example from the *Loci*. As above, one must use `\Schema` to prevent the opening braces from being slightly larger than the closing braces. First we show the schema, then we show the listing.

The macro \gk uses babel. The version included in schematest.tex can use babel, polyglossia or just transliterated text.

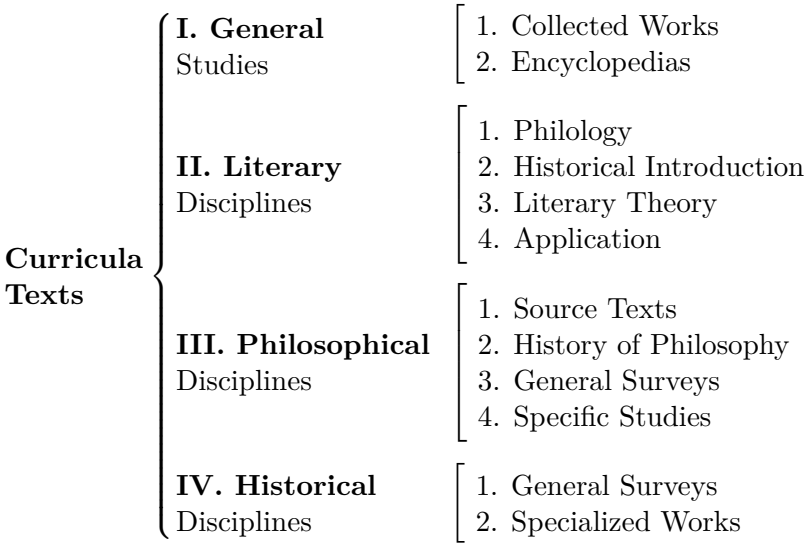
```
\Schema{-1.4ex}{10ex}
{\schemabox{Qu\ae{} sit\\ \textsc{Dei}, vel}}
{
  \Schema{-1ex}{5ex}
  {\schemabox{\textsc{Essentia}, in}}
  {
    \vskip1ex\schemabox{Unitate divina,}
    \medskip
    \Schema{0ex}{3.4ex}
    {\schemabox{Tribus perso-\\ nis divinitatis}}
    {
      \Schema[close]{0ex}{3.4ex}
      {\NudgeSB\schemabox{Patre,\\ Filio,\\ Spiritui Sancto}}
      {\schemabox{\gk{<omoous'iois}\\ \& co\ae{}ternis}}
    }
  }
  \medskip
  \Schema{-0.2ex}{6.4ex}
  {\schemabox{\textsc{Voluntas},\\ revelatur in\\ actione, sive}}
  {
    \Schema{0ex}{3.4ex}
    {\schemabox{Universali}}
    {
      \Schema[close]{0ex}{3.4ex}
      {\schemabox{Creationis,\\ Sustenationis,\\ Propagationis,}}
      {\schemabox{rerum creatarum.}}
    }
    \medskip
    \schema
    {\schemabox{Speciali, in beneficiis\\ erga Ecclesiam, eam}}
    {\schemabox{Colligendo.\\ Justificando.\\
     Conservando.\\ Glorificando.}}
  }
}
```



2.5 Fixed-Width Boxes

This final example illustrates using `\schemabox` with a fixed width. We have defined `\mybox` elsewhere with `\newbox`. Bear in mind that `\newbox` is an outer macro in Plain TeX.

```
\Schema{0ex}{15ex}
{\schemabox{\bfseries Curricula\\\bfseries Texts}}
{
  \DoBrackets%
  \setbox\mybox=\hbox{\bfseries III. Philosophical }%
  \dimen0=\wd\mybox%
  \schema
    {\schemabox[\dimen0]
      {\bfseries I. General\\Studies}}
    {\NudgeSB[before]%
     \schemabox
      {1. Collected Works\\
       2. Encyclopedias}}
  \medskip
  \schema
    {\schemabox[\dimen0]
      {\bfseries II. Literary\\Disciplines}}
    {\NudgeSB[before]%
     \schemabox{1. Philology\\
       2. Historical Introduction\\
       3. Literary Theory\\
       4. Application}}
  \medskip
  \schema
      {\schemabox[\dimen0]
        {\bfseries III. Philosophical\\Disciplines}}
      {\NudgeSB[before]%
       \schemabox{1. Source Texts\\
         2. History of Philosophy\\
         3. General Surveys\\
         4. Specific Studies}}
  \medskip
  \schema
      {\schemabox[\dimen0]
        {\bfseries IV. Historical\\Disciplines}}
      {\NudgeSB[before]%
        \schemabox{1. General Surveys\\
        2. Specialized Works}}
}
```



3 Implementation

Shorter macros are written entirely in both LaTeX and TeX. Longer macros implement both a LaTeX front end and a generic front end with a common TeX back end. If the format is `LaTeX2e` then the macros use the LaTeX 2_ε front end. Otherwise they use generic TeX, meaning Plain TeX, eplain, and Lollipop — maybe others too, but they are not supported.

In order to support the largest diversity of formats and TeX engines, we avoid newer primitives like `\unless` and `\ifdefined`. Thus, we must revert to the "old" way of testing whether or not a macro is defined.

`\schemataLaTeX` Below we manually duplicate with verbatim material what we put early in the `dtx` file for the versioning information to work. The `\schemataLaTeX` macro normally is undefined until it is assigned the value of `LaTeX2e`, to be compared with `\fmtname`. If we are not using LaTeX 2_ε, we do the equivalent of `\makeatletter` in either Plain TeX or eplain.

```
%<package>\expandafter\ifx \csname schemataLaTeX\endcsname\relax
%<package>  \def\schemataLaTeX{LaTeX2e}\fi
%<package>\ifx\fmtname\schemataLaTeX
%<package>\expandafter\NeedsTeXFormat\expandafter{\schemataLaTeX}[2005/12/01]
%<package>\ProvidesPackage{schemata}
%<*package>
  [2025/08/27 1.5 generic package to aid construction of topical categories]
%</package>
%<package>\else
%<package>\catcode`@=11\relax
%<package>\fi
```

3.1 Internal Variables

`\@schemata@LaTeX` We declare the macro `\@schemata@LaTeX` to be the value of `\schemataLaTeX` to safeguard package operation. From this point onward we can display or query `\schemataLaTeX` for user-side tests without affecting package internals.

```
\edef\@schemata@LaTeX{\schemataLaTeX}
```

Two box registers and two dimen registers are used to analyze the left-hand and right-hand vertical sizes of the boxes in a schema. Three more dimen registers are for scratchwork.

```
\newbox\@schemata@rhs%
\newbox\@schemata@lhs%
\newdimen\@schemata@rheight%
\newdimen\@schemata@lheight%
\newdimen\@schemata@one%
\newdimen\@schemata@two%
\newdimen\@schemata@three%
```

Two Boolean flags affect the height of a `\schemabox`, respectively setting and toggling that height for lowercase and uppercase content in order to add or remove space for boxes with only lowercase text.

```
\newif\if@schemata@LCBox%
\newif\if@schemata@SWBox%
```

These Boolean flags determine if a kern should be added to the beginning or the end of each line in a `\schemabox` (helps with closed braces and certain delimiters).

```
\newif\if@schemata@LNudge%
\newif\if@schemata@RNudge%
```

3.2 Package Options

We set braces to be the default set of delimiters. Apart from LaTeX 2_ε we ignore the options. Four options are implemented, namely, `braces` (the default), `brackets`, `parens`, and `groups`. Since the options are used infrequently, we naively process them in whatever order we get, each overwriting the last.

```
\ifx\fmtname\@schemata@LaTeX
  \DeclareOption{braces}%
    {\let\@schemata@LD\lbrace%
     \let\@schemata@RD\rbrace}
  \DeclareOption{brackets}%
    {\let\@schemata@LD\lbrack%
     \let\@schemata@RD\rbrack}
  \DeclareOption{parens}%
    {\let\@schemata@LD(%
     \let\@schemata@RD)}
  \DeclareOption{groups}%
    {\let\@schemata@LD\lgroup%
     \let\@schemata@RD\rgroup}
  \ExecuteOptions{braces}%
  \ProcessOptions\relax
\else
  \let\@schemata@LD\lbrace%
  \let\@schemata@RD\rbrace%
\fi
```

3.3 Macros

`\DoBraces` Set the delimiters to be braces. This is local to a scope, including within a schema.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\DoBraces}%
    {\let\@schemata@LD\lbrace%
     \let\@schemata@RD\rbrace}
\else
  \def\DoBraces%
    {\let\@schemata@LD\lbrace%
     \let\@schemata@RD\rbrace}
\fi
```

`\DoBrackets` Set the delimiters to be brackets. This is local, as above.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\DoBrackets}%
    {\let\@schemata@LD\lbrack%
     \let\@schemata@RD\rbrack}
\else
  \def\DoBrackets%
    {\let\@schemata@LD\lbrack%
     \let\@schemata@RD\rbrack}
\fi
```

`\DoParens` Set the delimiters to be parentheses. This is local, as above.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\DoParens}%
    {\let\@schemata@LD(%
     \let\@schemata@RD)}
\else
  \def\DoParens%
    {\let\@schemata@LD(%
     \let\@schemata@RD)}
\fi
```

`\DoGroups` Set the delimiters to be parentheses. This is local, as above.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\DoGroups}%
    {\let\@schemata@LD\lgroup%
     \let\@schemata@RD\rgroup}
\else
  \def\DoGroups%
    {\let\@schemata@LD\lgroup%
     \let\@schemata@RD\rgroup}
\fi
```

`\LCschema` Prevent `\schemabox` from adding a `\strut` in the first line.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\LCschema}{\@schemata@LCBoxtrue}
\else
  \def\LCschema{\@schemata@LCBoxtrue}
\fi
```

\UCschema Permit \schemabox to add a \strut in the first line (default).

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\UCschema}{\@schemata@LCBoxfalse}
\else
  \def\UCschema{\@schemata@LCBoxfalse}
\fi
```

\SwitchSB Flip the UC/LC settings for one \schemabox, which will reset this value on exit.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\SwitchSB}{\@schemata@SWBoxtrue}%
\else
  \def\SwitchSB{\@schemata@SWBoxtrue}
\fi
```

\NudgeSB Add a kern to the beginning or the end (or both, if invoked twice) of each line in one \schemabox. This will be reset on exit from that \schemabox. First comes the front end:

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\NudgeSB}[1][right]{\@schemata@NudgeSB[#1]}
\else
  \def\NudgeSB{\futurelet\@schemata@testchar\@schemata@N@dgeSB}
  \def\@schemata@N@dgeSB{%
    \ifx[\@schemata@testchar
      \let\next\@schemata@NudgeSB%
    \else
      \let\next\@schemata@@NudgeSB%
    \fi
    \next%
  }%
  \def\@schemata@@NudgeSB{\@schemata@NudgeSB[right]}
\fi
```

Next comes the common back end:

```
\def\@schemata@NudgeSB[#1]{%
  \edef\@schemata@NudgeTest{#1}%
  \def\@schemata@NudgeDefault{right}%
  \ifx\@schemata@NudgeTest\@schemata@NudgeDefault
    \@schemata@RNudgetrue%
  \else
    \@schemata@LNudgetrue%
  \fi
}
```

\SBNudgeFactor Define the \kern to be added to the end of each line in one \schemabox. The default is 0.2em, equal to the horizontal corrective.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand{\SBNudgeFactor}{\kern0.2em}
\else
  \def\SBNudgeFactor{\kern0.2em}
\fi
```

\schemabox If in internal vertical mode, restricted horizontal mode, or math mode, wrap a stack of \hboxes in a \vbox, then put that inside an \hbox. The first argument sets an optional width for those \hboxes. Normally insert a \strut in the first \hbox. The second argument contains the rows of horizontal material, where \\ is redefined to end one \hbox and begin another. Add kerns as directed by \NudgeSB. When in any other mode, just display the second argument as text. First is the front end:

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\schemabox}[2][0pt]{\@schemata@schemabox[#1]{#2}}
\else
  \def\schemabox{\futurelet\@schemata@testchar\@schemata@schemab@x}
  \def\@schemata@schemab@x{%
    \ifx[\@schemata@testchar
      \let\next\@schemata@schemabox%
    \else
      \let\next\@schemata@@schemab@x%
    \fi
    \next%
  }%
  \def\@schemata@@schemab@x#1{\@schemata@schemabox[0pt]{#1}}
\fi
```

Next comes the common back end:

```
\def\@schemata@schemabox[#1]#2{%
  \ifinner
    \if@schemata@LCBox
      \def\@schemata@Adj{}%
      \if@schemata@SWBox\def\@schemata@Adj{\strut}\fi
    \else
      \def\@schemata@Adj{\strut}%
      \if@schemata@SWBox\def\@schemata@Adj{}\fi
    \fi
    \if@schemata@RNudge
      \let\@schemata@NudgeR\SBNudgeFactor%
    \else
      \def\@schemata@NudgeR{}%
    \fi
    \if@schemata@LNudge
      \let\@schemata@NudgeL\SBNudgeFactor%
    \else
      \def\@schemata@NudgeL{}%
    \fi
    \ifdim#1<1pt
      \def\\{\@schemata@NudgeR\egroup%
        \hbox\bgroup\@schemata@NudgeL\ignorespaces }%
      \vbox{\hbox\bgroup%
        \@schemata@Adj\@schemata@NudgeL\ignorespaces #2%
        \@schemata@NudgeR\egroup}%
    \else
      \def\\{\hfil\egroup%
        \hbox to #1\bgroup\@schemata@NudgeL\ignorespaces }%
      \vbox{\hbox to #1\bgroup%
        \@schemata@Adj\@schemata@NudgeL\ignorespaces #2%
        \hfil\egroup}%
    \fi
  \else
```

```
    #2%
  \fi
  \@schemata@SWBoxfalse%
  \@schemata@RNudgefalse%
  \@schemata@LNudgefalse%
}
```

\schema This "simple" schema vertically centers two boxes of internal vertical material and puts a "simple" brace between the boxes based on the height of the box and the options passed to the schema.

There is something of a "magic" value for adjusting the height used for the larger side of a `\schema`, namely `1.44265ex`. By using this adjustment, which is slightly larger than $\sqrt{2}$ times the ex-height of the font, the results look more aesthetically pleasing in terms of centering and size of the braces.

By default, a schema has a box to the left, an open delimiter, and a box to the right. If any optional argument other than `open` is used, the schema prints a box to the left, a close brace, and a box to the right. First is the front end:

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand{\schema}[3][open]{%
    \@schemata@schema[#1]{#2}{#3}}
\else
  \long\def\schema{\futurelet\@schemata@testchar\@schemata@schem@}
  \long\def\@schemata@schem@{%
    \ifx[\@schemata@testchar
      \let\next\@schemata@schema%
    \else
      \let\next\@schemata@@schem@%
    \fi
    \next%
  }%
  \long\def\@schemata@@schem@#1#2{%
    \@schemata@schema[open]{#1}{#2}}
\fi
```

Next comes the common back end:

```
\long\def\@schemata@schema[#1]#2#3{%
  \edef\@schemata@option{#1}%
  \def\@schemata@open{open}%
  \ifx\@schemata@option\@schemata@open
    \setbox\@schemata@rhs=\vbox{#3}%
    \@schemata@rheight=\ht\@schemata@rhs%
    \advance\@schemata@rheight\dp\@schemata@rhs%
    \advance\@schemata@rheight by 1.44265ex%
    \hbox{$\vcenter{#2}%
      \@schemata@lbrace{\@schemata@rheight}%
      \vcenter{#3}$}%
  \else
    \setbox\@schemata@lhs=\vbox{#2}%
    \@schemata@lheight=\ht\@schemata@lhs%
    \advance\@schemata@lheight\dp\@schemata@lhs%
    \advance\@schemata@lheight by 1.44265ex%
    \hbox{$\vcenter{#2}%
      \kern-0.2em\@schemata@rbrace{\@schemata@lheight}%
      \vcenter{#3}$}%
  \fi
```

```
}
```

`\Schema` This is the general-purpose form of schemata. The arguments include whether it is an open or closed schema, the vertical adjustment of the left-hand side and delimiter over against the right-hand side, the size of the brace, and the contents of the left and right-hand sizes. It works about the same as above, but requires manual adjustment of the braces. Again we see the "magic" height adjustment value of `1.44265ex`. First is the front end:

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand{\Schema}[5][open]{%
    \@schemata@Schema[#1]{#2}{#3}{#4}{#5}}
\else
  \long\def\Schema{\futurelet\@schemata@testchar\@schemata@Schem@}
  \long\def\@schemata@Schem@{%
    \ifx[\@schemata@testchar
      \let\next\@schemata@Schema%
    \else
      \let\next\@schemata@@Schem@%
    \fi
    \next%
  }%
  \long\def\@schemata@@Schem@#1#2#3#4{%
    \@schemata@Schema[open]{#1}{#2}{#3}{#4}}
\fi
```

Next comes the common back end:

```
\long\def\@schemata@Schema[#1]#2#3#4#5{%
  \edef\@schemata@option{#1}%
  \def\@schemata@open{open}%
  \@schemata@one=#2%
  \ifx\@schemata@option\@schemata@open
    \hbox{$\vcenter{\vskip1.44265\@schemata@one#4}%
      \@schemata@biglbrace{#2}{#3}\vcenter{#5}$}%
  \else
    \hbox{$\vcenter{\vskip1.44265\@schemata@one#4}\kern-0.2em%
      \@schemata@bigrbrace{#2}{#3}\vcenter{#5}$}%
  \fi
}
```

`\@schemata@lbrace` Draw an on-center delimiter to the left of a simple box.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\@schemata@lbrace}[1]{%
    \ifmmode
      \left.\vcenter{\vbox to #1{\vfil}}\right\@schemata@LD%
    \fi
  }
\else
  \def\@schemata@lbrace#1{%
    \ifmmode
      \left.\vcenter{\vbox to #1{\vfil}}\right\@schemata@LD%
    \fi
  }
\fi
```

`\@schemata@rbrace` Draw an on-center delimiter to the right of a simple box.

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\@schemata@rbrace}[1]{%
    \ifmmode
      \left\@schemata@RD\vcenter{\vbox to #1{\vfil}}\right.%
    \fi
  }
\else
  \def\@schemata@rbrace#1{%
    \ifmmode
      \left\@schemata@RD\vcenter{\vbox to #1{\vfil}}\right.%
    \fi
  }
\fi
```

`\@schemata@biglbrace` Draw a vertically-adjustable delimiter to the left of a complex assortment of boxes. Again we see the "magic" height adjustment value of `1.44265ex`, but both positive and negative. First is the front end:

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\@schemata@biglbrace}[2]{%
    \@schemata@@biglbrace{#1}{#2}}
\else
  \def\@schemata@biglbrace#1#2{%
    \@schemata@@biglbrace{#1}{#2}}
\fi
```

Next comes the common back end:

```
\def\@schemata@@biglbrace#1#2{%
  \@schemata@one=#1%
  \@schemata@two=#2%
  \@schemata@three=-\@schemata@two%
  \ifdim\@schemata@three>\@schemata@two%
    \@schemata@two=\@schemata@three\fi
  \ifdim\@schemata@one<0pt
    \ifmmode\vcenter{\hbox{$\left.%
      \vbox to 1.44265\@schemata@two{\vfil}%
      \right\@schemata@LD%
      \atop\vbox to -1.44265\@schemata@one{\vfil}$}}\fi
  \else
    \ifmmode\vcenter{\hbox{$\vbox to 1.44265\@schemata@one{\vfil}%
      \atop\left.%
      \vbox to 1.44265\@schemata@two{\vfil}%
      \right\@schemata@LD$}}\fi
  \fi
}
```

`\@schemata@bigrbrace` Draw a vertically-adjustable delimiter to the right of a complex assortment of boxes. Again we see the "magic" height adjustment value of `1.44265ex`, but both positive and negative. First is the front end:

```
\ifx\fmtname\@schemata@LaTeX
  \newcommand*{\@schemata@bigrbrace}[2]{%
    \@schemata@@bigrbrace{#1}{#2}%
  }
\else
```

```
  \def\@schemata@bigrbrace#1#2{%
    \@schemata@@bigrbrace{#1}{#2}%
  }
\fi
```

Next comes the common back end:

```
\def\@schemata@@bigrbrace#1#2{%
  \@schemata@one=#1%
  \@schemata@two=#2%
  \@schemata@three=-\@schemata@two%
  \ifdim\@schemata@three>\@schemata@two%
    \@schemata@two=\@schemata@three\fi
  \ifdim\@schemata@one<0pt
    \ifmmode\vcenter{\hbox{$\left.%
      \vbox to 1.44265\@schemata@two{\vfil}%
      \right\@schemata@RD%
      \atop\vbox to -1.44265\@schemata@one{\vfil}$}}\fi
  \else
    \ifmmode\vcenter{\hbox{$\vbox to 1.44265\@schemata@one{\vfil}%
      \atop\left.%
      \vbox to 1.44265\@schemata@two{\vfil}%
      \right\@schemata@RD$}}\fi
  \fi
}
```

If we are not using LaTeX 2_ε, we do the equivalent of `\makeatother`.

```
\ifx\fmtname\@schemata@LaTeX\else
  \catcode`@=12\relax
\fi
```

4 Change History

5 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; numbers in roman refer to the code lines where the entry is used.