ifis-macros

Version 2.0, 25.02.2025

Macros for plain TeX by Udo Wermuth

There are three main macros in the files `ifisinteger.tex`, `ifisdimension.tex`, and `ifisglue.tex`. The macro `\ifisint` tests if a given input string represents a number for TeX. The macro `\ifisdim` does this for dimensions and `\ifisglue` for glue and muglue.

All macros generate errors but hide them from the terminal as they work in `\batchmode`. There is one configuration parameter: `\IIcurrentmode`. The default is `\errorstopmode`. Change this if you call the macros in a different interaction mode so that they can return to this mode.

For `\ifisglue` an application is included. The macros `\ifnatwd`, `\ifstretch`, and `\ifshrink` test the corresponding component of two glue or muglue specifications.

0. Installation

To use the macro `\ifisint` load via `\input ifisinteger.tex` the file that contains the code. For `\ifisdim` use `\input ifisdimension.tex` and for `\ifisglue` `\input ifisglue.tex`.

To use `\ifnatwd`, `\ifstretch`, or `\ifshrink` enter `\input gluecomp.tex`.

1. File `ifisinteger.tex`

The main macro is called `\ifisint` and must be used like an `\if`-conditional except that its argument is delimited by `\Boolend`: `\ifisint <argument>\Boolend <true branch>\else <false branch>\fi`.

The implemented algorithm has four steps:

1. 1) Remove signs with or without braces; add sentinel `W`.
   2) Test that the input isn't now `"W`, etc.; otherwise return false.
2. Create a canonical form with a leading zero.
3. 1) Assign the input to a `\count` register inside an `\hbox`.
   2) Test that the box width is the width of the sentinel.
   3) Otherwise return false.
4. 1) Return true if the number isn't TeX's maximum.
   2) Otherwise test if the canonical form is TeX's maximum. If yes, return true.
   3) Otherwise return false.

For more details see my article in TUGboat **45**:1 (2024), 106–109.

2. File `ifisdimension.tex`

The main macro is called `\ifisdim` and must be used like an `\if`-conditional except that its argument is delimited by `\Boolend`: `\ifisdim <argument>\Boolend <true branch>\else <false branch>\fi`.

The implemented algorithm has four steps:

1. 1) Remove signs with or without braces; add sentinel `mm`.
   2) Exclude trivial non-numerics as done in `\ifisint`.
   3) Otherwise return false.
2. 1) Get the integer part.
   2) Get fraction and the unit.
   3) Get the width of the unit.
3. 1) Assign the input to a `\dimen` register inside an `\hbox`.
   2) Test that the box width is the width of the sentinel.
   3) Otherwise return false.
4. 1) Return true if the dimension isn't TeX's `\maxdimen`.
   2) Otherwise test if the coerced sum of the integer part and the fraction expressed in the unit `sp` is `\maxdimen`.

3) If no, return false.
4) Otherwise return true.

For more details see my article in TUGboat **45**:1 (2024), 109–112.

3. File `ifisglue.tex`

The main macro is called `\ifisglue` and must be used like an `\if`-conditional except that its argument is delimited by `\Boolend`: `\ifisglue <argument>\Boolend <true branch>\else <false branch>\fi`.

In contrast to integers and dimensions glue and muglue are specific to TEX. The macro `\ifisglue` accepts only alphabetic constants without backslashes and shouldn't be used with the double-hat notation involving the hexadecimal digit 'f'. I assume no user enters glue specifications with such constructions.

The implemented algorithm has four steps:

1. Convert the characters of the keywords to lowercase letters; use uppercase for 'f'.
2. Split at keywords `minus` and `plus` and store the components in `\II@nw` (natural width), `\II@st` (stretchability), `\II@sh` (shrinkability).
3. 1) Replace in `\II@nw` `mu` by `pt`.
   2) Replace in `\II@st` `mu`, `Fil`, `Fill`, and `Filll` by `pt`.
   3) Do 2) for `\II@sh`.
   4) Count replacements and absences; if the count equals 3, set `\ifIImugluetrue` as the input must be muglue.
4. 1) Check if `\II@nw` is a valid dimension.
   2) If okay check `\II@st`.
   3) If okay check `\II@sh`.
   4) If okay perform the box-width test using `\skip` for glue and `\muskip` for muglue.
   5) If okay return true; otherwise return false.

For more details see my article in TUGboat **46**:1 (2025), 145–147.

4. File `gluecomp.tex`

This is an application of `\ifisglue`. It provides three macros `\ifnatwd`, `\ifstretch`, and `\ifshrink` to test individual components of two glue or muglue specifications. They must be used with so-called TEX split glue macros (Tsgm).

Usage: `\if<macro> <Tsgm1><relation><Tsgm2>\Boolend <true branch>\else <false branch>\fi`, where `\if<macro>` is one of `\ifnatwd`, `\ifstretch`, or `\ifshrink` and `<relation>` is one of `<`, `=`, or `>`.

First create a Tsgm via `\makeTsgm <glue or muglue>!<control sequence>`. It defines the `<control sequence>` and stores the data of `<glue or muglue>`.

Second test one component of two Tsgms. For example, after `\makeTsgm 3pt plus 1fil!\one` and `\makeTsgm 1pt plus 1.1fil!\two` the test `\ifnatwd\one>\two\Boolend` executes the `<true branch>`; with the relations `<` and `=` the `<false branch>` is chosen. Similar, `\ifstretch\one<\two\Boolend` and `\ifshrink\one=\two\Boolend` are true and other relations make the comparisons false.

For more details see my article in TUGboat **46**:1 (2025), 147–150.