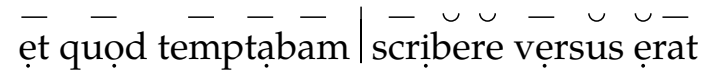
The mětrix package

Tobias Weh*

Version 1.5 – Released 2019/10/09

Abstract



The mětrix package can be used to print the prosodics/metrics of (latin) verses. It provides macros to typeset the symbols stand alone and in combination with syllables (including automatic alignment like seen above). Furthermore it defines a new brĕvis and a lōnga accent[1] and a ⏡ bow to contract syllables.

Thanks to David Carlisle, Marco Daniel, Enrico Gregorio, Bruno Le Floch and Joseph Wright who helped me with starting in LaTeX3 programming. The verse above is by Ovid in his Tristia 4,10,26.

1 Prerequisites

mětrix relies only on a few packages: tikz (including the `calc` library), xpatch and xparse, which stand for the whole LaTeX3 bundle.

2 Package loading

Load mětrix as usual with `\usepackage{metrix}`. At the moment it has no options.

A CWL file `metrix.cwl` for autocompletition in TeXstudio is available in the GitHub repo. To install the CWL file copy it to `~/.config/texstudio/` on Linux and OS X and to `C:\Documents and Settings/User/AppData/Roaming/texstudio/`. See section 1.5 of the TeXstudio manual for more information.

*URL: http://tobiw.de/en, Mail: mail@tobiw.de

[1] I know that these signs are no accents in the liguistic sense, but they are in the TeX tradition …

3 Bugs and feedback

3.1 Known issues

- At the moment the escaping of hyphen chars is not that good (see section 7.3).
- Unfortunatly you can't use the active quotes of csquotes inside of `\metrics` syllable list (see section 7.4).

I'm sure there are more bugs and issues let me know if you find them …

3.2 Feedback

Any feedback on **mětrix** is appreciated. You may use its GitHub repository at `https://github.com/tweh/metrix` to request features and report bugs or send me an e-mail (mail@tobiw.de).

Please note that I don't speak latin myself and forthat the examples in this manual may be wrong—as long as they show how to use the package I don't consider such errors as bugs `;-)`.

4 Metric symbols

4.1 Syntax for symbols

Before I'll show you the central macros for typesetting the symbols, you need to "learn" the syntax for the symbols. All symbols are represented by a single or a combination of characters. The list with all available abbreviations can be found in table 1. Please keep in mind that **mětrix** uses spaces to separate the abbreviations an something like `_'x` will cause an error, the correct input is `_␣'␣x` (where ␣ indicates a space/blank).

Accents above symbols (ictus)

If you want to add an additional accent above a symbol you may precede the symbol with an slash `/` for an acute or a star `*` for a grave accent. At the moment this only works with breve (`/u` or `*u`) and longum (`/_` or `*_`).

Example

An ictus (although out of date).

```
\metricsymbols{/_ u u /_ _ /_}
```

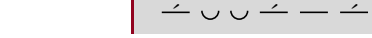


Table 1: Symbol abbreviations

abbreviation		symbol	name/explantion	accent/ictus?
`e`			empty (= invisible) symbol	
`u`		⏑	elementum breve	yes
`_`	*underscore*	—	elementum longum	yes
`uu`		⏑⏑	double breve	
`uu_`		⏑̲⏑̲	elementum biceps	
`_uu`		⏒⏑	elementum biceps	
`u_uu`		⏑⏒	elementum anceps	
`x`		×	elementum anceps	
`n`		◠	elementum indifferens	
`u_`		⏑̲	elementum indifferens	
`_u`		⏒	elementum indifferens	
`A`		^	*	
`v`		ˇ	*	
`o`		○	*	
`oo`		○○	aeolic base	
`l`		|	caesura or break	
`ll`		‖	caesura or end of period	
`p`		⁄	primary stress	
`s`		∖	secondary stress	
`|`	*pipe*	|	break (see 4.4)	
`||`	*two pipes*	‖	verse break (see 4.4)	
`'`	*apostrophe* (*shift* + #)	|	shorter break (see 4.4)	
`,`	*comma*	|	shorter break (see 4.4)	

** I added these symbols as someone might needs them. If you use them please let me know what to add as name/explanation.*

4.2 Stand alone metric symbols

`\metricsymbols ⋆`

`\metricsymbols`⟨`*`⟩`[`⟨*highlighting*⟩`]{`⟨*symbols*⟩`}`

This macro typesets stand alone versions of the symbols, i.e. without syllables below (or above) of them. Use the starred version for smaller (in line) symbols and the normal version for bigger symbols. ⟨*symbols*⟩ must be a list of abbreviations as explained in section 4.1; the abbreviations must be separated by one (or more) spaces.

Example

The *diphilius* can be shown with this code.

```
\metricsymbols{_ _uu _ _uu u_ | x _ u u _ x u_}
```

— ⏑⏑ — ⏑⏑ ⏒ | × — ⏑ ⏑ — × ⏒

4.3 Metric symbols above (or below) syllables

\metrics ⋆ `\metrics[`⟨*highlighting*⟩`]{`⟨*symbols*⟩`}{`⟨*syllables*⟩`}`

This command can be used to align the symbols above (or below) syllables. The first argument works as in `\metricsymbols`, the second argument ⟨*syllables*⟩ takes the hyphenated verse.

Example

```
\metrics{_    u  u  _   _  _  | _    _  u  u   _  _   _   }
        {flos ve-te-ris vi-ni | meis na-ri-bus ob-iec-tust}
```

— ⏑ ⏑ — — — | — — ⏑ ⏑ — — —
flos veteris vini | meis naribus obiectust

You may use multiple spaces to align the abbreviations above the syllables but this is not mandatory and does not affect the output. But mind that the number of syllables equals the number of symbols. If you use the ○○ symbol you may omit the hyphen between the two syllables beloning to this symbol. You can merge multiple words by *embracing* them.

Example

```
\metrics{_   u  u    _       _   _  _   _  }
        {mol-ta quo-{que et} bel-lo pas-sus}
```

— ⏑ ⏑ — — — — —
molta quoque et bello passus

The macros `\metrics` and `\metricsymbols` can also be used to typeset single symbols or symbol syllable combinations.

Example

```
The \metricsymbols*{_uu} shows an \emph{elementum biceps}.
```

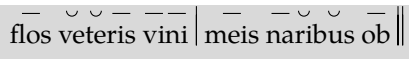
The ⏑⏑ shows an *elementum biceps*.

4.4 Adding symbols for breaks

As seen in the examples above you can use pipes, i.e. `|` or `||`, to mark breaks. In `\metrics` the markers must appear in ⟨*symbols*⟩ and ⟨*syllables*⟩.

Example

```
\metrics{_    u  u  _   _  _  | _    _  u  u   _  ||}
        {flos ve-te-ris vi-ni | meis na-ri-bus ob ||}
```

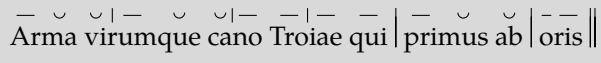


If you want the breaks to be shown in the symbol line only you can use the shorter break which is represented by an apostrophe (shift + #) or a comma. This mark must be used in ⟨*symbols*⟩ only and is kind of special:

- It *can't* be highlighted and thus doesn't count for the numbers used for highlights,
- it is ignored at the beginning and the end of ⟨*symbols*⟩,
- in `\metricsymbols` it is treated like the pipe, and
- TEX needs at least one additional run to get the right positions.

Example

```
\metrics{_  u  u ' _  u  u ' _  _ ' _  _  |  _  u  u  |  _ _  ||}
        {Ar-ma vi-rum-que ca-no Tro-iae qui | pri-mus ab |  o-ris ||}
```



The difference between `'` and `,` is that the break defined with an apostrophe is vertically centred between the surrounding symbols while the break set with the comma is vertically centred between the corresponding syllables. They both align horizontally within the row of symbols.

Example

```
\emph{apostrophe:}
\metrics{_  _  '  _          }
        {au-ra-{r\bow{um e}st}} \qquad
\emph{comma:}
\metrics{_  _  ,  _          }
        {au-ra-{r\bow{um e}st}}
```

4.5 Highlight certain symbols/syllabels

As you can see above `\metrics` and `\metricsymbols` got an optional argument taking some options to highlight a certain symbol/syllable. The ⟨*highlighting*⟩ list must contain one or more comma separated pairs of ⟨*numbers*⟩=⟨*style*⟩, where ⟨*numbers*⟩ is the number of a symbol/syllable (e.g. `3`) or a list of numbers separated by plus signs (e.g. `2+3+5`) in the list and ⟨*style*⟩ is any Ti*k*Z style (other Ti*k*Z options may not work properly, so you maybe must create your own style, see section 7.9.)

mětrix comes with several predefined highlighting styles:

- `add arrow`

  This style adds an arrow above the metric symbol. To change the arrow symbol, edit the **mětrix** variable `arrow`.

- `add text=`⟨*text*⟩

This style takes a *mandatory* argument to add some text above a symbol. To change the default font change the font of the Ti*k*Z node style `every metrix added text`.

- `bold highlight`

- `colored highlight=`⟨***color***⟩

This style has an *optional* argument to change the highlighting color on the fly. To change the color in general change the value of the variable `highlightcolor`.

- `dashed highlight`

- `filled highlight=`⟨***color***⟩

This style has an *optional* argument to change the filling color on the fly. To change the color in general change the value of the variable `fillcolor`.

- `superscript=`⟨***text***⟩

This style takes a *mandatory* argument to add a superscript letter or a number to a symbol. It is designed to work with the break symbols, but works with others too.

Sytles with an agrument must be set in braces (see the examples)**!**

Example

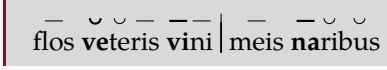
Higlight some syllables with color.

```
\metrics
   [
      2=colored highlight,
      4={colored highlight=orange},
      5={colored highlight=blue},
      7=colored highlight,
      11=colored highlight
  ]
  {_    u  u  _   _  _  | _    _  u  u   _ }
  {flos ve-te-ris vi-ni | meis na-ri-bus ob}
```

Example

The shorter version using the + syntax.

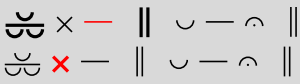
```
\metrics[2+5+9=bold highlight]
  {_    u  u  _   _  _  | _    _  u  u   _ }
  {flos ve-te-ris vi-ni | meis na-ri-bus ob}
```



Example

Mixing and combining styles is possible too.

```
\metricsymbols[1+4=bold highlight, 3=colored highlight]
  {u_uu x _ || u _ n ||} \\
\metricsymbols[2={bold highlight,colored highlight}]
  {u_uu x _ || u _ n ||}
```



Example

Add some superscripts to the breaks.

```
\metricsymbols[6={superscript=5},10={colored highlight,superscript=bD}]
   {_ _uu _ _uu _ | _uu _ _uu || _ uu _ u_}
```

5 Accents and bows

`\brv` ⋆ `\lng` ⋆ `\acct` ⋆

`\brv{`⟨*vowel*⟩`}` `\lng{`⟨*vowel*⟩`}` `\acct{`⟨*vowel*⟩`}`

The first commands offer an alternative to the standard accent macros `\u` and `\=`. The difference is that `\brv` centers the accent above the vowel or diphthong and `\lng` stretches the bar across the whole vowel or diphthong. `\acct` adds an accent dot below a vowel or dipthtong.[2]

Example

Add accents to all vowels.

```
\brv{a}m\acct{\lng{i}}c\brv{u}s pr\acct{\brv{o}}f\brv{u}g\brv{u}s
```

[2] Actually you can use any vowel, diphtong, syllable or word as ⟨*vowel*⟩, it makes no difference as long as it is text.

ămi̱cŭs prŏfŭgŭs

mětrix also tries to do some kind of italic correction, and shifts the accents a little to the right when an italic or slanted font is used.

ŭ	*ŭ*	*ŭ*	ı̆	*ı̆*	*ı̆*	aĕ	*aĕ*	*aĕ*	ŭ	*ŭ*	*ŭ*	ı̆	*ı̆*	*ı̆*	aĕ	*aĕ*	*aĕ*
ū	*ū*	*ū*	ī	*ī*	*ī*	ae͞	*ae͞*	*ae͞*	ū	*ū*	*ū*	ī	*ī*	*ī*	ae͞	*ae͞*	*ae͞*
ụ	*ụ*	*ụ*	ị	*ị*	*ị*	aẹ	*aẹ*	*aẹ*	ụ	*ụ*	*ụ*	ị	*ị*	*ị*	aẹ	*aẹ*	*aẹ*

Fine Tuning

To make some fine tuning for a certain accent possible the three macros actually got some additional, *optional* arguments:

`\brv(`⟨*coordinate*⟩`){`⟨*vowel*⟩`}`
`\lng(`⟨*coordinate*⟩`)[`⟨*left length*⟩`]{`⟨*vowel*⟩`}[`⟨*right length*⟩`]`
`\acct(`⟨*coordinate*⟩`){`⟨*vowel*⟩`}`

Where ⟨*coordinate*⟩ must be a valid Ti*k*Z coordinate and can be used to move the accent. In addition to that the accent produced by `\lng` can be extended with ⟨*left*/*right length*⟩ by a certain amount.

Example

Prevent collision between accent and descender of an *f*.

```
\itshape somn\acct(-0.05em,-0.45ex){i}fero
```

somnifero

`\bow` ⋆ `\bow{`⟨*syllables*⟩`}`

`\bow` can be used to show the contraction of two vowels or syllables.

Example

```
mult\bow{um i}lle or d\bow{ei}nde
```

multum ille or deinde

Fine Tuning

To make some fine tuning for a certain bow possible the macro actually has some additional, *optional* arguments:

`\bow(`⟨*coordinate*⟩`)[`⟨*left length*⟩`]{`⟨*syllable*⟩`}[`⟨*right length*⟩`]`

Where ⟨*left*/*right length*⟩ can be used to shorten the bow by a certain amount.

Example

Prevent collision between accent and bow.

```
c\acct{oe}-l\bow{um \acct{e}}[2pt]st
```

coe-lum est

6 Environments

`symbolline`

This environment can be used to display a line of stand alone symbols.

Example

```
Text text text ...
\begin{symbolline}
   \metricsymbols{oo e _ u u _ e u _ e u _ u_}
\end{symbolline}
Text text text ...
```

Text text text ...

Text text text ...

`metricverses`

`\begin{metricverses}[`⟨*source*⟩`]`
⟨*content optional* `\verseref{`⟨*reference*⟩`}`⟩
`\end{metricverses}`

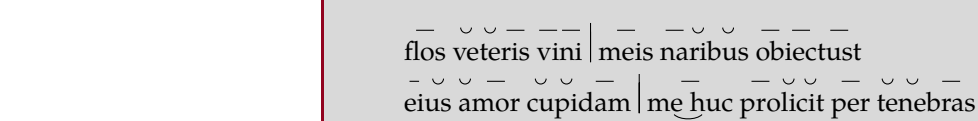
Use this environment to display a verse with metric symbols, separate multiple verses by a blank line.

Example

```
Text text text ...
\begin{metricverses}
   \metrics{_    u  u  _   _  _  | _    _  u  u   _  _   _   }
           {flos ve-te-ris vi-ni | meis na-ri-bus ob-iec-tust}

   \metrics{_  u  u _   u  u  _   | _             _   u  u
                                                  _   u  u  _   }
           {ei-us a-mor cu-pi-dam | {m\bow{e h}uc} pro-li-cit
                                                  per te-ne-bras}
\end{metricverses}
Text text text ...
```

Text text text ...



Text text text ...

\verseref `\verseref{⟨reference⟩}`

Inside of `{metricverses}` you may use `\verseref` to print a reference.

Example

```
Text text text ...
\begin{metricverses}
   \metrics{_    u  u  _   _  _  | _    _  u  u   _  _   _   }
           {flos ve-te-ris vi-ni | meis na-ri-bus ob-iec-tust}
   \verseref{Plaut. \emph{Curc.} 96f}

   \metrics{_  u  u _   u  u  _   | _             _   u  u
                                                  _   u  u  _   }
           {ei-us a-mor cu-pi-dam | {m\bow{e h}uc} pro-li-cit
                                                  per te-ne-bras}
\end{metricverses}
Text text text ...
```

Text text text ...

flos veteris vini | meis naribus obiectust — Plaut. *Curc.* 96f

eius amor cupidam | me huc prolicit per tenebras

Text text text ...

7 FAQs

7.1 How can I display the symbols below the syllables?

Change the variable `symbolshift` to a negative value.

Example

```
\setmetrixvar{symbolshift}{-0.6em}
% later ...
\metrics{_    u  u  _   _  _  | _    _  u  u   _  _   _   }
        {flos ve-te-ris vi-ni | meis na-ri-bus ob-iec-tust}
```

flos veteris vini | meis naribus obiectust

7.2 How can I combine two words below one symbol?

Use braces `{}` in the lists to keep them processed as one element.

Example

```
\metrics{u  u  _   | _             _   u  u   }
        {cu-pi-dam | {m\bow{e h}uc} pro-li-cit }
```

cupidam | me huc prolicit

7.3 How can I show a hyphen character?

To escape a hyphen - put it inside braces, but you must still add an unbraced hypen to show **mětrix** where your syllables split.

Example

If you enclose the hyphen in braces together with a syllable, the symbol gets centered above both.

```
\metrics{_     _  }
        {vi-{-ni}}
```

vi-ni

You can enclose only the hyphen in braces and treat it as a syllable but then you must add an empty symbol `e` too.

```
\metrics{_   e  _ }
        {vi-{-}-ni}
```

vi-ni

7.4 How can I use quotes in `\metrics`?

It should be possible to use all shorthands (or direct input with Unicode) etc. for quotation marks except the active quotes of csquotes, which won't work inside the `\metrics` syllable list. It is possible to use csquotes besides **mětrix** though.

Example

```
\metrics{ _   u }{ ``si me'' }
\metrics{ _   u }{ \glqq si me\grqq }% with \usepackage[<lang>]{babel}
\metrics{ _   u }{ "`si me"' }% with \usepackage[ngerman]{babel}
```

“si me” „si me“ „si me“

7.5 How can I add a superscript letter to a certain symbol?

Use the `superscript` highlighting style as described above.

7.6 How can I make subscripts instead of superscripts?

The easiest way is to use the `superscript` style and change a part of its definition to shift the superscripts to subscript positions.

Example

```
\metricsymbols[2={superscript=x}]{ u || u } \qquad vs. \qquad
% ...
\tikzset{
```

```
   every superscript picture/.style={
      baseline=1ex,
   },
}
% ...
\metricsymbols[2={superscript=x}]{ u || u }
```

$\smile \|^{x} \smile$ vs. $\smile \|_{x} \smile$

Normally the `\tikzset` should be part of your preamble, I used it this way to show the differences.

7.7 How can I highlight all symbols/syllables?

Way 1 Just call your desired highlighting style before using on of the macros `\metrics` or `\metricsymbols`. You may enclose this in a group to not affect the other following sequences. Mind that the higlighting styles must be in a way changing the `every ...` styles to make this way work.

Example

```
{% begin group
   \tikzset{colored highlight}
   \metrics{_    u  u  _   _  _ }
           {flos ve-te-ris vi-ni}
}% end group
```

flos veteris vini

Way 2 Change the `every metrix ...` styles.

Example

```
{% begin group
   \tikzset{every metrix symbol/.append style={red}}
   \metrics{_    u  u  _   _  _ }
           {flos ve-te-ris vi-ni}
}% end group
```

flos veteris vini

Leave out the grouping (and put this to your preamble) if yout want to highlight the symbols in your whole document.

7.8 How can I change the size of a symbol?

Change the two base vector units.

Example

```
\setmetrixvar{baseunit}{1em}
\setmetrixvar{bigbaseunit}{1.6em}
```

If you want to change the size of a single symbol to highlight it you must create your own highlighting style.

Example

```
\tikzset{
 bigger highlight/.style={
  every metrix symbol/.append style={x=2.5em,y=2.5em,line width=1.5pt},
 },
}
% later
\metricsymbols[2=bigger highlight]{u_uu x _ || u _ n x}
```

7.9 How can I stop highlighting the syllables too?

Way 1 Change the highlight styles (in your preamble).

Example

```
\tikzset{
   colored highlight/.style={
      every metrix symbol/.append style={
         draw=\usemetrixvar{highlightcolor},
      },
   },
}
% later ...
\metrics[3=colored highlight]{_    u  u  _   _  _ }
                             {flos ve-te-ris vi-ni}
```


flos veteris vini

Way 2 Create your own highlighting style, which is very similar to way 1, as the following example shows. Every own style should change the apperance by appending the settings to one of the `every ...` styles.

Example

```
\tikzset{
   my highlight/.style={
      every metrix symbol/.append style={draw=blue,line width=0.07em},
   }
}
\metrics[5=my highlight]{_    u  u  _   _  _ }
                        {flos ve-te-ris vi-ni}
```

flos veteris vini

7.10 Why got the highlight styles that long names?

To prevent conflict with other packages.

Example

If you want to shorten it create your own style as described above or use

```
\tikzset{
 hl/.style={colored highlight}
}
```

to map the style to a shorter name. Then you can use it like in

```
\metricsymbols[2=hl]{u _ _ u}
```

7.11 How can I change the font of all syllables?

Extend the `every metrix syllable node` style

Example

Print all syllables in italic with the following extension.

```
\tikzset{
   every metrix syllable node/.append syte={font=\itshape},
}
```

8 Customization

Some hints were already given in the FAQ section (see section 7) but here I will list all variables and Ti*k*Z styles that are in use and can be changed to customize **mětrix** easily.

8.1 Variables

`\setmetrixvar` `\usemetrixvar`

`\setmetrixvar{`⟨*variable*⟩`}{`⟨*value*⟩`}`

To customize the rendering of the symbols, accents and bow **mětrix** has some variables that you can change. Use `\setmetrixvar` to change a value. The variables and the default values are listed in table 2. To access a value you can use `\usemetrixvar{`⟨*variable*⟩`}`.

It is highly recommended to use font size depending units, i.e. `em` or `ex`, for all lengthen to keep the symbols usable in different font sizes, for example in headlines or footnotes.

Example

Change the highlighting color to blue.

```
\setmetrixvar{higlightcolor}{blue}
% later
```

```
\metrics[5=colored highlight]{_    u  u  _   _  _ }
                             {flos ve-te-ris vi-ni}
```
flos veteris vini

Example

Create your own highlighting style but use the default highlighting color.

```
\tikzset{
   my highlight/.style={
      every metrix symbol/.append style={
         draw=\usemetrixvar{highlightcolor},
         line width=0.15em
      },
   },
}
\metrics[5=my highlight]{_    u  u  _   _  _ }
                        {flos ve-te-ris vi-ni}
```
flos veteris vini

Table 2: Variables

variable	default	explanation
`symbollinewidth`	`0.04em`	line width of symbols above syllables and small stand alone symbols
`bigsymbollinewidth`	`0.06em`	line width of big stand alone symbols
`accentlinewidth`	`0.03em`	line width of accents (`\lng` and `\brv`)
`bowlinewidth`	`0.03em`	line width of bows (`\bow`)
`symbolsep`	`0.4em`	gap between symbols in stand alone lists
`baseunit`	`0.9em`	length of the base vector for drawing symbols above syllables, small stand alone symbols, accents and bows
`bigbaseunit`	`1.4em`	length of the base vector for drawing stand alone symbols
`shortsyllablelimit`	`0.8em`	all syllables shorter than this can be treated specially, e.g. they'll get a shorter elementum longum.
`gap`	`0.09em`	small gap between lines of the symbols, e.g. the distance between the two lines of a verse break
`symbolshift`	`1.1em`	leght to shift the symbols above or below the syllables (try `-0.6em` to display the symbols below the base line)
`lngshift`	`0.8em`	length to shift the longa accent

...

Table 2: Variables (cont.)

variable	default	explanation
lngshortening	0.075em	length to shorten the longa accent a little
lngminlength	0.25em	minimum width of a longa accent
brvshift	0.9em	length to shift the brevis accent
dotshift	-0.15em	length to shift the dot accent
itcorrection	0.11em	length to shift the accents above italic/slanted letters
accentxshift	-0.025em	length to shift the accents horizontally
bowshift	-0.15em	length to shift the bow below the base line
bowshortening	0.15em	length to shrink the bow a little
bowlooseness	0.75	value to influence the bending of the bow
symbolcolor	. (curr. color)	color of metric symbols
accentcolor	. (curr. color)	color of accents (\lng and \brv)
bowcolor	. (curr. color)	color of bows (\bow)
highlightcolor	red	color of highlighted symbols and syllabels used in colored highlight style
fillcolor	yellow	color of filled symbol nodes used in filled highlight style
arrow	$\downarrow$	arrow for highlighting
breakgap	0.6em	gap before and after a (verse) break
emptywidth	1em	gap replacing an empty symbol (abbreviation e)
acute	\textasciiacute	acute accent character/symbol
grave	\textasciigrave	grave accent character/symbol

8.2 Ti*k*Z styles

Beside the variables you may change the Ti*k*Z styles used by **mětrix**. But please mind that all styles are not empty by default so you should prefer /.append style against /.style. Otherwise it may cause strange effects. Remind that you can use \usemetrixvar to access a variable.

every␣metrix␣symbol
every␣metrix␣big␣symbol
every␣metrix␣symbol␣node

These three styles define the apperance of the metric symbols. They define the line width, the color, the basis vectors and other things.

every␣metrix␣syllable␣node
every␣metrix␣break␣node

These styles defines the nodes in which a syllable or a break symbol (the ones spanning across the symbol and the syllable line) is typeset, e.g. it aligns these nodes at their base line.

every␣metrix␣ictus␣node	This style defines the apperance of accents above symbols (ictus, e.g. /u).
every␣metrix␣accent	This style defines the apperance of accents created by \lng and \brv.
every␣metrix␣bow	This style defines the apperance of bows below symbols.
bold␣highlight colored␣highlight dashed␣highlight filled␣highlight superscript	These styles can be used to highlight a certain symbol.
every␣superscript␣picture every␣superscript␣node every␣superscript␣label	These styles are used to define the superscript highlighting style.
every␣metrix␣added␣text	This style is used for text added to a symbol with the add text highlighting.

9 Implementation

```
⟨*package⟩
⟨@@=metrix⟩
\ProvidesExplPackage
  {\metrixFileName}{\metrixFileDate}{\metrixFileVersion}{\metrixFileDescription}
```

9.1 Required packages

```
\RequirePackage{xparse}
\RequirePackage{xpatch}
\RequirePackage{tikz}
\ExplSyntaxOff
 \usetikzlibrary{calc}
\ExplSyntaxOn
\RequirePackage{textcomp}
```

9.2 Constants and internal variables

These constants variables are for internal use only and can't be changed/accessed by the user.

9.2.1 Constants

\c__metrix_acute_accent_tl Constant holding an acute accent char.

```
\tl_const:Nn \c__metrix_acute_accent_tl { / }
```

(*End definition for* `\c__metrix_acute_accent_tl`.)

\c__metrix_acute_accent_tl Constant holding an grave accent char.

```
\tl_const:Nn \c__metrix_grave_accent_tl { * }
```

(*End definition for* `\c__metrix_acute_accent_tl`.)

9.2.2 Variables

\l__metrix_words_tl This list stores the words of the `\metrics` macro.

```
\tl_new:N \l__metrix_words_tl
```

(*End definition for* `\l__metrix_words_tl`.)

\l__metrix_syllables_seq This list stores the words of the `\l__metrix_words_tl` list.

```
\seq_new:N \l__metrix_syllables_seq
```

(*End definition for* `\l__metrix_syllables_seq`.)

\l__metrix_symbols_seq This list stores the metric symbols of `\metrics` and `\metricsymbols`.

```
\seq_new:N \l__metrix_symbols_seq
```

(*End definition for* `\l__metrix_symbols_seq`.)

\l__metrix_short_breaks_seq This list stores the short and foot breaks of `\metrics`.
\l__metrix_foot_breaks_seq

```
\seq_new:N \l__metrix_short_breaks_seq
\seq_new:N \l__metrix_foot_breaks_seq
```

(*End definition for* `\l__metrix_short_breaks_seq` *and* `\l__metrix_foot_breaks_seq`.)

\l__metrix_highlights_prop This list stores the higlighting styles of `\metrics` and `\metricsymbols`.

```
\prop_new:N \l__metrix_highlights_prop
```

(*End definition for* `\l__metrix_highlights_prop`.)

\l__metrix_highlight_seq This lists are used to evaluate a higlight style.
\l__metrix_highlight_pos_seq

```
\seq_new:N \l__metrix_highlight_seq
\seq_new:N \l__metrix_highlight_pos_seq
```

(*End definition for* `\l__metrix_highlight_seq` *and* `\l__metrix_highlight_pos_seq`.)

\q__metrix_space_marker This is the marker for spaces inside of the `\l__metrix_words_tl` list.

```
\quark_new:N \q__metrix_space_marker
```

(*End definition for* `\q__metrix_space_marker`.)

\l__metrix_process_int This process counter is used to combine the symbols and syllables.

```
\int_new:N \l__metrix_process_int
```

(*End definition for* `\l__metrix_process_int`.)

\l__metrix_short_syllable_bool
\l__metrix_syllable_box
\g__metrix_variable_shortsyllablelimit_tl

This boolean can be used to store that a syllable is short, e.g. *li* will be defined as short wheras *man* is long. That will be used to shorten the |_| symbol. Furthermore we'll need a box to measure the length of a syllable and a variable to save the limit for short syllables.

```
\bool_new:N \l__metrix_short_syllable_bool
\box_new:N \l__metrix_syllable_box
\tl_new:N \g__metrix_variable_shortsyllablelimit_tl
\tl_set:Nn \g__metrix_variable_shortsyllablelimit_tl { 0.8em }
```

(*End definition for* `\l__metrix_short_syllable_bool`, `\l__metrix_syllable_box`, *and* `\g__metrix_variable_shortsyllablelimit_tl`.)

\l__metrix_current_symbol_tl

Variable holding the current symbol abbreviation.

```
\tl_new:N \l__metrix_current_symbol_tl
```

(*End definition for* `\l__metrix_current_symbol_tl`.)

\l__metrix_current_symbol_head_tl

Variable holding firt char of the current symbol abbreviation.

```
\tl_new:N \l__metrix_current_symbol_head_tl
```

(*End definition for* `\l__metrix_current_symbol_head_tl`.)

\l__metrix_has_acute_accent_bool

Bool for switching on an acute accent of a symbol.

```
\bool_new:N \l__metrix_has_acute_accent_bool
```

(*End definition for* `\l__metrix_has_acute_accent_bool`.)

\l__metrix_has_grave_accent_bool

Bool for switching on an grave accent of a symbol.

```
\bool_new:N \l__metrix_has_grave_accent_bool
```

(*End definition for* `\l__metrix_has_grave_accent_bool`.)

9.3 User variables

These variables save things, the user can change via `\setmetrixvar` and use via `\usemetrixvar`.

\g__metrix_variable_symbollinewidth_tl

This variable stores the line width for all metric symbols above (or below) syllables.

```
\tl_new:N \g__metrix_variable_symbollinewidth_tl
\tl_set:Nn \g__metrix_variable_symbollinewidth_tl { 0.04em }
```

(*End definition for* `\g__metrix_variable_symbollinewidth_tl`.)

\g__metrix_variable_bigsymbollinewidth_tl

This variable stores the line width for all stand alone metric symbols.

```
\tl_new:N \g__metrix_variable_bigsymbollinewidth_tl
\tl_set:Nn \g__metrix_variable_bigsymbollinewidth_tl { 0.06em }
```

(*End definition for* `\g__metrix_variable_bigsymbollinewidth_tl`.)

\g__metrix_variable_accentlinewidth_tl

This variable stores the line width of the accent like symbols.

```
\tl_new:N \g__metrix_variable_accentlinewidth_tl
\tl_set:Nn \g__metrix_variable_accentlinewidth_tl { 0.04em }
```

(*End definition for* `\g__metrix_variable_accentlinewidth_tl`.)

\g__metrix_variable_bowlinewidth_tl This variable stores the line width of the bow.

```
\tl_new:N \g__metrix_variable_bowlinewidth_tl
\tl_set:Nn \g__metrix_variable_bowlinewidth_tl { 0.04em }
```

(*End definition for* `\g__metrix_variable_bowlinewidth_tl`.)

\g__metrix_variable_symbolsep_tl This variable stores the gap between two or more stand alone metric symbols.

```
\tl_new:N \g__metrix_variable_symbolsep_tl
\tl_set:Nn \g__metrix_variable_symbolsep_tl { 0.4em }
```

(*End definition for* `\g__metrix_variable_symbolsep_tl`.)

\g__metrix_variable_baseunit_tl This variable stores the length of the basis vector for all metric symbols above (or below) syllables and accent like symbols.

```
\tl_new:N \g__metrix_variable_baseunit_tl
\tl_set:Nn \g__metrix_variable_baseunit_tl { 0.9em }
```

(*End definition for* `\g__metrix_variable_baseunit_tl`.)

\g__metrix_variable_bigbaseunit_tl This variable stores the length of the basis vector for all stand alone metric symbols.

```
\tl_new:N \g__metrix_variable_bigbaseunit_tl
\tl_set:Nn \g__metrix_variable_bigbaseunit_tl { 1.4em }
```

(*End definition for* `\g__metrix_variable_bigbaseunit_tl`.)

\g__metrix_variable_gap_tl Length for small gaps in the symbols, e.g. the gap between the two bows of an elementum biceps.

```
\tl_new:N \g__metrix_variable_gap_tl
\tl_set:Nn \g__metrix_variable_gap_tl { 0.09em }
```

(*End definition for* `\g__metrix_variable_gap_tl`.)

\g__metrix_variable_symbolshift_tl This variable stores the value to shift metric symbols above (or below) syllables. Set this variable to approx `1.1em` to draw the symbols above the syllable and to `-0.6em` to draw them below.

```
\tl_new:N \g__metrix_variable_symbolshift_tl
\tl_set:Nn \g__metrix_variable_symbolshift_tl { 1.1em }
```

(*End definition for* `\g__metrix_variable_symbolshift_tl`.)

\g__metrix_variable_lngshift_tl This variable stores the value to shift the longa accent.

```
\tl_new:N \g__metrix_variable_lngshift_tl
\tl_set:Nn \g__metrix_variable_lngshift_tl { 0.15em }
```

(*End definition for* `\g__metrix_variable_lngshift_tl`.)

\g__metrix_variable_lngshortening_tl This variable stores the value to shorten the longa accent.

```
\tl_new:N \g__metrix_variable_lngshortening_tl
\tl_set:Nn \g__metrix_variable_lngshortening_tl { 0.075em }
```

(*End definition for* `\g__metrix_variable_lngshortening_tl`.)

\g__metrix_variable_lngminlength_tl This variable stores the value to shorten the longa accent.

```
\tl_new:N \g__metrix_variable_lngminlength_tl
\tl_set:Nn \g__metrix_variable_lngminlength_tl { 0.25em }
```

(*End definition for* `\g__metrix_variable_lngminlength_tl`.)

`\g__metrix_variable_brvshift_tl` This variable stores the value to shift the brevis accent.

```
\tl_new:N \g__metrix_variable_brvshift_tl
\tl_set:Nn \g__metrix_variable_brvshift_tl { 0.25em }
```

(*End definition for* `\g__metrix_variable_brvshift_tl`.)

`\g__metrix_variable_dotshift_tl` This variable stores the value to shift the brevis accent.

```
\tl_new:N \g__metrix_variable_dotshift_tl
\tl_set:Nn \g__metrix_variable_dotshift_tl { -0.15em }
```

(*End definition for* `\g__metrix_variable_dotshift_tl`.)

`\g__metrix_variable_itcorrection_tl` These variables are used to set the italic correction of accents.
`\l__metrix_internal_itcorrection_tl`
`\g__metrix_internal_itcorrection_zero_tl`

```
\tl_new:N \g__metrix_variable_itcorrection_tl
\tl_set:Nn \g__metrix_variable_itcorrection_tl { 0.11em }
\tl_new:N \l__metrix_internal_itcorrection_tl
\tl_set:Nn \l__metrix_internal_itcorrection_tl { 0em }
\tl_new:N \g__metrix_internal_itcorrection_zero_tl
\tl_set:Nn \g__metrix_internal_itcorrection_zero_tl { 0em }
```

(*End definition for* `\g__metrix_variable_itcorrection_tl`, `\l__metrix_internal_itcorrection_tl`, *and* `\g__metrix_internal_itcorrection_zero_tl`.)

`\g__metrix_variable_accentxshift_tl` This variable ise used to shift the accents horizontally.

```
\tl_new:N \g__metrix_variable_accentxshift_tl
\tl_set:Nn \g__metrix_variable_accentxshift_tl { -0.025em }
```

(*End definition for* `\g__metrix_variable_accentxshift_tl`.)

`\g__metrix_variable_bowshift_tl` This variable stores the value to shift the bow.

```
\tl_new:N \g__metrix_variable_bowshift_tl
\tl_set:Nn \g__metrix_variable_bowshift_tl { -0.15em }
```

(*End definition for* `\g__metrix_variable_bowshift_tl`.)

`\g__metrix_variable_bowshortening_tl` This variable stores the value to shrink the bow.

```
\tl_new:N \g__metrix_variable_bowshortening_tl
\tl_set:Nn \g__metrix_variable_bowshortening_tl { 0.15em }
```

(*End definition for* `\g__metrix_variable_bowshortening_tl`.)

`\g__metrix_variable_bowlooseness_tl` This variable stores the value to shrink the bow.

```
\tl_new:N \g__metrix_variable_bowlooseness_tl
\tl_set:Nn \g__metrix_variable_bowlooseness_tl { 0.75 }
```

(*End definition for* `\g__metrix_variable_bowlooseness_tl`.)

`\g__metrix_variable_symbolcolor_tl` These variables store the color of symbols, accents and bows.
`\g__metrix_variable_accentcolor_tl`
`g__metrix_variable_bowcolor_tl`

```
\tl_new:N \g__metrix_variable_symbolcolor_tl
\tl_set:Nn \g__metrix_variable_symbolcolor_tl { . }
\tl_new:N \g__metrix_variable_accentcolor_tl
\tl_set:Nn \g__metrix_variable_accentcolor_tl { . }
\tl_new:N \g__metrix_variable_bowcolor_tl
\tl_set:Nn \g__metrix_variable_bowcolor_tl { . }
```

(*End definition for* `\g__metrix_variable_symbolcolor_tl`, `\g__metrix_variable_accentcolor_tl`, *and* `g_-_metrix_variable_bowcolor_tl`.)

\g__metrix_variable_highlightcolor_tl These variable stores the color used in the `colored highlight` style.

```
\tl_new:N \g__metrix_variable_highlightcolor_tl
\tl_set:Nn \g__metrix_variable_highlightcolor_tl { red }
```

(*End definition for* `\g__metrix_variable_highlightcolor_tl`.)

\g__metrix_variable_fillcolor_tl These variable stores the color used in the `filled highlight` style.

```
\tl_new:N \g__metrix_variable_fillcolor_tl
\tl_set:Nn \g__metrix_variable_fillcolor_tl { yellow }
```

(*End definition for* `\g__metrix_variable_fillcolor_tl`.)

\g__metrix_variable_arrow_tl These variable stores the color used in the `filled highlight` style.

```
\tl_new:N \g__metrix_variable_arrow_tl
\tl_set:Nn \g__metrix_variable_arrow_tl { $\downarrow$ }
```

(*End definition for* `\g__metrix_variable_arrow_tl`.)

\g__metrix_variable_breakgap_tl This variable stores the width of the gap around the two break symbols.

```
\tl_new:N \g__metrix_variable_breakgap_tl
\tl_set:Nn \g__metrix_variable_breakgap_tl { 0.6em }
```

(*End definition for* `\g__metrix_variable_breakgap_tl`.)

\g__metrix_variable_emptywidth_tl This variable stores the width of the gap caused by an empty symbol (abbreviation `e`).

```
\tl_new:N \g__metrix_variable_emptywidth_tl
\tl_set:Nn \g__metrix_variable_emptywidth_tl { 1em }
```

(*End definition for* `\g__metrix_variable_emptywidth_tl`.)

\g__metrix_variable_acute_tl This variable stores acute symbol.

```
\tl_new:N \g__metrix_variable_acute_tl
\tl_set:Nn \g__metrix_variable_acute_tl { \textasciiacute }
```

(*End definition for* `\g__metrix_variable_acute_tl`.)

\g__metrix_variable_grave_tl This variable stores grave symbol.

```
\tl_new:N \g__metrix_variable_grave_tl
\tl_set:Nn \g__metrix_variable_grave_tl { \textasciigrave }
```

(*End definition for* `\g__metrix_variable_grave_tl`.)

9.4 Variants

Later we'll need the following variant.

```
\cs_generate_variant:Nn \prop_item:Nn { No , Nf , NV , Nx }
\cs_generate_variant:Nn \prop_put:Nnn { Nnx , Nxx , Nff  , Noo }
\cs_generate_variant:Nn \seq_item:Nn { Nf , NV , Nx }
\cs_generate_variant:Nn \seq_set_split:Nnn { Nnf , NnV , Nnx }
\cs_generate_variant:Nn \tl_remove_once:Nn { NV }
```

9.5 Internal main macros

__metrix_metrics:nn

This macro processes the two lists of \metrics and combines the symbols and syllables.[3]

```
\cs_new_protected:Npn \__metrix_metrics:nn #1 #2
 {
  \tl_set:Nx \l__metrix_words_tl { \tl_trim_spaces:n { #2 } }
```

First replace the spaces by a special marker \q__metrix_space_marker and add hyphens: a space becomes a syllable.

```
  \tl_replace_all:Nnn \l__metrix_words_tl { ~ } { - \q__metrix_space_marker - }
```

Then split the word list at hypens.

```
  \seq_set_split:NnV \l__metrix_syllables_seq { - } \l__metrix_words_tl
```

Split the symbol list at spaces.

```
  \seq_set_split:Nnx \l__metrix_symbols_seq { ~ } { \tl_trim_spaces:n { #1 } }
```

Search for the short and foot breaks and remove them afterwards.

```
  \int_zero:N \l__metrix_process_int
  \seq_clear:N \l__metrix_short_breaks_seq
  \seq_clear:N \l__metrix_foot_breaks_seq
  \seq_map_inline:Nn \l__metrix_symbols_seq {
    \int_incr:N \l__metrix_process_int
    \tl_if_eq:nnT { ##1 } { ' } {
      \seq_put_right:Nx \l__metrix_short_breaks_seq { \int_use:N \l__metrix_process_int }
      \int_decr:N \l__metrix_process_int
    }
    \tl_if_eq:nnT { ##1 } { , } {
      \seq_put_right:Nx \l__metrix_foot_breaks_seq { \int_use:N \l__metrix_process_int }
      \int_decr:N \l__metrix_process_int
    }
  }
  \seq_remove_all:Nn \l__metrix_symbols_seq { , }
  \seq_remove_all:Nn \l__metrix_symbols_seq { ' }
```

Test whether both lists got the same length:

```
  \int_zero:N \l__metrix_process_int
  \seq_map_inline:Nn \l__metrix_syllables_seq
   {
    \tl_if_eq:nnT { ##1 } { \q__metrix_space_marker }
     { \int_incr:N \l__metrix_process_int }
   }
  \int_compare:nTF
   {
    \seq_count:N \l__metrix_syllables_seq -
      \seq_count:N \l__metrix_symbols_seq = \l__metrix_process_int
   }
   {
```

[3] The framing of this macro was provided by Enrico Gregorio at http://tex.stackexchange.com/q/124528/4918, a follow up question was http://tex.stackexchange.com/q/124698/4918. David Carlisle and Bruno Le Floch lead me to the implementation of the highlighting mechanism, see http://tex.stackexchange.com/q/124782/4918

continue with list processing, if the numbers are equal:

```
    \int_zero:N \l__metrix_process_int
    \seq_map_inline:Nn \l__metrix_syllables_seq
     {
      \int_incr:N \l__metrix_process_int
      \tl_if_eq:nnTF { ##1 } { \q__metrix_space_marker }
       {
```

If the syllable is a space the process counter must be decremented and a space is typeset.

```
        \int_add:Nn \l__metrix_process_int { -1 }
        \c_space_token
       }
       {
```

Finally typeset the syllable and it's symbol.

```
        \str_case:nnF { ##1 }
         {
          { | }
           {
            \__metrix_break_node:n { \__metrix_l_break: }
           }
          { || }
           {
            \__metrix_break_node:n { \__metrix_ll_break: }
           }
         }
         {
          \__metrix_print_syllable:n { ##1 }
         }
       }
     }
```

And add the short break symbols if necessary:

```
    \seq_if_empty:NF \l__metrix_short_breaks_seq {
      \seq_map_inline:Nn \l__metrix_short_breaks_seq {
        \int_set:Nn \l_tmpa_int { ##1 - 1 }
        \bool_if:nF {
          \int_compare_p:n
            { 0 = \l_tmpa_int }
          ||
          \int_compare_p:n
            { \seq_count:N \l__metrix_symbols_seq = \l_tmpa_int }
        } {
          \tikz [remember~picture, overlay] {
            \node [every~metrix~symbol~node] at
              ($(l__metrix_symbol_node_\int_use:N \l_tmpa_int.east)!
               0.5!(l__metrix_symbol_node_##1.west)$)
              { \__metrix_short_break: };
          }
        }
      }
    }
```

And add the foot break symbols if necessary:

```
    \seq_if_empty:NF \l__metrix_foot_breaks_seq {
```

```
      \seq_map_inline:Nn \l__metrix_foot_breaks_seq {
        \int_set:Nn \l_tmpa_int { ##1 - 1 }
        \bool_if:nF {
          \int_compare_p:n
            { 0 = \l_tmpa_int }
          ||
          \int_compare_p:n
            { \seq_count:N \l__metrix_symbols_seq = \l_tmpa_int }
        } {
          \tikz [remember~picture, overlay] {
            \coordinate (l__metrix_tmp_coord) at
              ($(l__metrix_syllable_node_\int_use:N \l_tmpa_int.east)!
                0.5!(l__metrix_syllable_node_##1.west)$);
            \node [every~metrix~symbol~node] at
              ($(l__metrix_symbol_node_\int_use:N \l_tmpa_int.east)!
               (l__metrix_tmp_coord)!(l__metrix_symbol_node_##1.west)$)
              { \__metrix_foot_break: };
          }
        }
      }
    }
```

Send an error, else.

```
    {
     \__metrix_error_msg:n
      {
       Numbers~of~symbols~(\seq_count:N \l__metrix_symbols_seq)~and~syllables~
       (\int_eval:n
         {
          \seq_count:N \l__metrix_syllables_seq - \l__metrix_process_int
         }
       )~mismatch.
      }
    }
  }
```

(*End definition for* `\__metrix_metrics:nn`.)

`\__metrix_metricsymbols:n` This macro works like `\__metrix_metrics` but is used to print stand alone metric symbols via `\metricsymbols`.

```
\cs_new_protected:Npn \__metrix_metricsymbols:n #1
 {
  \seq_set_split:Nnx \l__metrix_symbols_seq { ~ } { \tl_trim_spaces:n { #1 } }
  \int_zero:N \l__metrix_process_int
  \seq_map_inline:Nn \l__metrix_symbols_seq
   {
    \int_incr:N \l__metrix_process_int
    \int_compare:nT { \l__metrix_process_int > 1 }
     {
      \hspace { \usemetrixvar{symbolsep} }
     }
    \str_case:nnF { ##1 }
     {
      { , }
```

```
      {
       \__metrix_break_gap:
       \__metrix_align_symbol:n { \__metrix_l_bigmark:  }
       \__metrix_break_gap:
      }
     { ' }
      {
       \__metrix_break_gap:
       \__metrix_align_symbol:n { \__metrix_l_bigmark:  }
       \__metrix_break_gap:
      }
     { | }
      {
       \__metrix_break_gap:
       \__metrix_align_symbol:n { \__metrix_l_bigmark:  }
       \__metrix_break_gap:
      }
     { '' }
      {
       \__metrix_break_gap:
       \__metrix_align_symbol:n { \__metrix_ll_bigmark: }
       \__metrix_break_gap:
      }
     { || }
      {
       \__metrix_break_gap:
       \__metrix_align_symbol:n { \__metrix_ll_bigmark: }
       \__metrix_break_gap:
      }
    }
    {
     \__metrix_align_symbol:n { \__metrix_print_symbol: }
    }
  }
}
```

(*End definition for* `\__metrix_metricsymbols:n`.)

`\__metrix_print_syllable:n` This macro combines a single syllable and the corrosponding metric symbol taken frome the symbol list index with the process counter.

```
\cs_new_protected:Npn \__metrix_print_syllable:n #1
{
 \group_begin:
```

Check wether the current syllable is short or long and set the corresponding bbol.

```
  \hbox_set:Nn \l__metrix_syllable_box { #1 }
  \dim_compare:nTF
     { \box_wd:N \l__metrix_syllable_box < \g__metrix_variable_shortsyllablelimit_tl }
     { \bool_set_true:N \l__metrix_short_syllable_bool }
     { \bool_set_false:N \l__metrix_short_syllable_bool }
```

Set up the currend highlight is it is definded

```
  \cs_set:Npx \__metrix_current_highlight: {
   \prop_item:NV \l__metrix_highlights_prop \l__metrix_process_int
  }
  \expandafter\tikzset\expandafter{\__metrix_current_highlight:}
```

Finally print the syllable and the symbol above. Use {pgfinterruptboundingbox} so that the symbol doesn't takes space ad doesn't cause gaps between the syllables.

```
    \hbox_set:Nn \l_tmpa_box { \__metrix_print_symbol: }
    \begin{tikzpicture}
     [
      remember~picture,
      baseline=(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.base),
     ]
     \node [every~metrix~syllable~node]
        (l__metrix_syllable_node_\int_use:N \l__metrix_process_int)
        { #1 };
     \begin{pgfinterruptboundingbox}
      \node [every~metrix~symbol~node]
       (l__metrix_symbol_node_\int_use:N \l__metrix_process_int)
       at ($
          (l__metrix_syllable_node_\int_use:N \l__metrix_process_int.base)
          +
          (0,\usemetrixvar{symbolshift})
          +
          (\tl_use:N \l__metrix_internal_itcorrection_tl,0)
       $)
       { \box_use:N \l_tmpa_box };
     \end{pgfinterruptboundingbox}
    \end{tikzpicture}
   \group_end:
  }
```

(*End definition for* `\__metrix_print_syllable:n`.)

`\__metrix_print_symbol:` This command selects the right symbol by it's abbreviation.

```
 \cs_new_protected:Npn \__metrix_print_symbol:
  {
   \tl_set:Nx \l__metrix_current_symbol_tl
    {
     \seq_item:Nn \l__metrix_symbols_seq { \l__metrix_process_int }
    }
   \tl_set:Nx \l__metrix_current_symbol_head_tl
    {
     \tl_head:N \l__metrix_current_symbol_tl
    }
   \tl_case:NnT \l__metrix_current_symbol_head_tl
    {
     \c__metrix_acute_accent_tl { \bool_set_true:N \l__metrix_has_acute_accent_bool }
     \c__metrix_grave_accent_tl { \bool_set_true:N \l__metrix_has_grave_accent_bool }
    }
    {
     \tl_remove_once:NV \l__metrix_current_symbol_tl \l__metrix_current_symbol_head_tl
    }
   \cs_if_exist_use:cF
    {
     __metrix_
     \tl_use:N \l__metrix_current_symbol_tl
     _mark:
    }
```

```
    {
     \__metrix_error_msg:n
      {
       Unknown~symbol~abbreviation~'\tl_use:N \l__metrix_current_symbol_tl'.
      }
    }
  }
```

(*End definition for* `\__metrix_print_symbol:`.)

9.6 Internal auxiliary macros

`\__metrix_error_msg:n` An abbreviation to throw an error message.

```
\cs_new_protected:Npn \__metrix_error_msg:n #1
 {
  \PackageError{ \metrixFileName } { #1 }
   {
    Please~take~a~look~at~the~manual~or~send~an~email.
   }
 }
```

(*End definition for* `\__metrix_error_msg:n`.)

`\__metrix_warning_msg:n` An abbreviation to throw an error message.

```
\cs_new_protected:Npn \__metrix_warning_msg:n #1
 {
    \PackageWarning{ \metrixFileName } { #1 }
 }
```

(*End definition for* `\__metrix_warning_msg:n`.)

`\__metrix_align_symbol:n` This macro alings the metric symbols in a stand alone list.

```
\cs_new_protected:Npn \__metrix_align_symbol:n #1
 {
  \group_begin:
   \cs_set:Npx \__metrix_current_highlight: {
    \prop_item:NV \l__metrix_highlights_prop \l__metrix_process_int
   }
   \expandafter\tikzset\expandafter{\__metrix_current_highlight:}
   \begin{tikzpicture}
    [
     baseline={(0,-0.25*\usemetrixvar{baseunit})},
    ]
     \node [every~metrix~symbol~node] {#1};
   \end{tikzpicture}
  \group_end:
 }
```

(*End definition for* `\__metrix_align_symbol:n`.)

`\__metrix_break_gap:` This macro typesets the gap around the two break symbols.

```
\cs_new_protected:Npn \__metrix_break_gap:
 {
  \hspace { \usemetrixvar { breakgap } }
 }
```

(*End definition for* `\__metrix_break_gap:`.)

`\__metrix_break_node:n` This macro typsets the gap around the two break symbols.

```
\cs_new:Npn \__metrix_break_node:n #1
 {
  \group_begin:
   \cs_set:Npx \__metrix_current_highlight: {
    \prop_item:NV \l__metrix_highlights_prop \l__metrix_process_int
   }
   \expandafter\tikzset\expandafter{\__metrix_current_highlight:}
   \tikz[baseline=(l__metrix_break_node.base)]
    \node (l__metrix_break_node) [every~metrix~break~node] { #1 }
   ;
  \group_end:
 }
```

(*End definition for* `\__metrix_break_node:n`.)

`\__metrix_e_gap:` This macro typsets the gap around the two break symbols.

```
\cs_new_protected:Npn \__metrix_e_gap:
 {
  \hspace* { \usemetrixvar { emptywidth } }
 }
```

(*End definition for* `\__metrix_e_gap:`.)

`\__metrix_evaluate_higlights:N` This macro evaluates and prints the highlighting options.

```
\cs_new_protected:Npn \__metrix_evaluate_higlights:n #1
 {
```

Start with clearing the property list, otherwise the highlights from the last time will survive.

```
  \prop_clear:N \l__metrix_highlights_prop
```

Then spilt and process the argument as a comma separated list.

```
  \clist_map_inline:nn { #1 }
   {
```

The result is a sequence of key value pairs that we store in `\l__metrix_highlight_seq`. The first part of this sequence must be split again at the plus sign—store it in `\l__metrix_highlight_pos_seq`.

```
    \seq_set_split:Nnn \l__metrix_highlight_seq { = } { ##1 }
    \seq_set_split:Nnf \l__metrix_highlight_pos_seq { + }
     {
      \seq_item:Nn \l__metrix_highlight_seq { 1 }
     }
```

Process the `\l__metrix_highlight_pos_seq` list and set up the property list:

```
    \seq_map_inline:Nn \l__metrix_highlight_pos_seq
     {
      \prop_put:Nnx \l__metrix_highlights_prop
```

The key is the current item of `\l__metrix_highlight_pos_seq`.

```
       {
        ####1
       }
       {
```

The value is the second item of \l__metrix_highlight_seq.

```
        \seq_item:Nn \l__metrix_highlight_seq { 2 }
      }
    }
  }
}
```

(*End definition for* `\__metrix_evaluate_higlights:N`.)

9.7 Patching font macros

To apply the italic correction of the accents we need to patch the font switches.

```
\xpretocmd { \itshape }
 {
  \tl_set_eq:NN
   \l__metrix_internal_itcorrection_tl
   \g__metrix_variable_itcorrection_tl
 }
 { }
 {
  \__metrix_warning_msg:n { Could~not~patch~\string\itshape. }
 }
\xpretocmd { \slshape }
 {
  \tl_set_eq:NN
   \l__metrix_internal_itcorrection_tl
   \g__metrix_variable_itcorrection_tl
 }
 { }
 {
  \__metrix_warning_msg:n { Could~not~patch~\string\slshape. }
 }
\xpretocmd { \upshape }
 {
  \tl_set_eq:NN
   \l__metrix_internal_itcorrection_tl
   \g__metrix_internal_itcorrection_zero_tl
 }
 { }
 {
  \__metrix_warning_msg:n { Could~not~patch~\string\upshape. }
 }
\xpretocmd { \normalfont }
 {
  \tl_set_eq:NN
   \l__metrix_internal_itcorrection_tl
   \g__metrix_internal_itcorrection_zero_tl
 }
 { }
 {
  \__metrix_warning_msg:n { Could~not~patch~\string\normalfont. }
 }
```

9.8 Internal macros for metric symbols

__metrix_e_mark: The empty symbol.

```
\cs_new:Npn \__metrix_e_mark: { \__metrix_e_gap: }
```

(*End definition for* __metrix_e_mark:.)

__metrix_u_mark: The brevis symbol ◡.

```
\cs_new:Npn \__metrix_u_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) arc [start~angle=0, end~angle=180, radius=-0.225];
   \bool_if:NT \l__metrix_has_acute_accent_bool
    {
     \node [every~metrix~ictus~node] at (0.225,0) { \usemetrixvar { acute } };
    }
   \bool_if:NT \l__metrix_has_grave_accent_bool
    {
     \node [every~metrix~ictus~node] at (0.225,0) { \usemetrixvar { grave } };
    }
  \end{tikzpicture}
 }
```

(*End definition for* __metrix_u_mark:.)

__metrix___mark: The longa symbol —.

```
\cs_new:Npn \__metrix___mark:
 {
  \bool_if:NTF \l__metrix_short_syllable_bool
   {
    \begin{tikzpicture}[every~metrix~symbol]
     \draw (0,0) -- ++(0.4,0);
     \bool_if:NT \l__metrix_has_acute_accent_bool
      {
       \node [every~metrix~ictus~node] at (0.2,0) { \usemetrixvar { acute } };
      }
     \bool_if:NT \l__metrix_has_grave_accent_bool
      {
       \node [every~metrix~ictus~node] at (0.2,0) { \usemetrixvar { grave } };
      }
    \end{tikzpicture}
   }
   {
    \begin{tikzpicture}[every~metrix~symbol]
     \draw (0,0) -- ++(0.75,0);
     \bool_if:NT \l__metrix_has_acute_accent_bool
      {
       \node [every~metrix~ictus~node] at (0.375,0) { \usemetrixvar { acute } };
      }
     \bool_if:NT \l__metrix_has_grave_accent_bool
      {
       \node [every~metrix~ictus~node] at (0.375,0) { \usemetrixvar { grave } };
      }
    \end{tikzpicture}
   }
```

```
}
```

(*End definition for* `\__metrix___mark:`.)

`\__metrix_uu_mark:` The biceps symbol ⌣⌣.

```
\cs_new:Npn \__metrix_uu_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) arc [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0.4,0)+(\pgflinewidth,0)+(\usemetrixvar{gap},0)$) arc
    [start~angle=0, end~angle=180, radius=-0.2];
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_uu_mark:`.)

`\__metrix_uu__mark:` The biceps symbol ⌣⌣.

```
\cs_new:Npn \__metrix_uu__mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) arc [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0.4,0)+(\pgflinewidth,0)+(\usemetrixvar{gap},0)$) arc
    [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0,-0.2)+(-0.5\pgflinewidth,-\pgflinewidth)-(0,\usemetrixvar{gap})$) --
    ($(0.8,-0.2)+(1.5\pgflinewidth,-\pgflinewidth)
    +(\usemetrixvar{gap},-\usemetrixvar{gap})$);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_uu__mark:`.)

`\__metrix__uu_mark:` Another biceps symbol ⌣⌣.

```
\cs_new:Npn \__metrix__uu_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) arc [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0.4,0)+(\pgflinewidth,0)+(\usemetrixvar{gap},0)$) arc
    [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0,0)+(-0.5\pgflinewidth,0.5\pgflinewidth)+(0,\usemetrixvar{gap})$) --
    ($(0.8,0)+(1.5\pgflinewidth,0.5\pgflinewidth)
    +(\usemetrixvar{gap},\usemetrixvar{gap})$);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix__uu_mark:`.)

`\__metrix_u_uu_mark:` An another biceps symbol ⌣⌣.

```
\cs_new:Npn \__metrix_u_uu_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) arc [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0.4,0)+(\pgflinewidth,0)+(\usemetrixvar{gap},0)$) arc
    [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0,0)+(-0.5\pgflinewidth,0.5\pgflinewidth)+(0,\usemetrixvar{gap})$) --
    ($(0.8,0)+(1.5\pgflinewidth,0.5\pgflinewidth)
```

```
    +(\usemetrixvar{gap},\usemetrixvar{gap})$);
   \draw ($(0.2,0.2)+(0.5\pgflinewidth,1.5\pgflinewidth)
   +(0.5*\usemetrixvar{gap},2*\usemetrixvar{gap})$)
    arc [start~angle=0, end~angle=180, radius=-0.2];
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_u_uu_mark:`.)

`\__metrix_x_mark:` The anceps symbol ×.

```
\cs_new:Npn \__metrix_x_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (-0.2,0.2) -- (0.2,-0.2);
   \draw (-0.2,-0.2) -- (0.2,0.2);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_x_mark:`.)

`\__metrix_v_mark:` The (yet) unnamed symbol ⌄.

```
\cs_new:Npn \__metrix_v_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) -- (0.225,-0.225) -- (0.45,0);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_v_mark:`.)

`\__metrix_A_mark:` The (yet) unnamed symbol ⌃.

```
\cs_new:Npn \__metrix_A_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) -- (0.225,0.225) -- (0.45,0);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_A_mark:`.)

`\__metrix_o_mark:` The (yet) unnamed symbol ○.

```
\cs_new:Npn \__metrix_o_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) circle [radius=0.2];
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_o_mark:`.)

`\__metrix_oo_mark:` The aeolic symbol ○○.

```
\cs_new:Npn \__metrix_oo_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) circle [radius=0.2];
```

```
    \draw ($(0.4,0)+(1\pgflinewidth,0)+(\usemetrixvar{gap},0)$) circle [radius=0.2];
   \end{tikzpicture}
  }
```

(*End definition for* `\__metrix_oo_mark:`.)

`\__metrix_u__mark:` The indifferent symbol ≚.

```
\cs_new:Npn \__metrix_u__mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) arc [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0,-0.2)+(-0.5\pgflinewidth,-\pgflinewidth)-(0,\usemetrixvar{gap})$) --
    ($(0.4,-0.2)+(0.5\pgflinewidth,-\pgflinewidth)
    +(0,-\usemetrixvar{gap})$);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_u__mark:`.)

`\__metrix__u_mark:` The indifferent symbol ≛.

```
\cs_new:Npn \__metrix__u_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) arc [start~angle=0, end~angle=180, radius=-0.2];
   \draw ($(0,0)+(-0.5\pgflinewidth,0.5\pgflinewidth)+(0,\usemetrixvar{gap})$) --
    ($(0.4,0)+(0.5\pgflinewidth,0.5\pgflinewidth)
    +(0,\usemetrixvar{gap})$);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix__u_mark:`.)

`\__metrix_n_mark:` An alternative indifferent symbol ≏.

```
\cs_new:Npn \__metrix_n_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) arc [start~angle=0, end~angle=180, radius=0.225];
   \fill (-0.225,0.75*\usemetrixvar{symbollinewidth})
    circle [radius=0.7\pgflinewidth];
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_n_mark:`.)

`\__metrix_p_mark:` The primary stress symbol ⟋.

```
\cs_new:Npn \__metrix_p_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (-0.2,-0.2) -- (0.2,0.2);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_p_mark:`.)

__metrix_s_mark: The secondary stress symbol ˌ.

```
\cs_new:Npn \__metrix_s_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (-0.2,0.2) -- (0.2,-0.2);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_s_mark:`.)

__metrix_l_mark: The simple break symbol | (above syllables).

```
\cs_new:Npn \__metrix_l_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) -- (0,0.5);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_l_mark:`.)

__metrix_ll_mark: The verse break symbol ‖ (above syllables).

```
\cs_new:Npn \__metrix_ll_mark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) -- (0,0.5);
   \draw ($(\pgflinewidth,0)+(1.5*\usemetrixvar{gap},0)$) -- ++(0,0.5);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_ll_mark:`.)

__metrix_l_bigmark: The simple break symbol | (stand alone version).

```
\cs_new:Npn \__metrix_l_bigmark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) -- (0,0.8);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_l_bigmark:`.)

__metrix_ll_bigmark: The verse break symbol ‖ (stand alone version).

```
\cs_new:Npn \__metrix_ll_bigmark:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0) -- (0,0.8);
   \draw ($(\pgflinewidth,0)+(1.5*\usemetrixvar{gap},0)$) -- ++(0,0.8);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_ll_bigmark:`.)

__metrix_l_break The simple break symbol | (between syllables with symbols).

```
\cs_new:Npn \__metrix_l_break:
 {
  \begin{tikzpicture}[every~metrix~symbol,baseline=0.05em]
   \draw (0,\usemetrixvar{symbolshift}+0.325em)
    -- (0,-0.05em) -- (0,0.8em) -- (0,\usemetrixvar{symbolshift});
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_l_break`.)

__metrix_ll_break The verse break symbol || (between syllables with symbols).

```
\cs_new:Npn \__metrix_ll_break:
 {
  \begin{tikzpicture}[every~metrix~symbol,baseline=0.05em]
   \draw (0,\usemetrixvar{symbolshift}+0.325em)
    -- (0,-0.05em) -- (0,0.8em) -- (0,\usemetrixvar{symbolshift});
   \draw
    [
     shift={($(\pgflinewidth,0)+(1.5*\usemetrixvar{gap},0)$)},
    ]
    (0,\usemetrixvar{symbolshift}+0.325em) -- (0,-0.05em) -- (0,0.8em)
    -- (0,\usemetrixvar{symbolshift});
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_ll_break`.)

__metrix_short_break: The shorter break symbol.

```
\cs_new:Npn \__metrix_short_break:
 {
  \begin{tikzpicture}[every~metrix~symbol]
   \draw (0,0.3) -- (0,-0.3);
  \end{tikzpicture}
 }
```

(*End definition for* `\__metrix_short_break:`.)

__metrix_foot_break: The shorter break symbol for foot breakt is the same as the regular short break.

```
\cs_set_eq:NN \__metrix_foot_break: \__metrix_short_break:
```

(*End definition for* `\__metrix_foot_break:`.)

9.9 User level macros

\setmetrixvar This macro saves the value to an internal variable.

```
\NewDocumentCommand{ \setmetrixvar }{ m m }
 {
  \tl_if_exist:cTF { g__metrix_variable_#1_tl } {
   \tl_set:cn { g__metrix_variable_#1_tl } { #2 }
  }
  {
   \__metrix_error_msg:n { Unknown~variable~'#1'. }
  }
 }
```

(*End definition for* `\setmetrixvar`. *This function is documented on page 14.*)

`\usemetrixvar` With this command one can access the value of an internal variable.[4]

```
\DeclareExpandableDocumentCommand{ \usemetrixvar }{ m }
 {
  \tl_if_exist:cTF { g__metrix_variable_#1_tl } {
   \tl_use:c { g__metrix_variable_#1_tl }
  }
  {
   \__metrix_error_msg:n { Unknown~variable~'#1'. }
  }
 }
```

(*End definition for* `\usemetrixvar`. *This function is documented on page 14.*)

`\metrics` This user macro calls `\@_metrics` to typset syllables with symbols.

```
\NewDocumentCommand { \metrics } { O{} m m }
 {
  \__metrix_evaluate_higlights:n { #1 }
  \__metrix_metrics:nn { #2 } { #3 }
 }
```

(*End definition for* `\metrics`. *This function is documented on page 4.*)

`\metricsymbols` This command typesets stand alone symbols. The starred version prints smaller versions.

```
\NewDocumentCommand { \metricsymbols } { s O{} m }
 {
  \group_begin:
   \IfBooleanF { #1 } { \tikzset{every~metrix~symbol/.style={every~metrix~big~symbol}} }
   \__metrix_evaluate_higlights:n { #2 }
   \__metrix_metricsymbols:n { #3 }
  \group_end:
 }
```

(*End definition for* `\metricsymbols`. *This function is documented on page 3.*)

`\lng` This macro prints the longa accent above its argument.

```
\NewDocumentCommand { \lng } { D(){0,0} O{0pt} m O{0pt} }
 {
  \begin{tikzpicture}[
     baseline = (l__metrix_syllable_node_\int_use:N \l__metrix_process_int.base),
     every~metrix~accent
  ]
   \node [every~metrix~syllable~node]
      (l__metrix_syllable_node_\int_use:N \l__metrix_process_int)
      { #3 };
   \begin{pgfinterruptboundingbox}
    \draw [shorten~< = -#2, shorten~> = -#4]
     ($(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.north)
     - (\usemetrixvar{lngminlength}/2,0)
     +(\usemetrixvar{accentxshift},\usemetrixvar{lngshift})
     + (\tl_use:N \l__metrix_internal_itcorrection_tl,0)
     + (#1)$)
```

[4] Marco Daniel showed me this hint at http://tex.stackexchange.com/q/124600/4918.

```
    --
    ($(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.north)
    + (\usemetrixvar{lngminlength}/2,0)
    +(\usemetrixvar{accentxshift},\usemetrixvar{lngshift})
    + (\tl_use:N \l__metrix_internal_itcorrection_tl,0)
    + (#1)$)
   ;
   \draw [shorten~< = -#2, shorten~> = -#4]
    ($(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.north~west)
    +(\usemetrixvar{lngshortening}+\usemetrixvar{accentxshift},\usemetrixvar{lngshift})
    + (\tl_use:N \l__metrix_internal_itcorrection_tl,0)
    + (#1)$)
    --
    ($(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.north~east)
    +(-\usemetrixvar{lngshortening}+\usemetrixvar{accentxshift},\usemetrixvar{lngshift})
    + (\tl_use:N \l__metrix_internal_itcorrection_tl,0)
    + (#1)$)
   ;
  \end{pgfinterruptboundingbox}
 \end{tikzpicture}%
}
```

(*End definition for* `\lng`. *This function is documented on page* 7.)

`\brv` This macro prints the brevis accent above its argument.

```
\NewDocumentCommand { \brv } { D(){0,0} m }
 {
  \begin{tikzpicture}[
     baseline = (l__metrix_syllable_node_\int_use:N \l__metrix_process_int.base),
     every~metrix~accent
  ]
   \node [every~metrix~syllable~node]
      (l__metrix_syllable_node_\int_use:N \l__metrix_process_int)
      { #2 };
   \begin{pgfinterruptboundingbox}
    \draw ($(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.north)
     + (-0.15,0)
     + (\usemetrixvar{accentxshift},\usemetrixvar{brvshift})
     + (\tl_use:N \l__metrix_internal_itcorrection_tl,0)
     + (#1)$)
     arc [start~angle=0, end~angle=180, radius=-0.15];
   \end{pgfinterruptboundingbox}
  \end{tikzpicture}
 }
```

(*End definition for* `\brv`. *This function is documented on page* 7.)

`\acct` This macro prints the dot accent below its argument.

```
\NewDocumentCommand { \acct } { D(){0,0} m }
 {
  \begin{tikzpicture}[
     baseline = (l__metrix_syllable_node_\int_use:N \l__metrix_process_int.base),
     every~metrix~accent
   ]
   \node [every~metrix~syllable~node]
```

```
      (l__metrix_syllable_node_\int_use:N \l__metrix_process_int)
      { #2 };
   \begin{pgfinterruptboundingbox}
    \fill ($(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.south)
     + (0,\usemetrixvar{dotshift})
     + (#1)$)
     circle [radius=1.25\pgflinewidth];
   \end{pgfinterruptboundingbox}
  \end{tikzpicture}
 }
```

(*End definition for* `\acct`. *This function is documented on page 7.*)

`\bow` This macro prints the bow below it's argument.

```
\NewDocumentCommand { \bow } { O{0pt} m O{0pt} }
 {
  \begin{tikzpicture}[
    baseline = (l__metrix_syllable_node_\int_use:N \l__metrix_process_int.base),
    every~metrix~bow
  ]
   \node [every~metrix~syllable~node]
      (l__metrix_syllable_node_\int_use:N \l__metrix_process_int)
      { #2 };
   \draw [shorten~< = #1, shorten~> = #3]
    ($(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.base~west)+
    (\usemetrixvar{bowshortening},\usemetrixvar{bowshift})$)
    to [out=-45, in=225,looseness=\usemetrixvar{bowlooseness}]
    ($(l__metrix_syllable_node_\int_use:N \l__metrix_process_int.base~east)+
    (-\usemetrixvar{bowshortening},\usemetrixvar{bowshift})$);
  \end{tikzpicture}
 }
```

(*End definition for* `\bow`. *This function is documented on page 8.*)

9.10 Ti*k*Z styles

The **mětrix** package uses several Ti*k*Z sytles to draw the macros.

```
\ExplSyntaxOff
\tikzset {
 every metrix symbol/.style={
  line width=\usemetrixvar{symbollinewidth},
  color=\usemetrixvar{symbolcolor},
  x=\usemetrixvar{baseunit},y=\usemetrixvar{baseunit},
 },
 every metrix big symbol/.style={
  line width=\usemetrixvar{bigsymbollinewidth},
  color=\usemetrixvar{symbolcolor},
  x=\usemetrixvar{bigbaseunit},y=\usemetrixvar{bigbaseunit},
 },
 every metrix symbol node/.style={
  inner sep=0pt, anchor=center,
 },
 every metrix ictus node/.style={
   overlay,
```

```
},
every metrix break node/.style={
 inner sep=0pt, anchor=base,
},
every metrix syllable node/.style={
 inner sep=0pt, anchor=base,
},
every metrix bow/.style={
 line width=\usemetrixvar{bowlinewidth},
 color=\usemetrixvar{bowcolor},
 x=\usemetrixvar{baseunit},y=\usemetrixvar{baseunit},
},
every metrix accent/.style={
 line width=\usemetrixvar{accentlinewidth},
 color=\usemetrixvar{accentcolor},
 x=\usemetrixvar{baseunit},y=\usemetrixvar{baseunit},
},
bold highlight/.style={
 every metrix symbol/.append style={line width=2\pgflinewidth},
 every metrix syllable node/.append style={font=\bfseries},
 every superscript node/.append style={font/.expand once=\tikz@textfont\bfseries},
},
colored highlight/.style={
 every metrix symbol/.append style={draw=#1},
 every metrix syllable node/.append style={text=#1},
 every superscript node/.append style={text=#1},
},
colored highlight/.default={
 \usemetrixvar{highlightcolor}
},
dashed highlight/.style={
 every metrix symbol/.append style={dash pattern=on 1pt off 0.4pt},
},
filled highlight/.style={
 every metrix symbol node/.append style={inner sep=2pt,fill=#1},
},
filled highlight/.default={
 \usemetrixvar{fillcolor},
},
every superscript picture/.style={
 baseline=-3ex,
},
every superscript node/.style={
 inner sep=0pt,
 font=\scriptsize,
},
every superscript label/.style={
 inner xsep=0pt,
 inner ysep=-3ex,
 label distance=0.5pt,
},
add superscript/.style={
 label={[every superscript label]right:{%
  \tikz[every superscript picture]\node at (0,0) [every superscript node] {#1};%
```

```
  }},
 },
 superscript/.style={
  every metrix symbol node/.append style={
   add superscript=#1,
  },
  every metrix break node/.append style={
   add superscript=#1,
  },
 },
 superscript/.value required,
 add arrow/.style={
  every metrix symbol node/.append style={
   label=90:\usemetrixvar{arrow},
  },
 },
 add text/.style={
  every metrix symbol node/.append style={
   label={[every metrix added text]#1},
  },
 },
 every metrix added text/.style = {
  font = \scriptsize\itshape,
 },
 add text/.value required,
}
\ExplSyntaxOn
```

9.11 Environments

symbolline Environment to display stand alone symbols.

```
\NewDocumentEnvironment{symbolline} { }
 {
  \par\addvspace{\baselineskip}
  \centering
 }
 {
  \par\vspace{\baselineskip}
  \noindent\ignorespacesafterend
 }
```

(*End definition for* `symbolline`. *This function is documented on page* 9.)

__metrix_print_vers_ref:n The internal macro to print the verse reference inside of {metricvers}

```
\cs_new:Npn \__metrix_print_vers_ref:n #1
 {
     \hspace*{\fill}\nolinebreak[1] \quad \hspace*{\fill} \mbox{\footnotesize #1}
 }
```

(*End definition for* `\__metrix_print_vers_ref:n`.)

metricverses Environment to display a verse with metric symbols and a source. And a macro to print
\verseref a right aligned reference.

```
\NewDocumentCommand { \verseref } { m }
```

```
 {
  \__metrix_error_msg:n {
   \string\verseref\space can~only~be~used~in~{metricverses}~env.
  }
 }
\NewDocumentEnvironment { metricverses } { }
 {
  \RenewDocumentCommand { \verseref } { m }
   {
    \__metrix_print_vers_ref:n { ##1 }
   }
  \par
  \addvspace { 0.7\baselineskip }
  \fp_compare:nT { \usemetrixvar { symbolshift } < 0.0 }
   {
    \vspace { \usemetrixvar { symbolshift } }
   }
  \addtolength { \baselineskip } { 0.6\baselineskip }
 }
 {
  \par
  \addtolength { \baselineskip } { -0.6\baselineskip }
  \vspace { \baselineskip }
  \noindent \ignorespacesafterend
 }
```

(*End definition for* `metricverses` *and* `\verseref`. *These functions are documented on page 9.*)

⟨/package⟩

10 Change History

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.