Summary of `qsymbols`

Kristoffer H. Rose*

Version 1.12 ⟨1997/01/07⟩

Abstract

`qsymbols` is a LaTeX [1] package defining systematic mnemonic abbreviations, starting with a single open quote ‘ for symbols, and in double quotes " ... " for arrows, for characters from the `amssymb` and `stmaryrd` fonts. Optionally a very large class of arrows can be typeset using the XY-pic package.

Contents

1 Introduction

`qsymbols` sets up a number of mnemonic and compact abbreviations for math symbols from LaTeX and the packages `amssymb` and `stmaryrd`, which it loads. The abbreviations all start with the backquote character ‘ except for arrows that are of the form `"->"`. Some are a single characters, some a more complicated pattern, but always the idea is to use abbreviations that hint at the *visual* appearance of the symbol. Finally it is possible for the user to add more abbreviations of the simpler categories.

*E-mail: `krisdiku.dk`, W3 URL: `http://www.diku.dk/users/kris/`.

You can retrieve `qsymbols` as well as the `amssymb` and `stmaryrd` packages by anonymous ftp from all CTAN archives in directory `/tex-archive/macros/latex/contrib/supported/`[1] (each package has its own subdirectory).

2 Simple symbols

2.1 Greek letters

All the standard greek letters used in math are available as ‘ followed by a letter: either lowercase:

x	a	b	c	d	e	f	g	h	i	j	k	l	m	n	p	q	r	s	t	u	v	w	x	y	z
‘x	α	β	χ	δ	ϵ	ϕ	γ	η	ι	ψ	κ	λ	μ	ν	π	θ	ρ	σ	τ	ϱ	φ	ω	ξ	υ	ζ

or uppercase:

X	D	F	G	J	L	P	Q	S	W	X	Y
‘X	Δ	Φ	Γ	Ψ	Λ	Π	Θ	Σ	Ω	Ξ	Υ

2.2 Common symbols

Simple symbols are available using ‘ followed by a symbolic representation of the symbol. The most common have single character representations:

x	+	*	:	;	/	U	C	_	T	o	.	=	~	E	A	!	^	V
‘x	$\pm$	$\times$	$\in$	$\notin$	$\setminus$	$\cup$	$\complement$	$\bot$	$\top$	$\circ$	$\cdot$	$\equiv$	$\simeq$	$\exists$	$\forall$	$\neg$	$\wedge$	$\vee$

2.3 Circled and Boxed Symbols

These are represented using round and square brackets, respectively:

x	+	-	‘*	/	|	‘/	‘.	*	‘o	‘^	‘V	<	>	?	!	:-	R	C	a
‘(x)	$\oplus$	$\ominus$	$\otimes$	$\oslash$	$\odot$	$\obslash$	$\circledcirc$	$\circledast$	$\circledcirc$	$\circ$	$\circledwedge$	$\circledvee$	$\circledless$	$\circledgtr$	$\circ$	$\bullet$	$\circledsmile$	$\circledR$	$\circledC$	$\circled{a}$
‘[x]	$\boxplus$	$\boxminus$	$\boxtimes$	$\boxslash$	$\boxbar$	$\boxbslash$	$\boxdot$	$\boxast$	$\boxcirc$	$\boxwedge$	$\boxvee$	$\boxless$	$\boxgtr$	$\square$	$\blacksquare$	$\boxed{\frown}$	$\boxed{R}$	$\boxed{C}$	$\boxed{a}$
‘<x>	$\lozenge$													$\diamond$		$\circledequal$				
‘{x}	$\varnothing$															$\circledsim$				

As it can be seen, ‘undefined’ codes of the forms ‘`(a)` and ‘`[a]` result in the contents being circled and boxed, respectively.

2.4 Bold symbols

The $\mathcal{AMS}$-LaTeX `\boldsymbol` command is available by using the special abbreviation ‘`@`x for the bold version $\boldsymbol{x}$ of x as well as ‘`@‘`x where x is on one of the forms described in this section, i.e., ‘`@‘a` gives $\boldsymbol{\alpha}$.

2.5 Adding new symbols

Symbols of all the above forms can be added using the form

`\newqsymbol {‘`*code*`} {`*expansion*`}`

which makes ‘*code* behave as *expansion* in math mode. *code* should be either a single character or some characters enclosed in `()`, `[]`, `<>`, or `{}`.

[1] The ‘home’ of `qsymbols` is `ftp.diku.dk` in directory `/diku/users/kris/TeX/qsymbols/`.

3 Orderings

Two to four consecutive ‘s indicate an ordering relation:

$\epsilon, \ni$	‘‘ϵ	‘‘/ϵ	‘‘ϵ=	‘‘/ϵ=	‘‘‘ϵ	‘‘‘‘ϵ	‘‘$\ni$	‘‘/$\ni$	‘‘$\ni$=	‘‘/$\ni$=	‘‘‘$\ni$	‘‘‘‘$\ni$
<, >	$<$	$\not<$	$\leq$	$\not\leq$	$\wedge$	$\bigwedge$	$>$	$\not>$	$\geq$	$\not\geq$	$\vee$	$\bigvee$
(,)	$\subset$	$\not\subset$	$\subseteq$	$\not\subseteq$	$\cap$	$\bigcap$	$\supset$	$\not\supset$	$\supseteq$	$\not\supseteq$	$\cup$	$\bigcup$
[,]	$\sqsubset$	$\not\sqsubset$	$\sqsubseteq$	$\not\sqsubseteq$	$\sqcap$	$\bigsqcap$	$\sqsupset$	$\not\sqsupset$	$\sqsupseteq$	$\not\sqsupseteq$	$\sqcup$	$\bigsqcup$
\{, \}	$\prec$	$\not\prec$	$\preceq$	$\not\preceq$	$\curlywedge$	$\bigcurlywedge$	$\succ$	$\not\succ$	$\succeq$	$\not\succeq$	$\curlyvee$	$\bigcurlyvee$
\<, \>	$\lhd$	$\not\lhd$	$\trianglelefteq$	$\not\trianglelefteq$	$\triangle$	$\bigtriangleup$	$\rhd$	$\not\rhd$	$\trianglerighteq$	$\not\trianglerighteq$	$\bigtriangledown$	$\bigtriangledown$
~, \~	$\sim$	$\not\sim$	$\simeq$	$\not\simeq$	$\wr$		$\backsim$	$\not\backsim$	$\eqsim$	$\not\eqsim$	$\int$	$\int$
(-, -)	$\in$	$\notin$			$\barwedge$		$\owns$					
(+, +)	$\Subset$	$\not\Subset$	$\Subset$	$\not\Subset$	$\doublebarwedge$	$\bigdoublebarwedge$	$\Supset$	$\not\Supset$	$\Supset$	$\not\Supset$	$\uplus$	$\biguplus$

Some abbreviations are provided for convenience:

x	U	V	S	P
‘‘x	$\bigcup$	$\bigvee$	$\sum$	$\prod$

There is no simple way to add more orderings.

4 Arrows

Double quotes " ... " make it possible to typeset arrows. On some systems " is reserved for other uses, in that case you can use ‘" ... " instead.

4.1 Canned arrows

The available arrows are shown in figure 1 where ! means that the arrow is available in a long version by adding a ! just after the stem character (one of -=), and ? means that it stretches to accomodate labels (when no !s are given, see below).

4.2 Labelling arrows

Inserting {^*s*} or {_*s*}, where *s* is a legal super- or subscript, respectively, will typeset these as limits, and even grows it in those cases where the arrow is marked with a “@” in the table.

x	-{_1}>	3>{^{‘a‘.‘b}}	<|-|>{_{\mbox{push}}}
"x"	$\xrightarrow[1]{}$	$\overset{\alpha\cdot\beta}{\Longrightarrow}$	$\xleftrightarrow[\text{push}]{}$

4.3 Adding new arrows

You can add more ‘canned’ arrows of this kind with commands

\newqsymbol {"*arrow*"} {*expansion*}

which makes "*arrow*" behave as *expansion* in math mode.

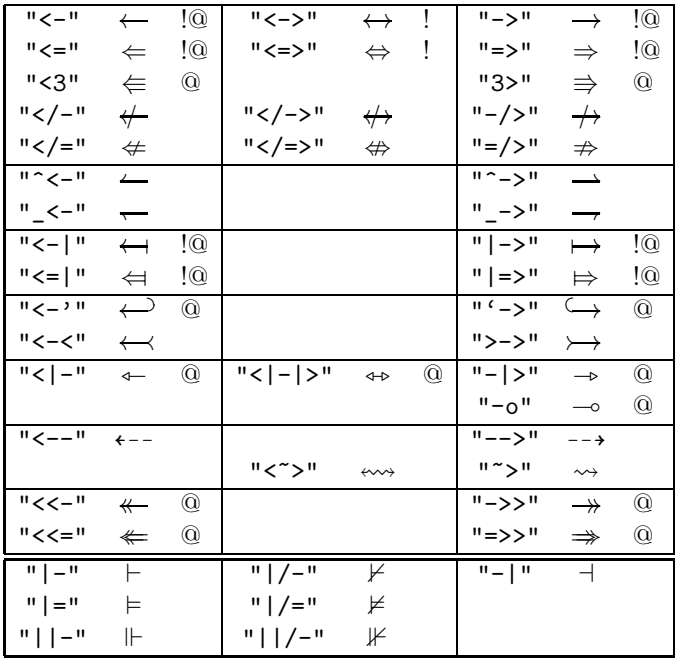


Figure 1: Standard 'canned' arrow symbols.

Similarly, you can add stretchable arrows using commands of the form

`\newqsymbol {"`*arrow*`@"} {`*filler*`}`

which makes "*arrow*" stretch under long labels as *filler* dictates: this should behave as the plain TEX command `\rightarrowfill` or use the macro

`\genericarrowfill{`*tail*`}{`*leader*`}{`*head*`}`

`qsymbols` includes, for example, the declaration

```
\newqsymbol{"3>@"}{\genericarrowfill\equiv\equiv\Rrightarrow}
```

4.4 Using XY-pic for arrows

If the option `[xy]` is used in the `\usepackage` command, or if XY-pic [3][2] is already loaded, then the XY-pic arrow feature (with the 'cmtip' extension) is used to allow a much more general class of arrows.

First, all blank entries in figure 1 are filled; if the option `[purexy]` is used instead of `[xy]` then all the entries of the table are replaces with XY-pic generated arrows (this gives a somewhat more homogenous look and avoids loading of `ams` and `stmary` arrows).

Second, general arrows can be constructed according to the following rules:

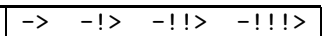
- Basic arrows are composed by combining the variants `23^_`, the tips `<>|xo''`, and the connectors `-=.:~`.

[2] XY-pic version 3 is needed for this to work.

- The character `/` 'negates' the arrow (once or twice) similar to the way `\not` does for relations:

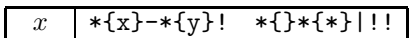
x	`'-/>` `=//!>`
"x"	

- Each `!` character makes the arrow a bit longer.

x	`->` `-!>` `-!!>` `-!!!>`
"x"	

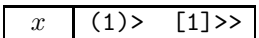
Note: Some arrows are automatically made a bit longer, e.g., the `<~>` arrow shown above.

- The form `*`*object* inserts the XY-pic *object* which will be used for the tail, shaft, or tip as indicated by the position. Here are some examples:

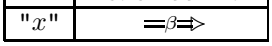
x	`*{x}-*{y}!` `*{}*{*}|!!`
"x"	

as in the examples it is recommended to specify all three of tail, shaft, and head, when using this, in particular an empty tail when the shaft is specified with `*` because otherwise it is taken as the tail.

- The forms `(`x`)` and `[`x`]` insert a break with x in a circle and box, respectively:

x	`(1)>` `[1]>>`
"x"	

- The special code `{`ℓ`}` adds the ℓ material to the end of the XY-pic arrow: All XY-pic ⟨labels⟩ can be used as described in [3, §16], for example,

x	`={|{'b}}!!|>`
"x"	

Use this with care!

- Similarly the special code `@{`ℓ`}` adds the `@`ℓ material (note the omission of the braces) to the beginning of the XY-pic arrow: all XY-pic arrow ⟨form⟩s can be used ¡form¿ to the beginning of the arrow specification; this can be used to as described in [3, §16], for example,

x	`|-@{/^/}!!!>|` `->!!!!@{(dr,ul)}`
"x"	

Use this with care!

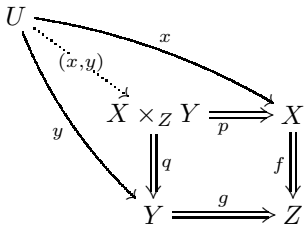
4.5 Using `qsymbols` arrows in XY-pic diagrams

Finally it is possible to some extent to use `qsymbols` arrows in XY-pic matrices (as described in the XY-pic User's Guide [2]) and graphs (as described in the XY-pic Reference Manual [3, §19]). First notice that you should always use the `‘" ... "` form. Second, the entire `‘" ... "` construction behaves as an arrow made with `\ar` for matrices and `:` for graphs, that is, you must add a 'target address' for the arrow after it. Further information of this can be found

Here is the canonical pull-back example diagram from category theory typeset using qsymbols:

```
\xymatrix{
 U ‘"->"@/_/[ddr]_y ‘".>"[dr]|-{(x,y)} ‘"->"@/^/[drr]^x \\
  & X \times_Z Y ‘"=>"[d]^q ‘"=>"[r]_p & X ‘"=>"[d]_f   \\
  & Y ‘"=>"[r]^g                       & Z             }
```

typesets



As you can see, XY-pic is loaded by `qsymbols` and as a convenience XY-pic options may be passed to `qsymbols`.

References

[1] Leslie Lamport. *LaTeX—A Document Preparation System*. Addison-Wesley, 2nd edition, 1994.

[2] Kristoffer H. Rose. XY-pic user's guide. Mathematics Report 94–148, MPCE, Macquarie University, NSW 2109, Australia, June 1994. For version 2.10+. Latest version available with URL `ftp://ftp.diku.dk/diku/users/kris/TeX/xy/xyguide.ps`.

[3] Kristoffer H. Rose and Ross Moore. XY-pic reference manual. Mathematics Report 94–155, MPCE, Macquarie University, NSW 2109, Australia, June 1994. For version 2.10+. Latest version available by anonymous ftp in `ftp.diku.dk: /diku/users/kris/TeX/xyrefer.ps.Z`.