XY-pic

4. Features

4.1. Path and Arrow
4.2. Graph
4.3. Matrix
4.4. Polygon
4.5. Circle, Ellipse and Arc
4.6. Two-cell
4.7. Lattice and web
4.8. Knots and Links
4.9. Other options

5. Basic syntax of XY-pic

Some informations through internet

XY-pic home page
XY-pic user's guide
XY-pic tutorial with an archive of examples
An introductory tutorial on drawing knots in XY-pic

xypic.zip Package of XY-pic(Ver.3.7)
should be expanded where `.\texmf` exists.
This contains the followng first two references

References

Kristoffer H. Rose, *XY-pic user's guide*, 1999.

Kristoffer H. Rose and Ross R. Moore, *XY-pic reference manual*, 1999.

Michel Goosens, Sebastian Rahtz, Franklin Mittebach, *The LaTeX Graphic Companion*, 1997.

Tutorial (in Japanese)

Xy-pic \xymatrix

Top

```
\input xy \xyoption{all}
```

```
\xymatrix{
 entry & entry & ··· \\
 entry & entry & ··· \\
 ···}
```

entry:*modifier* *object* *decoration*

1.1. Examples

```
\xymatrix{
 A &*+[F]{\sum_{k=m}^n a_k} \\
   & {\bullet} & D \ar[ul] }
```

$$\begin{tikzcd}
A & \boxed{\sum_{k=m}^{n} a_k} & \\
 & \bullet & D \arrow[ul]
\end{tikzcd}$$

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

$$\begin{tikzcd}
U \arrow[ddr, bend right, "y"'] \arrow[drr, bend left, "x"] \arrow[dr, dotted, "(x{,}y)" description] & & \\
 & X \times_Z Y \arrow[d, "q"] \arrow[r, "p"'] & X \arrow[d, "f"'] \\
 & Y \arrow[r, "g"] & Z
\end{tikzcd}$$

`*` draw object
with `+` room and `[F]` frame

`\ar` draw an arrow from current
position to `[`*hop*`]` with options:

hop: left right up down
`[ddr]`: 2 down + 1 right

`@/_/`: curve right `@/^/`: curve left

`[`*hop*`]_{`*it*`}`: put *it*em on the right (down) side
`[`*hop*`]^{`*it*`}`: put *it*em on the left (up) side
`[`*hop*`]|{`*it*`}`: put *it*em in the middle with a hole

`@{`*style*`}`: define a *style* of the arrow (default: `@{->}`)
style: *tail shaft head*

`@{=>}`	⟹	`@{.>}`	┈┈>	`@{:>}`	⇢	`@{~>}`	⇝
`@{-->}`	- - >	`@{-}`	——	`@{<->}`	⟷	`@3{|->}`	⇛
`@{{*}-{*}}`	●——●	`@{}`		`@_{x->}`	×——>	`@{^{(}->}`	↪
`@{>->}`	↣	`@2{~}`	≈≈	`@{.>>}`	┈┈↠	`@{{}{x}{}}`	××××××

```
\xymatrix{
 {\bullet} \ar@{--}[d]\ar@=[dr]\ar@-[r]
           & {\bullet} \ar@.[d] \\
 {\bullet} & {\bullet} \ar[l] }
```

$$\begin{tikzcd}
\bullet \arrow[d, dashed, no head] \arrow[dr, equal] \arrow[r, no head] & \bullet \arrow[d, dotted, no head] \\
\bullet & \bullet \arrow[l]
\end{tikzcd}$$

Top

1.2. Labels

```
\xymatrix@1{X\ar[r]^a_b&Y&Z\ar[l]^A_B}
```

$$\begin{tikzcd} X \arrow[r, "a", "b"'] & Y & Z \arrow[l, "A", "B"'] \end{tikzcd}$$

@1: better result for one-line diagrams

^{*text*}, _{*text*}: put *text* on the left or right side.

```
\xymatrix@1{A\cup B\cup C\cup D\ar[r]^-{+}&X}
```

$$\begin{tikzcd} A \cup B \cup C \cup D \arrow[r, "+"] & X \end{tikzcd}$$

($\begin{tikzcd} A \cup B \cup C \cup D \arrow[r, "+" near start] & X \end{tikzcd}$ without the -)

```
\xymatrix{
 A \ar[r]^f \ar[dr]_{f;g}
  & B \ar[d]^g \ar[dr]^{g;h}\\
  & C \ar[r]_h & D}
```

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "f;g"'] & B \arrow[d, "g"] \arrow[dr, "g;h"] & \\ & C \arrow[r, "h"'] & D \end{tikzcd}$$

1.3. Breaks

```
\xymatrix@1{A\ar[r]|f&B}
```

$$\begin{tikzcd} A \arrow[r, "f" description] & B \end{tikzcd}$$

```
\xymatrix@1{A\ar[r]|\hole&B}
```

$$\begin{tikzcd} A \arrow[r, "\ " description] & B \end{tikzcd}$$

```
\xymatrix{
 \ar@{}[dr]|{\color{red}\circlearrowright}
 A \ar[d] \ar[r] & B \ar[d] \\
 C \ar[r] & {\color{green}D} }
```

$$\begin{tikzcd} A \arrow[d] \arrow[r] \arrow[dr, phantom, "\circlearrowright"] & B \arrow[d] \\ C \arrow[r] & D \end{tikzcd}$$

1.4. Curving

@/*curving*/

```
\xymatrix@1{A \ar@/^/[r] & B}
```

$$\begin{tikzcd} A \arrow[r, bend left] & B \end{tikzcd}$$

```
\xymatrix@1{A \ar@/_/[r] & B}
```

$$\begin{tikzcd} A \arrow[r, bend right] & B \end{tikzcd}$$

```
\xymatrix@1{A \ar@/_1pc/[r] & B}
```

$$\begin{tikzcd} A \arrow[r, bend right=60] & B \end{tikzcd}$$

@(*in*, *out*): *in*, *out*: u ur r dr d dl l ul

```
\xymatrix{ x \ar@(ul,dl)[]|{id}
  \ar@/^/[rr]|f
   && f(x) \ar@/^/[ll]|{f^{-1}} }
```

$$\begin{tikzcd} x \arrow[loop left, "id" description] \arrow[rr, bend left, "f" description] & & f(x) \arrow[ll, bend left, "f^{-1}" description] \end{tikzcd}$$

1.5. Explicit label positioning

< tail of the arrow

> head of the arrow

<< or <<< etc. intermediate position and - is the center of arrow.

```
\xymatrix@1{A\ar[r]^<{+}&B}
```

$$\begin{tikzcd} A \arrow[r, "+" pos=0] & B \end{tikzcd}$$

```
\xymatrix@1{A\ar[r]^>{+}&B}
```

$$\begin{tikzcd} A \arrow[r, "+" pos=1] & B \end{tikzcd}$$

```
\xymatrix@1{A\ar[r]^>>{+}&B}
```

$$\begin{tikzcd} A \arrow[r, "+" pos=0.85] & B \end{tikzcd}$$

```
\xymatrix@1{A\ar[r]^>>>{+}&B}
```

$$\begin{tikzcd} A \arrow[r, "+" pos=0.7] & B \end{tikzcd}$$

(*factor*): *factor* based on the objects
<(*factor*): *factor* based on <
>(*factor*): *factor* based on >   (- equals <>(.5))

```
\xymatrix@1{A\ar[r]^(.2){+}&B}
```

$$\begin{tikzcd} A \arrow[r, "+" near start] & B \end{tikzcd}$$

```
\xymatrix@1{A\ar[r]^<(.2){+}&B}
```

$$\begin{tikzcd} A \arrow[r, "+" pos=0.3] & B \end{tikzcd}$$

!{*s*;*t*}: the point where the line from *s* to *t* crosses it

```
\xymatrix{
 1 \ar[rr]^-1000000x
   \ar[dr]_(.2)2000x
      |!{[d];[rr]}\hole
              && 1000000 \\
 1000 \ar[r]_2x
   \ar[urr]_>>>>{x^2}
         & 2000 }
```

$$\begin{tikzcd} 1 \arrow[rr, "1000000x"] \arrow[dr, "2000x"' near start] & & 1000000 \\ 1000 \arrow[r, "2x"'] \arrow[urr, "x^2"' near end, crossing over] & 2000 & \end{tikzcd}$$

1.6. Labeling with any object

Top

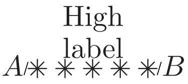
{*math*}: *math* material as object
*{*math*}: similar but original style and blank margin are ignored
**modifiers*{*text*}: change the shape and side according to *modifiers*

+	grow
+<*dimen*>	grow by *dimen*
+=	grow to enclosing square
-	shrink
-<*dimen*>	shrink by *dimen*
-=	shrink to contained square
!	do not center
[o]	round
[l] [r] [u] [d]	adjust left, right, . . .
[F] [F=]	frame, double frame
[F,] [F--]	dotted frame, dashed frame
[F-,] [F-;<*3pt*>]	shaded frame, frame with rounded edges
@*variant*{*tip*}	*tip* (or *shaft*) object (cf. next section)
\txt<*width*>{*text*}	similar as \hbox{} but \\ can be used in *text*
\composit{*obj****obj*}	combine objects
\frm{}	last object

```
\xymatrix@1{ A \ar[r]^*+[Fo]{x} & B }
```

$$\begin{tikzcd} A \arrow[r, "\textcircled{x}"] & B \end{tikzcd}$$

```
\xymatrix@1{ A \ar
  @{/*\composite{{+}*{\times}}/} [rr]
  ^*+\txt{High\\label} && B}
```



1.7. More arrow styles

Top

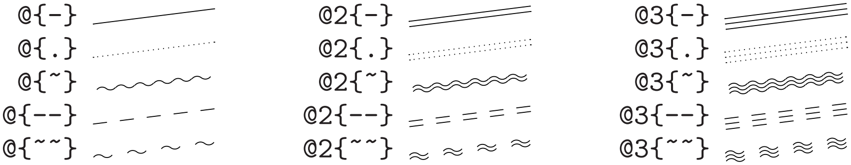
@*variant*{*tail shaft head*}: *tail* and *head* can be omitted

@*variant*{*head*}

variant{*tip*}   *variant*{*shaft*}

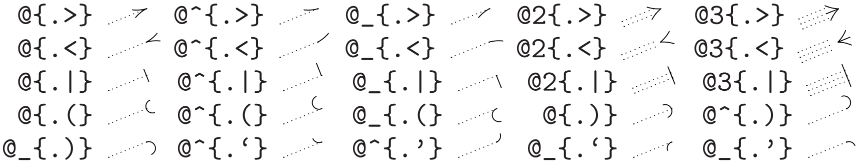
variant: ^ left   _ right   2 double   3 triple

Plain *shaft*s (in kernel library)



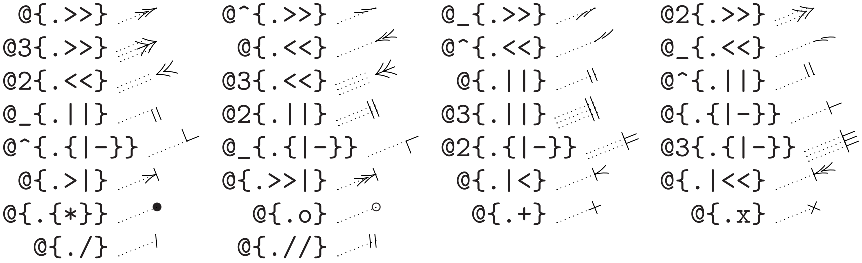
`@{-}`	`@2{-}`	`@3{-}`
`@{.}`	`@2{.}`	`@3{.}`
`@{~}`	`@2{~}`	`@3{~}`
`@{--}`	`@2{--}`	`@3{--}`
`@{~~}`	`@2{~~}`	`@3{~~}`

Plain *tip*s which are *head*s or *tail*s (in kernel library)



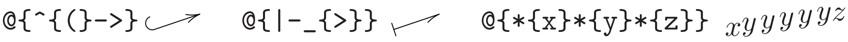
`@{.>}`	`@^{.>}`	`@_{.>}`	`@2{.>}`	`@3{.>}`
`@{.<}`	`@^{.<}`	`@_{.<}`	`@2{.<}`	`@3{.<}`
`@{.|}`	`@^{.|}`	`@_{.|}`	`@2{.|}`	`@3{.|}`
`@{.(}`	`@^{.(}`	`@_{.(}`	`@{.)}`	`@^{.)}`
`@_{.)}`	`@^{.‘}`	`@^{.’}`	`@_{.‘}`	`@_{.’}`

Constructed *tip*s which are *head*s or *tail*s (in kernel library)



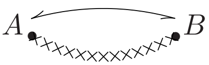
`@{.>>}`	`@^{.>>}`	`@_{.>>}`	`@2{.>>}`
`@3{.>>}`	`@{.<<}`	`@^{.<<}`	`@_{.<<}`
`@2{.<<}`	`@3{.<<}`	`@{.||}`	`@^{.||}`
`@_{.||}`	`@2{.||}`	`@3{.||}`	`@{.{|-}}`
`@^{.{|-}}`	`@_{.{|-}}`	`@2{.{|-}}`	`@3{.{|-}}`
`@{.>|}`	`@{.>>|}`	`@{.|<}`	`@{.|<<}`
`@{.{*}}`	`@{.o}`	`@{.+}`	`@{.x}`
`@{./}`	`@{.//}`		

Some arrows



`@{^{(}->}`	`@{|-_{>}}`	`@{*{x}*{y}*{z}}`

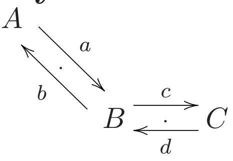
```
\xymatrix{A \ar @/^/ @{^{<}-_{>}} [rr]
 \ar @/_1pc/ @{{*}{x}{*}} [rr] && B}
```



1.8. Sliding arrows sideways

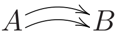
@<*len*>: slide the arrow left sideways by *len*

```
\xymatrix{ A \ar@<1ex>[dr]^a_. \\
  & B \ar@<1ex>[ul]^b \ar@<1ex>[r]^c
    & C \ar@<1ex>[l]^d_. }
```



$$\begin{tikzcd} A \arrow[dr, shift left, "a", "." '] & \\ & B \arrow[ul, shift left, "b"] \arrow[r, shift left, "c"] & C \arrow[l, shift left, "d", "." '] \end{tikzcd}$$

```
\xymatrix@1{
  A \ar@/^/[r] \ar@/^/@<-1ex>[r] & B }
```



$$\begin{tikzcd} A \arrow[r, bend left] \arrow[r, bend left, shift right] & B \end{tikzcd}$$

Top

1.9. More addresses of targets

[r,c]: relative entry. [1,2] and [-2,0] means [drr] and [uu]

"r,c": absolute entry. The top left is "1,1"

b;t: change the base and target to b and t, respectively.

```
\xymatrix{ A \\
           B & C \ar@<1ex>[ul]
                 \ar@<1ex>[ul];[] }
```

$$\begin{tikzcd} A \arrow[dr, shift left] & \\ B & C \arrow[ul, shift left] \end{tikzcd}$$

1.10. Changing the position of a target

!*vector*: moves the center of the target by *vector*

+*vector* or -*vector*: change the target with size 0 and the central position is shifted by *vector*:

<x,y>: *vector* giving by size   0: zero *vector*

U UR R DR D DL L UL : *vector* to the corners of the target

/d *dimen*/: *vector* going *dimen* in the direction where d is:

u ur r dr d dl l ul

va(θ): absolute angle

a(θ) : relative angle

(x,y): relative vector

^ _ : a(90) and a(-90), respectively

empty: empty means the current (=last)

1.11. Arrows passing under

't: visited target t except the last

```
\xymatrix{ {\circ}
 \ar@{|-->} '[dr] ^a
            '[rr]+D*\bullet ^b
             [drrr] ^c
      &{\circ}&{\circ}&{\circ}\
 {\circ}&{\circ}&{\circ}&{\circ} }
```

$$\begin{tikzcd} \circ \arrow[dr, dash, dashed, "a"] & \circ & \circ \arrow[dr, dashed, "c"] & \circ \\ \circ & \circ \arrow[ur, dash, dashed, "b"] & \circ & \circ \end{tikzcd}$$

```
\xymatrix@!0{       % see 1.16 for @!0
 & \lambda\omega
   \ar@{-}[rr]\ar@{-}'[d][dd]
  && \lambda C \ar@{-}[dd]\\
 \lambda2 \ar@{-}[ur]
   \ar@{-}[rr]\ar@{-}[dd]
  && \lambda P2\ar@{-}[ur]\ar@{-}[dd]\\
 & \lambda\underline\omega \ar@{-}'[r][rr]
   && \lambda P\underline\omega\\
 \lambda{\to} \ar@{-}[rr]\ar@{-}[ur]
  && \lambda P \ar@{-}[ur]}
```

$$\begin{tikzcd} & \lambda\omega \arrow[rr, dash] \arrow[dd, dash] & & \lambda C \arrow[dd, dash] \\ \lambda2 \arrow[ur, dash] \arrow[rr, dash] \arrow[dd, dash] & & \lambda P2 \arrow[ur, dash] \arrow[dd, dash] & \\ & \lambda\underline{\omega} \arrow[rr, dash] & & \lambda P\underline{\omega} \\ \lambda\to \arrow[rr, dash] \arrow[ur, dash] & & \lambda P \arrow[ur, dash] & \end{tikzcd}$$

1.12. More bending arrows

Top

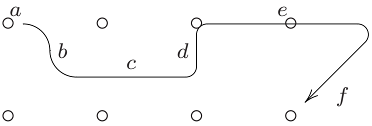
‘*d t*: start *d*irection and turn to *t*arget

d may be u ur r,...

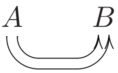
curve ends after a quarter turn

default radius is 10pt. It can be changed by /*radius* just after ‘

```
\xymatrix{
 {\circ} \ar ‘r[d]      ^a
             ‘[rr]      ^b
             ‘/4pt[rr] ^c
             ‘[rrr]     ^d
             ‘_dl[drrr]^e
                [drrr]^f
         & {\circ} & {\circ} & {\circ} \\
 {\circ} & {\circ} & {\circ} & {\circ} }
```



```
\xymatrix@1{
 A \ar@<-2pt> ‘d[r] ‘[r] [r]
   \ar@<+2pt> ‘d[r] ‘[r] [r] & B }
```



1.13. Defining new arrow types

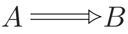
> < | o x + / ( ) [ ] ␣ : *tip* characters

- . ~ : = : *shaft* characters

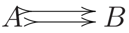
\newdir*variant*{*directional*}{*composite*}: *directional* should be a sequence of *tip*/*shaft* characters

!*vector*: shift object by *vector*

```
\newdir{|>}{%
 !/4.5pt/@{|}*:(1,-.2)@^{>}*:(1,+.2)@_{>}}
\xymatrix{ A \ar @{=|>} [r] & B }
```



```
\newdir{ >}{{}*!/-5pt/@{>}}
\xymatrix{ A \ar @{>->} @< 2pt> [r]
             \ar@{ >->} @<-2pt> [r] & B }
```

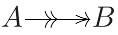


\SelectTips{cm}{*point*}: style for Computer Modern fonts

\SelectTips{eu}{*point*}: style for Euler fonts

point: default is 10

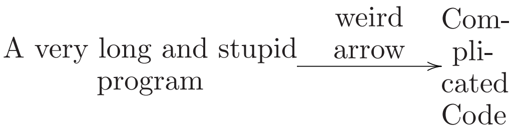
```
\xymatrix@1{ A \ar[r]
   |-{\SelectTips{cm}{}\object@{>>}}
   |>{\SelectTips{eu}{}\object@{>}} & B }
```



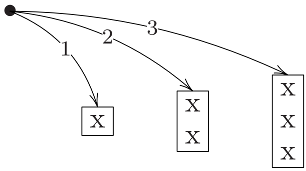
1.14. Manual entry formatting

Top

```
\xymatrix{
 *\txt{A very long and stupid\\program}
  \ar[rr]^-{\txt{weird\\arrow}}
 &&*\txt<2pc>{Com\-pli\-cated\\Code}}
```



```
\xymatrix{
 *=0{\bullet}
 \ar@/^/[dr]!U|1
 \ar@/^/[drr]!U|2
 \ar@/^/[drrr]!U|3 \
   &*+[F]\txt{x}
   &*+[F]\txt{x\\x}
   &*+[F]\txt{x\\x\\x} }
```

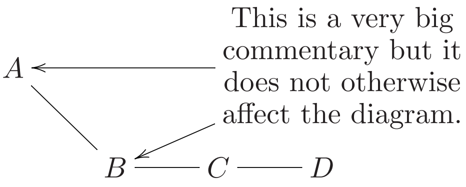


1.15. Extra entries outside the matrix

`\save` *t*: *t* is any kind of typesetting

`\restore`: restore the above *t* as an extra entry

```
\xymatrix{
 A \ar@{-}[dr]
 &{}\save[]+<3cm,0cm>*\txt<8pc>{%
   This is a very big commentary
   but it does not otherwise affect
   the diagram.}
   \ar[l] \ar[d] \restore \\
 & B \ar@{-}[r] & C \ar@{-}[r] & D }
```


This is a very big commentary but it does not otherwise affect the diagram.
A
B
C
D

1.16. Spacing and rotation

`@=`*dimen*: set spacing

`@R=`*dimen*: set row spacing

`@C=`*dimen*: set column spacing

`@M=`*dimen*: set entry default margin

`@W=`*dimen*: set entry default width

`@H=`*dimen*: set entry default height

`@L=`*dimen*: set label margin

They should be put between `\xymatrix` and the following {. `=` can be replaced by `+ += - -=`. Then “set” is replaced by “increase” “increase at most” “decrease” “decrease at most”

`\xymatrixrowsep={`*dimen*`}`: change the default (=2 pt)

`\xymatrixcolsep={`*dimen*`}`: same as above for column spacing

```
\xymatrix@1@=0pt@M=0pt{A&B\\C&D}
```

$$\begin{array}{cc} A & B \\ C & D \end{array}$$

`@!` : force all spacing equal

`@!0`: and ignore entry sizes

`@!R`: force equal row spacing

`@!C`: force equal column spacing

`@`*d* : rotate towards *d*

```
\xymatrix@dr@C=1pc{
 a \ar[r]\ar@{->>}[d]
   & a' \ar@{.>>}[d] \\
 b \ar@{.>>}[r]
   & b' }
```

$$\begin{tikzcd}
 & a \arrow[dl, two heads] \arrow[dr] & \\
b \arrow[dr, dotted, two heads] & & a' \arrow[dl, dotted, two heads] \\
 & b' &
\end{tikzcd}$$

1.17. Entry style

Top

`\objectstyle`: entry style (default: mathmode in text style)

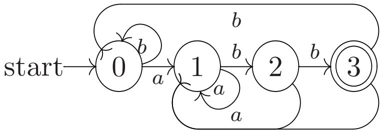
`\labelstyle` : label syle

`\entrymodifiers={`*modifier*`}`: apply *modifier* to all entries

```
$\left(
 \def\objectstyle{\scriptstyle}
 \def\labelstyle{\scriptstyle}
  \vcenter{\xymatrix @-1.2pc @ur {
   A \ar[r]^{a}  & B \ar[d]^{b} \\
   A'\ar[u]^{a'} & B'\ar[l]^{b'} }} \right)$
```

$$\left(\begin{tikzcd}
 & B \arrow[dr, "b"] & \\
A \arrow[ur, "a"] & & B' \arrow[dl, "b'"] \\
 & A' \arrow[ul, "a'"] &
\end{tikzcd}\right)$$

```
\entrymodifiers={++[o][F-]}
\SelectTips{cm}{}
\xymatrix @-1pc {
 *\txt{start} \ar[r]
 & 0 \ar@(r,u)[]^b \ar[r]_a
 & 1 \ar[r]^b \ar@(r,d)[]_a
 & 2 \ar[r]^b
   \ar `dr_l[l] `_ur[l] _a [l]
 &*++[o][F=]{3}
   \ar `ur^l[lll]`^dr[lll]^b [lll]
   \ar `dr_l[ll] `_ur[ll]    [ll] }
```

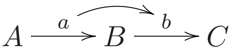


1.18. Naming for later use as targets

Top

t="*name*": give "*name*" to *t*arget to be referred it later

```
\xymatrix{
 A \ar[r] ^a="a"
  & B \ar[r] ^b="b" & C
    \ar @/^/ "a";"b" }
```

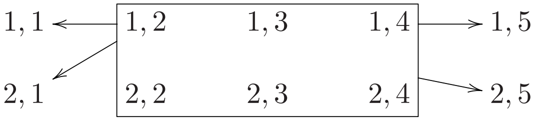


1.19. Grouping objects

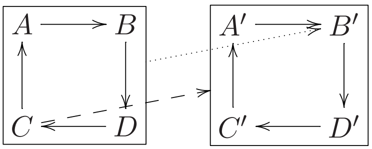
t.*s*: marge *t* with simple *s*

{*t*}: make *t* simple

```
\xymatrix @R=1pc {
 1,1 & 1,2 & 1,3 & 1,4 & 1,5 \\
 2,1 & 2,2 & 2,3 & 2,4 & 2,5
 \save "1,2"."2,4"*[F]\frm{}
  \ar"1,1" \ar"2,1" \ar"1,5" \ar"2,5"
 \restore }
```

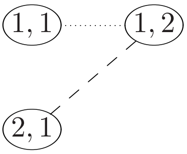


```
\def\g#1{\save
 [].[dr]!C="g#1"*[F]\frm{}\restore}%
\xymatrix{
 \g1 A\ar[r]&B\ar[d]&\g2 A'\ar[r]&B'\ar[d]\\
     C\ar[u]&D\ar[l]&    C'\ar[u]&D'\ar[l]
 \ar @{.>}  "g1" ;"1,4"
 \ar @{-->} "2,1";"g2" }
```

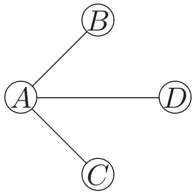


1.20. More examples

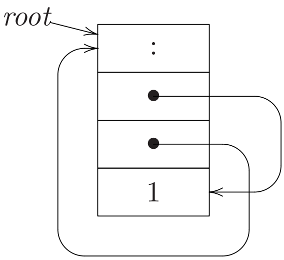
```
\everyentry={{\the\Row,\the\Col}}
\xymatrix @*[F]@*[o] {
 {} \POS[];[r]**\dir{..} & \\
 {} \POS[];[ur]**\dir{--} }
```



```
\entrymodifiers={=<1pc>[o][F-]}
\xymatrix @ur {
 A \save[];[r] **\dir-,
        [];[dr]**\dir-,
        [];[d] **\dir-\restore
   & B \\
 C & D }
```



```
\xymatrix @W=3pc @H=1pc @R=0pc @*[F-] {%
 : \save+<-4pc,1pc>*\hbox{\it root}
     \ar[]
   \restore \\
 {\bullet}
   \save*{}
    \ar`r[dd]+/r4pc/`[dd][dd]
   \restore \\
 {\bullet}
   \save*{}
    \ar`r[d]+/r3pc/`[d]+/d2pc/
       `[uu]+/l3pc/`[uu][uu]
   \restore \\
   1 }
```



Xy-pic Kernel

\xy *pos* *decor* \endxy

\begin{xy} *pos* *decor* \end{xy}

Top

2.1. Positions

A position may represent a rectangular region with a reference point in the rectangular.

c: current position

p: previous position

<*X-dimen*,*Y-dimen*>: absolute vector

(*x-factor*,*y-factor*): relative vector (coordinate)

$(x,y) = \langle X_o + x \times X_x + y \times Y_x, Y_o + x \times X_y + y \times Y_y \rangle$

$\langle X_o, Y_o \rangle$ *origin* (default: $\langle 0\,\mathrm{pt}, 0\,\mathrm{pt} \rangle$)

$\langle X_x, Y_x \rangle$ *x-base* (default: $\langle 1\,\mathrm{mm}, 0\,\mathrm{mm} \rangle$)

$\langle X_y, Y_y \rangle$ *y-base* (default: $\langle 0\,\mathrm{mm}, 1\,\mathrm{mm} \rangle$)

pos : *cod* — *origin* ← *p*, *x-base* ← *cod* − *origin*
y-base ← $\langle -Y_x, X_x \rangle$, *c* ← *cod*

pos : : *cod* — *y-base* ← *cod* − *origin*, *c* ← *cod*

pos±*cod* — *c* ← *pos*±*cod* (region is kept)

pos ! *cod* — *c* ← *pos*, skew *c* by *cod* (move reference point)

pos`.`*cod*	$c \leftarrow$ *pos* with covering *cod*	(shape changes)
pos`,`*cod*	$c \leftarrow$ *pos*, then $c \leftarrow$ *cod*	
pos`;`*cod*	$c \leftarrow$ *pos*, swap p c, $c \leftarrow$ *cod*	
pos`*`*obj*	$c \leftarrow$ *pos*, drop (type set) *obj* at c	
pos`**`*obj*	$c \leftarrow$ *pos*, connect using *obj*	
pos`?`*place*	$c \leftarrow$ *pos*, $c \leftarrow$ *place*	
pos`@`*stack*	$c \leftarrow$ *pos*, do *stack*ing	
pos`=`*save*	$c \leftarrow$ *pos*, do *sav*ing (Sometimes *save* is `"`*id*`"`)	

One of the following is recognized as *cod*

`{`*pos decor*`}`	c resulting from interpreting the group
`"`*id*`"`	restore what was saved as *id*
p	p
`x y`	axis intersection with the line through p c
`s`digit `s{`num`}`	stack position (one) digit or num below the top

or the following *vector*s

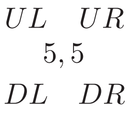
`0` or `<`*X-dimen*`,`*Y-dimen*`>` or `(`*x-factor*`,`*y-factor*`)`	
`<`*dimen*`>`	= `<`*dimen*`,`*dimen*`>`
`L R D U`	offset to left,...
`CL CR CD CU C`	offset to center of left side,...
`LD RD LU RU`	offset to left/down orner,...
`E P`	offset to nearest/proportional edge point to p
`L(`*factor*`)` etc.	the above offset multiplied with *factor*
`a(`num`)`	angle in current base ($\cos$ num^o, $\sin$ num^o) `a(90)`=(0, 1). num is an integer.
`/`*direction dimen*`/`	vector *dimen*(default: 0.5pt) in *direction*: $\langle Z\cos\alpha, Z\cos\alpha\rangle$ if $Z =$*dimen* and $\alpha =$*direction* Sometimes *dimen* or *direction* is omitted

Moreover *place* is one of the followings

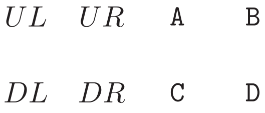
`<`*place*	shave (0) to edge of p, $f \leftarrow 0$
`>`*place*	shave (1) to edge of c, $f \leftarrow 1$ (c `?>` equals c `+E`)
`(`*factor*`)`*place*	$f \leftarrow$ *factor*
`/`*dimen*`/`	pick place and *slide* further by *dimen*
`!{`*pos*`}`	intercept with line setup by *pos* (`/`*dimen*`/`may follow)

Here f is a factor to multiply the offset vector in the last.

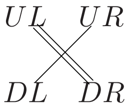
```
\xy
 0*{DL} ,+/r1cm/*{DR}
 , <0cm,1cm>*{UL} ,<1cm,1cm>*{UR}
 , (5,5)*{5,5}
\endxy
```



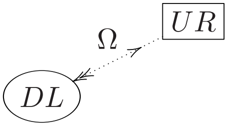
```
\xy 0;<5mm,0mm>:
 0*{DL} ,(2,0)*{DR} ,<0cm,1cm>*{UL} ,(2,2)*{UR}
 , 0+(4,0)="c"*{\mathtt C}
 , "c"+(2,2)="b"*{\mathtt B}
 , "b"-<1cm,0cm>*{\mathtt A}
 , "b"-(0,2)*{\mathtt D}
\endxy
```



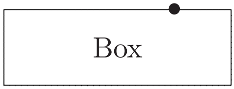
```
\begin{xy}
 0*{DL};<1cm,1cm>*{UR}**@{-},
 <0cm,1cm>*{UL};<1cm,0cm>*{DR}**@{=}
\end{xy}
```



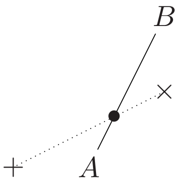
```
\begin{xy}
 0*++[o][F]{DL};<2cm,1cm>*+[F]{UR}**@{.}
 ?<*@{<<} ?>>>>*@{>}
 ?(0.5)*!/_3mm/{\Omega}
\end{xy}
```



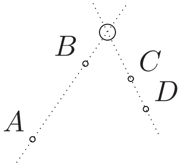
```
\begin{xy}
 *=<3cm,1cm>\txt{Box}*\frm{-}
 !U!R(.5) *\frm{..}*{\bullet}
\end{xy}
```



```
\begin{xy}
 <1cm,0cm>:
 (0,0)*=0+="+" ;     % set size 0
 (2,1)*=0{\times}="*" **@{.} ,
 (1,0)*+{A} ; (2,2)*+{B} **@{-}
 ?!{"+";"*"} *{\bullet}
\end{xy}
```



```
\begin{xy}
 (0,0) ="A" *\cir<1pt>{} *+!DR{A},
 (7,10)="B" *\cir<1pt>{} *+!DR{B},
 (13,8)="C" *\cir<1pt>{} *+!DL{C},
 (15,4)="D" *\cir<1pt>{} *+!DL{D},
 {"A";"B":"C";"D",x}="I" *\cir<3pt>{},
 "I";"A"**{} +/1pc/;-/1pc/ **@{..},
 "I";"D"**{} +/1pc/;-/1pc/ **@{..}
\end{xy}
```



"A";"B": x-axis ← $\overrightarrow{AB}$

"C";"D" c ← "C" and p ← "D"

, x get intersection of $\overrightarrow{pc}$ and x-axis

Top

2.2. Stack

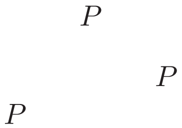
The *stack* is used to store *cod*s.

`@+`*cod*	push *cod*
`@-`*cod*	c ← *cod* then pop
`@=`*cod*	load stack with *cod*
`@@`*cod*	do *cod* for *c* ← *stack*
`@i`	initialize
`@(`	enter new frame
`@)`	leave current frame

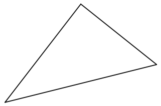
After saving *c* with `···="`*id*`"`

`@:"`*id*`"`	`"`*id*`"` restores current base
`@`*cod*`"`*id*`"`	`"`*id*`"` reinterprets *cod*
`@@"`*id*`"`	`@="`*id*`"` reloads this stack

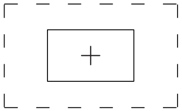
```
\begin{xy}
 @={(0,-10),(10,3),(20,-5)}
 @@{*{P}}
\end{xy}
```



```
\begin{xy}
 @={(0,-10),(10,3),(20,-5)}
 , s0="prev" @@{;"prev";**@{-}="prev"}
\end{xy}
```



```
\begin{xy}
  ={.{+DL(2)}.{+UR(2)}}"dbl"
  ,*+<2pc,1pc>{+}*\frm{-},"dbl"*\frm{--}
\end{xy}
```



2.3. Objects

*obj*ects are used to actual type setting by `*` or `**`at *pos*. They are *modifier obj* or *objectbox*. Here *objectbox* is one of the followings.

`{`*txt*`}`	build default object(reference point is Center)
library object or `@`*dir*	
`{`*TEX box*`}`	usual TEX box such as `\hbox`
`\object`*obj*	wrap up *obj* as a final object box, which can be used outside XY-pic
`\composit{`*obj*`*`*obj*`*···}`	build composit object box
`\xybox{`*pos dec*`}`	package entire XY-picture as *obj*

modifiers are one of the followings

`!`*vector*	*obj* has reference point shifted by *vector*(See §2.1)
`!`	*obj* has its original reference point reinstalled

add-op size	*add-op* is one of `+ - = += -=` (grow, shrink, set, grow to, shrink to) and *size* is a rectangle covering the *vector* in §2.1(*size* may be omitted)
`h i`	hidden or invisible
`[`*shape*`]`	*obj* is given the specified *shape* (=`empty . o` etc)
`[=`*shape*`]`	define *shape* to reestablish current object style
direction	set current direction for this *obj*. They are
`v`*vector*	direction of *vector*
`v{`*pos decor*`}`	direction from *p* to *c* after *pos decor*
direction`:`*vector*	vector relative to *direction*
direction`_` *direction*`^`	(clock/anticlock)wise to *direction*
`l r d u`	left, right, down, up diagonal (called *diag*)
`ld rd lu ru`	left/down,. . . diagonal (also called *diag*)

If *size* is omitted in the above,

`+`	`+<`2× *objectmargin*`>`
`-`	`-<`2× *objectmargin*`>`
`=`	`=<`*objectwidth*`,`*objectheight*`>`
`+=`	`+=<`$\min(\mathtt{L}_c+\mathtt{R}_c,\mathtt{D}_c+\mathtt{U}_c)$`>`
`-=`	`-=<`$\max(\mathtt{L}_c+\mathtt{R}_c,\mathtt{D}_c+\mathtt{U}_c)$`>`

Default values can be changed

`\objectmargin`*add-op*`{`*dimen*`}`

`\objectwidth` `\objectheight` are same as above

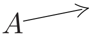
2.4. Decorations

Top

*decor*ators are sequences of commands, which are followings.

`\save` *pos*	save state, then do *pos*
`\restore`	restore state save by matching `\save`
`\POS` *pos*	interpret *pos*
`\aftrePOS{`*decor*`}` *pos*	interpret *pos* and then perform *decor*
`\drop` *obj*	drop *obj*ect as the *pos* `*` operation
`\connect` *obj*	connect *obj*ect as the *pos* `**` operation
`\relax` *obj*	do nothing
TeXcommands	
`\xyverbose` `\xytracing` `\xyquite`	tracing commands
`\xyignore{`*pos decor*`}`	ignore XY-pic code
`\xycompile{`*pos decor*`}`	compile to file *prefix no*.xyc
`\xycompileto{`*name*`}{`*pos decor*`}`	compile to file *name*.xyc

```
\def\ToPOS{\save\afterPOS{%
 \POS**{}?>*@2{>}**@{-}\restore};p,}
\xy *{A} \ToPOS +<10mm,2mm>\endxy
```



2.5. Kernel object library

\dir*variant*{*main*}   *variant*{*main*} is called *dir*ectional
variant is empty or one of ^ _ 2 3, *main* is in the list in §1.7.

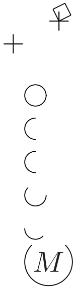
\newdir*variant*{*directional*}{*composite*}   See §1.13.

\cir<*vector*>*cir*}   arc defined by *cir* and *vector*
radius is x-component of *vector* (default ⇒ $\mathtt{R}_c$) and *cir* is

diag1 orient diag2   tangent direction corresponds to *diag*
partial circle from *diag1* to *diag2* in the *orient*ation
(default ⇒ a full circle)
*orient*ation is ^ (anticlockwise) or _ (clockwise).

\txt<*width*>*style*{*text*}

```
\begin{xy}
 *{+}; p+(6,3)*{+} **{} ?(1)
 *@{-} *!/-5pt/^\dir{-}
 *^\dir{-} *!/^-5pt/\dir{-}\end{xy}
\xy*\cir<4pt>{}\endxy
\xy*\cir<4pt>{l^r}\endxy
\xy*\cir<4pt>{l_r}\endxy
\xy*\cir<4pt>{dl^u}\endxy
\xy*\cir<4pt>{dl_u}\endxy
\xy*+M*\cir{dr_ur}\endxy
```



Xy-pic Extensions

3.1. Curve, Circle and Ellipse

Top

\xyoption{curve}

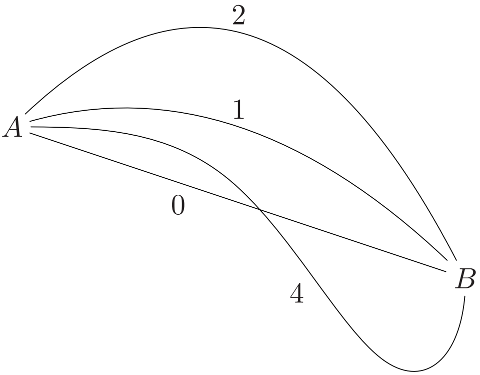
This option provides the typeset spline curved connections using arbitrary directional objects.

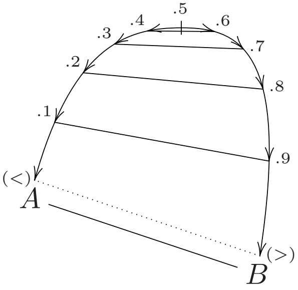
Def

**\crv{*poslist*}   curved connection (*poslist* is a list of *pos*itions)
**\crvs{*direction*}   get *poslist* from the stack
\curve{*poslist*}   as a *decor*ation (\curve equals \connect\crv)

#*poslist*= 0	straight connection
1	single-segment Bézier
2	cubic spline
≥3	cubic B-spline construction

```
\begin{xy}  (0,20)*[o]+A;(60,0)*[o]+{B}="B"
            **\crv{} \POS?(.4)*_+!UR{0},"B"
            **\crv{(30,30)} \POS?*^+!D{1},"B"
            **\crv{(20,40)&(40,40)} \POS?*^+!D{2},"B"
            **\crv{(10,20)&(30,20)&(50,-20)&(60,-10)}
              \POS?*+^!UR{4}\end{xy}
```





```
\def\ssz#1{\hbox{$_{^{#1}}$}}
\begin{xy}
 (0,0)*+{A};(30,-10)*+{B}="B",**\dir{-},
 "B"**\crv{(5,20)&(20,25)&(35,20)}
 ?<(0)*\dir{<}="a" ?>(1)*\dir{>}="h" ?(.1)*\dir{<}="b"
 ?(.9)*\dir{>}="i" ?(.2)*\dir{<}="c" ?(.8)*\dir{>}="j"
 ?(.3)*\dir{<}="d" ?(.7)*\dir{>}="k" ?(.4)*\dir{<}="e"
 ?(.6)*\dir{>}="l" ?(.5)*\dir{|}="f",
 "a"*!RC\txt{\ssz{(<)}}
                ; "h"*!LC\txt{\ssz{\;(>)}},**\dir{.},
 "b"*!RD{\ssz{.1}}; "i"*!L{\ssz{\;.9}},**\dir{-},
 "c"*!RD{\ssz{.2}}; "j"*!L{\ssz{\;.8}},**\dir{-},
 "d"*!RD{\ssz{.3}}; "k"*!L{\ssz{\;.7}},**\dir{-},
 "e"*!RD{\ssz{.4}}; "l"*!LD{\ssz{.6}}, **\dir{-},
 "f"*!D!/^-3pt/{\ssz{.5}}
\end{xy}
```

`\curve`*modifier*`{`*curve-object poslist*`}`

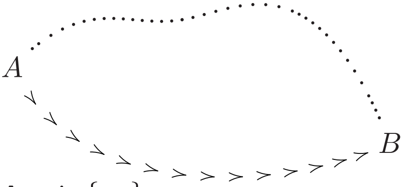
modifier (default `~C`) is `~`*curve-option* with *curve-option*:

`p P l L c C`	only control points, jointed by lines, or curve Lower case version ignores *curve-object*
`pc pC Pc PC`	control points and curves
`lc lC Lc LC`	lines joining control points and curves
`cC`	plot curve twice

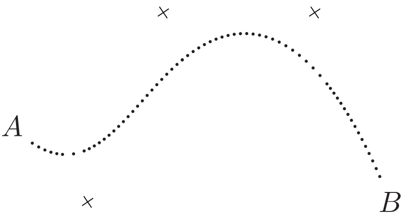
curve-object is empty or a sequence of `~*`*obj* and/or `~**`*obj*.

poslist is empty or a sequence of *pos* and/or `~@` separated by `&`.

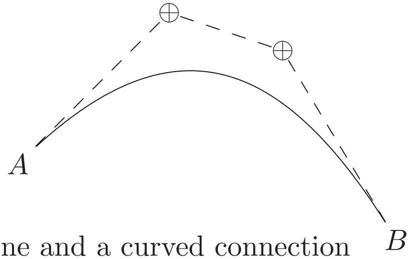
```
\begin{xy}
 (0,0)*+{A}; (50,-10)*+{B}
 **\crv{~*=<4pt>{.} (10,10)&(20,0)&(40,15)}
 **\crv{~*=<8pt>{}~**!/-5pt/\dir{>}(10,-20)&(40,-15)}
\end{xy}
```



```
\begin{xy}
 (0,0)*+{A};(50,-10)*+{B}
 **\crv~pC{~*=<\jot>{.}(10,-10)&(20,15) &(40,15)}
\end{xy}
```

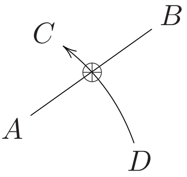


```
\begin{xy}
 (0,0)*+{A};(50,-10)*+{B}
 **\crv~Lc{~**\dir{--}
  ~*{\oplus}
  (20,20)&(35,15)}
\end{xy}
```



Intersection of a straight line and a curved connection

```
\begin{xy}
 *+{A}="A";p+/r5pc/+(0,15)*+{B}="B"
 ,p+<1pc,3pc>*+{C}="C"
 ,"A"+<4pc,-1pc>*+{D}="D",{\ar@/_/"C"}
 ,?!{"A";"B"**@{-}}*++{\oplus}
\end{xy}
```

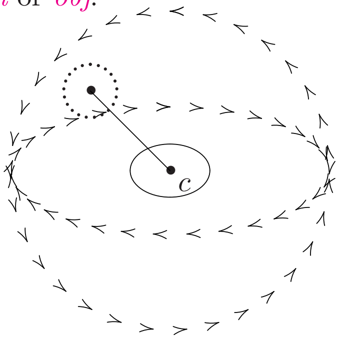


Circles and Ellipse

Top

\xycircle*vector*{*style*} *style* is any *conn* or *obj*.

```
\begin{xy}
 0;/r5pc/:*\dir{*}
  ;p+(.5,-.5)*\dir{*}="c"
 ,**\dir{-},*+!UL{c},"c"
 ,*\xycircle(1,.4){++\dir{<}}
 ,*\xycircle(1,1){++\dir{>}}
 ,*\xycircle<15pt,10pt>{}
 ;*\xycircle<10pt>{{.}}
\endxy
```



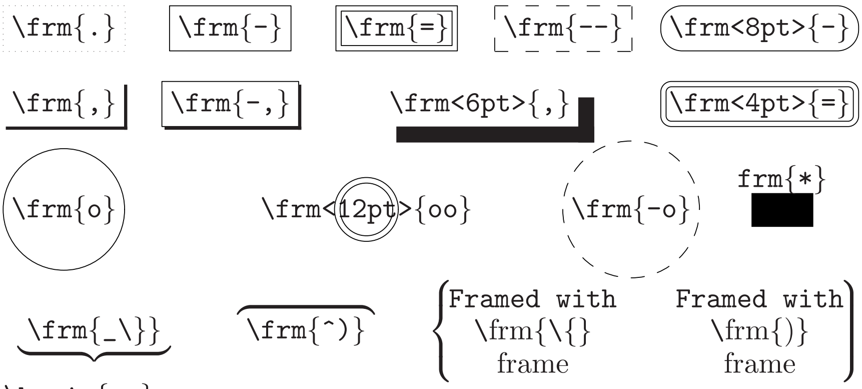
3.2. Frame and Bracket

Top

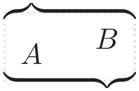
`\xyoption{frame}`	
`\frmopt{`*frame*`}`	put frame at *pos*
`[F `*frame*`: `*opt*`]`	`*\frmopt{`*frame*`}`
`[F `*frame*`]`	`*\frm{`*frame*`}`
`[F]`	`*\frm{-}`

Here *frame* is empty or

`. - =`	allow an optional corner radius by *opt*=<*dimen*>
`-- o-`	`o-` has rounded corner
`, -,`	put a shade with *opt*ional depth of shade by <*dimen*>
`*`	fill a box
`o .o -o oo`	circle with *opt*ional radius by <*dimen*> outer most one of a doubled frame = the single one
`_\} ^\} \{ \}`	put braces bottom, top, left, right
`_) ^) ( )`	put parenthesis bottom, top, left, right



```
\begin{xy}
 (0,0) *++{A} ;
 (10,2) *++{B} **\frm{.}
  **\frm{^\}} ; **\frm{_\}}
\end{xy}
```



3.3. Import Graphics

`\xyimport(`*width*`,`*height*`){`*graphic*`}`
`\xyimport(`*width*`,`*height*`)(`*x-off*`,`*y-off*`){`*graphic*`}`

graphic is a box with a graphic imported by a graphic package.

```
\def\GraA{\resizebox{6cm}{!}{\includegraphcs{foo.eps}}}
\begin{xy}
\xyimport(4,3)(1.2,1,2){\GraA}*\frm{-},
...
```

Xy-pic Features

`\xyoption{all}` Load XY-pic extensions `curve frame cmtip line rotate color matrix arrow` and `graph`.

4.1. Path and Arrow

Top

`\xyoption{arrow}`

PATH

A Path is defined by the command

Def

`\PATH`*path*	interpret *path*
`\afterPATH{`*decor*`}`*path*	interpret *path* and run *decor*

path includes several *segment*s, which are

path-pos dimen labels	*dimen* is optional to *slide*s *obj*

labels are sequence of

`^`*anchor it* `="`*id*`"`	label with *it*em above *anchor* (`="`*id*`"` is optional)
`_`*anchor it* `="`*id*`"`	label with *it*em below *anchor* (`="`*id*`"` is optional)
`|`*anchor it* `="`*id*`"`	label with *it*em at *anchor* (`="`*id*`"` is optional)
	anchor is *place* or `-`*place* (`-` means `<>(0.5)`)

Here *it*em is *digit* or *letter* or `{`*text*`}` or *cs* (default labels) or

`*`*obj*	*obj*ect
`@`*dir*	*dir*ectional

which may be preceded by `[`*shape*`]`

Except for the last one, *segment*s should be either in the forms

`'`*segment*	make straight segment
`` ` ``*diag turnradius segment*	1/4 turning *segment* starting in *diag*
`` ` ``*cirturnradius segment*	explicit turning *segment*

turnradius is `/`*dimen* and use default turn if it is omitted.

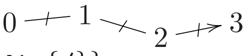
`\turnradius`*add-op*`{`*dimen*`}` to change it (default:10pt).

`` ` ``*segment*	turning *segment* as at the last *diag radius*

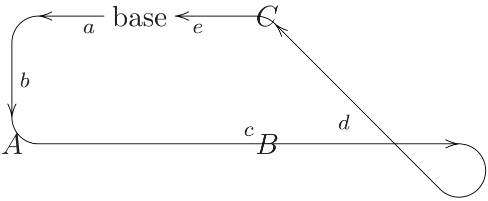
Moreover *path* may also contain

`~`*action*`{`*stuff*`}`	set *action* to *stuff*
`~`*which*`{`*labels*`}`	add *labels* prefix for some segments with *which*
	`<` next segment only
	`>` last segment only
	`=` every segment
`~{`*stuff*`}`	set failure continuation to *stuff*

```
\begin{xy}
 *+0\PATH
  ~={**\dir{-}} ~>{|>*\dir{>}} ~+{|*\dir{/}}
  '(10,1)*+{1}  '(20,-2)*+{2}   (30,0)*+{3}
\end{xy}
```



```
\begin{xy} <4pc,0pc>:(0,0)
*+\txt{base}="base"
\PATH ~={**\dir{-}?>*\dir{>}}
 `l  (-1,-1)*{A} ^a
 `    (1,-1)*{B} ^b
 `_ul (1, 0)*{C} ^c
 `ul^l "base"    ^d
       "base"    ^e
\end{xy}
```

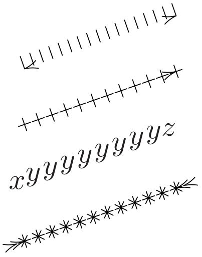


Arrows

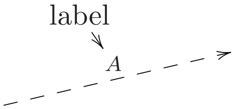
Arrows are paths with *tail*, *stem* and *head* described in §1.7. They are constructed as follows

`\ar`*arrow path*	make *arrow* along *path*
	Here *arrow* is of the form
@*variant*	use *variant* of arrow. *variant* is empty or one of ^ _ 0 1 2 3 with above, below, double or triple
@*variant*{*tip*}	build arrow using *variant* of a standard stem and *tip* for the head. Here *tip* is some of *tipcharacter*s < > ( ) | ' ` + / or *letter* or *space* with a preceding optional *dir*ectional
@*variant*{*tip conn tip*}	make arrow using *variant* with indicated tail, stem and head (in this order). *conn* is a sequence of *connchar*s (with *dir*ectional)
@*connchar*	change stem by *connchar*, which is one of - . ~ =
@!	dash the arrow stem by doubling it
@/*direction dist*/	curve arrow the *dist*ance towards *direction* default *dist*ance is .5pc (may be omitted)
@(*direction*,*direction*)	curve to fit with in-out directions
@`{*control-point-list*}	curve setup with explicit control points See §3.1 for control points.
@[*shape*]	add [*shape*] to object *modifier* for all object
*{*modifier*}	add object *modifier*s for all object
<*dimen*>	slide arrow the *dimen*
|*anchor it*	break each segment at *anchor* with *it*em
^*anchor it*	label each segment at *anchor* with *it*em
_*anchor it*	label each segment at *anchor* with *it*em
@?	reverse meaning of above and below
	anchor is *place* or -*place* (- means <>(0.5))

```
\xy\ar @{<^{|}>} (20,7) \endxy

\xy\ar @{{}{+}>} (20,7) \endxy

\xy\ar @{*{x}*{y}*{z}} (20,7) \endxy

\xy\ar @{>>*\composite{%
     \dir{x}*\dir{+}}<<} (20,7) \endxy
```



```
\begin{xy}
 (0,0) \ar @{-->} (30,7) ^A="a"
 \POS(10,12)*+\txt{label} \ar "a"
\end{xy}
```

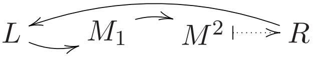


4.2. Graph

Top
Def

See Reference Manual for details.

```
\xygraph{
 []L :@/_/ [r]{M_1} :@/^/ [r]{M^2}
     :@{|.>}[r]R     :@/_1em/"L"
}
```



\xygraph{*graph*} perform *graph* which are *node* or the followings
-*arrow node labels* draw line to *node* with *labels* and move there
:*arrow node labels* draw *arrow* to *node* with *labels* and move there
(*list*) map *list* (= *graph*s separated by ,) to the current *node*

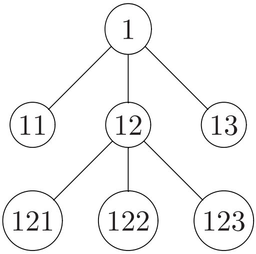
Here *node*s are

[*move*] & \\ ”*id*” ? may be followed by *it*em or ="*id*" or !*escape*

move by some d u l r (may be followed by *place* and *move*)

escape one of M P E ~ (*matrix polygon ellipse* set-up)

```
\xygraph{
[] *+[o]+[F]{1}
  (-[dl]  *+=[o]+[F]{11}
  ,-[d]   *+=[o]+[F]{12}
     (-[dl]  *+=[o]+[F]{121}
     ,-[d]   *+=[o]+[F]{122}
     ,-[dr]  *+=[o]+[F]{123}
     )
  ,-[dr]  *+=[o]+[F]{13}
  )
}
```



4.3. Matrix

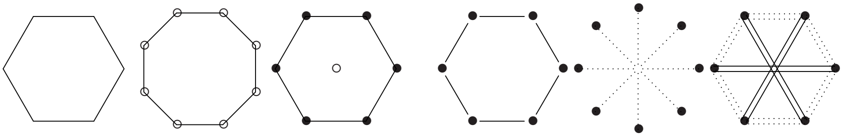
See §1.

4.4. Polygon

See Reference Manual for details.

Def Top

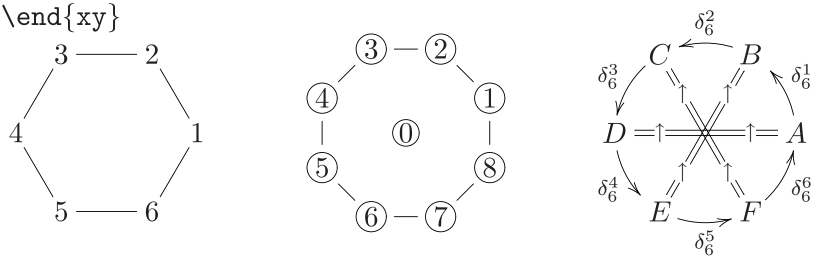
```
\begin{xy} /r8mm/:
 ,   0    ,{\xypolygon6{}}
 ,+/r18mm/,{\xypolygon8{@{o}}}
 ,+/r18mm/,{*@{o}\xypolygon6{@{*}}}
\end{xy}
```



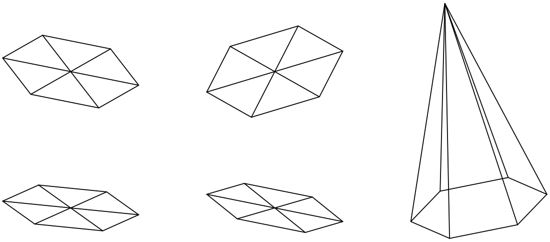
```
\begin{xy} /r8mm/:
 ,   0    ,{\xypolygon6{~*{\dir{*}}}}
 ,+/r18mm/,{\xypolygon8{~<{.}~>{}~={45}{\dir{*}}}}
 ,+/r18mm/,{\xypolygon6{~<{=}~>{:}{\dir{*}}}}
\end{xy}
```

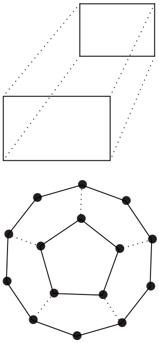
`~:`	any *pos decor* performed first
`~* ^=`	set *obj*ect, *angle* to vertex
`~< ~<< ~<>`	set *dir*ectional, arrow, labels to spokes
`~> ~>< ">>`	set *dir*ectional, arrow, labels to sides

```
\newcounter{node}
\newcommand{\Letter}%
  {{\setcounter{node}%
    {\xypolynode}\Alph{node}}}
\begin{xy} /r12mm/:
  (0,0), {\xypolygon6{~*{\xypolynode}}}
 ,(3.3,0), {*{0}*\cir<5pt>{}
           \xypolygon8{~*{\xybox{%
            *{\xypolynode}*\cir<2mm>{}}}}}
 ,(6.6,0),  {\xypolygon6{~><{@/_.9ex/}
            ~>>{_{\delta^{\xypolynode}_{\xypolynum}}}
            ~<<{@{=}} ~<>{|\uparrow} ~*{\Letter}}}
\end{xy}
```



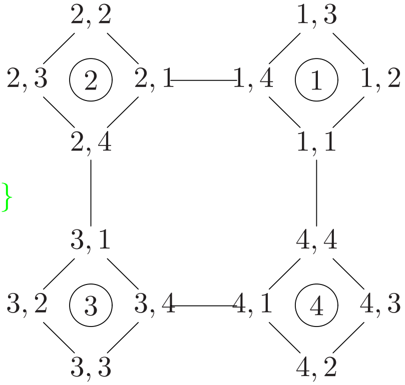
```
\begin{xy} /r9mm/:
  (0,0), {\xypolygon6{~:{(1,-.1):(0,.33)::}~<{-}}}
 ,(0,2), {\xypolygon6{~:{(1,-.2):(0,.5)::}~<{-}}}
 ,(3,0), {\xypolygon6{~:{(1,-.2):(0,-.3)::}~<{-}}}
 ,(3,2), {\xypolygon6{~:{(1,.3):(0,-.6)::}~<{-}}}
 ,(6,0)="0", +(-.5,3)="T","0"
 ,{\xypolygon6{~:{(1,.2):(0,.4)::}~<>{;"T"**@{-}}}}
\end{xy}
```





```
\begin{xy} /r1cm/:
  {\xypolygon4"F"{~:{(0,.6)::}}},+(.8,1.3)
 ,{\xypolygon4"B"{~:{(.7,0):(0,.7)::}}}
 ,"F1";"B1"**@{.}, "F2";"B2"**@{.}
 ,"F3";"B3"**@{.}, "F4";"B4"**@{.},-(1.0,2.7)
 ,{\xypolygon10"O"{~={16}\dir{*}}}
 ,{\xypolygon5"I"{~:{(0.55,0):}
     ~={16}{\dir{*}}}}
 ,"O1";"I1"**@{.}, "O3";"I2"**@{.}
 ,"O5";"I3"**@{.}, "O7";"I4"**@{.}
 ,"O9";"I5"**@{.}
\end{xy}
```

```
\xypolygon4{~:{/r5pc/:}
 ~<>{*\frm<8pt>{o}
  \xypolygon4{~:{/-2pc/:}
  ~*{\xypolyname\xypolynode}}}
[o]=<5pc>{\xypolynode}}
```



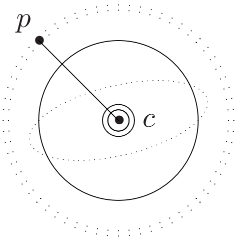
4.5. Circle, Ellipse and Arc

Top

\ellipse(num){*style*} radius = num×dist(p,c), default num=1

\ellipse<*dimen*>{*style*} radius is *dimen*sion

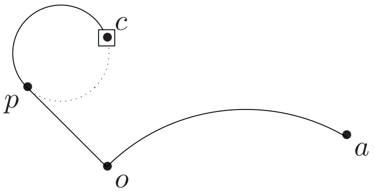
```
\begin{xy}
 0;/r5pc/:*\dir{*}="p",*+!DR{p};
 p+(.5,-.5)*\dir{*}="c"
 ,*+++!L{c}**\dir{-}
 ,{\ellipse<>{:}},{\ellipse(.5){}}
 ,0;(.5,.5)::,"p";"c",{\ellipse(.5){.}}
 ,{\ellipse<5pt>{=}}
\end{xy}
```

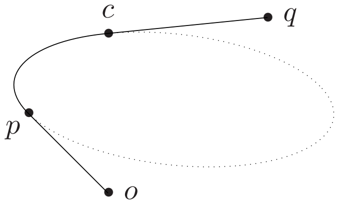


\ellipse_{*style*} clockwise arc from p to c

\ellipse^{*style*} counter-clockwise arc from p to c

```
\begin{xy}
 0;/r5pc/:*=+\dir{*}**+!UR{p};
 p+(.5,-.5)*\dir{*}="o",**+!UL{o}
 ,+(0,.81)*=<6.1pt>\dir{*}*\frm{-}="c"
 ,**+!DL{c},"o",**\dir{-}
 ,"c",{\ellipse_{}},{\ellipse^{.}}
 ,"o"+(1.5,.2)*\dir{*}="a"**+!UL{a}
 ,"o";p+/_1pc/,**{},"a",{\ellipse_{}}
\end{xy}
```





See Reference Manual for more details.

```
\begin{xy}
 0;/r5pc/:*\dir{*}="p",*+!UR{p}
 ;p+(.5,-.5)*\dir{*}="o",*++!L{o}**\dir{-}
 ,p+(.5,.5)*\dir{*}="c",*++!D{c},"c"
 ;p+(1,.1)*\dir{*}="q",*++!L{q}**\dir{-}
 ,"o";"p",**{};"c"
 ,{\ellipse![["o";"p"]],_![["q";"c"]]{}}
 ,{\ellipse![["o";"p"]],![["c";"q"]]{.}}
\end{xy}
```

4.6. Two-cell

See Reference Manual for details.

Top

```
\xymatrix{
 L\rtwocell^u_d & R }
```

$$\begin{tikzcd}
L \arrow[r, bend left, "u"] \arrow[r, bend right, "d"'] \arrow[r, phantom, "\Downarrow"] & R
\end{tikzcd}$$

```
\xymatrix @=15mm{
  L_1 \rlowertwocell<-3>_{a_1}{<-1>}
      \rcompositemap<6>_{a_2}^{a_3}{\omit}
& R_1 \dtwocell<0>_{a_4}{"}\\
  L_2 \uuppertwocell_u^d{\omit}
      \rtwocell<2>{'id}
& R_2}
```

$$\begin{tikzcd}[column sep=15mm, row sep=15mm]
 & {} \arrow[dr, "a_3"] & \\
L_1 \arrow[ur, "a_2"] \arrow[rr, bend right, "a_1"'] \arrow[r, phantom, "\Downarrow"] & & R_1 \arrow[d, leftrightarrow, "a_4"'] \\
L_2 \arrow[u, bend left, "d"] \arrow[r, phantom, "u"] \arrow[rr, bend left] \arrow[rr, phantom, "id"] & & R_2 \arrow[ll, bend left]
\end{tikzcd}$$

4.7. Lattice and web

Top

Lattice is the set given by

$$L = \{a\vec{u} + b\vec{v}\,;\ a \text{ and } b \text{ are integers}\}$$

Here $a_{min} \le a \le a_{max}$, $b_{min} \le b \le b_{max}$ are plotted.

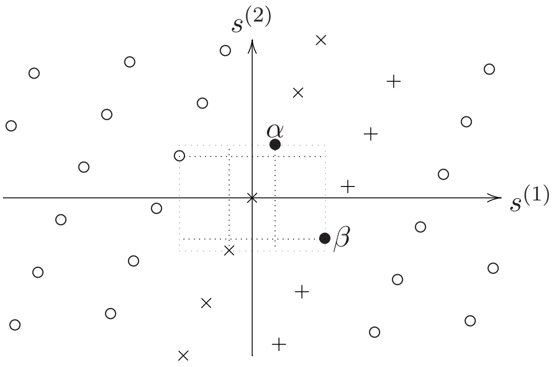
\xylattice#1#2#3#4 with (#1,#2,#3,#4)= $(a_{min}, a_{max}, b_{min}, b_{max})$

\croplattice#1#2#3#4#5#6#7#8 cropped by X-coordinates of #5$\times\vec{u}$ and #6$\times\vec{u}$ and Y-coordinates of #5$\times\vec{v}$ and #8$\times\vec{v}$

\latticebody expaned at lattice points with

- \LatticeA \LatticeB (a,b) coordinate
- \LatticeX \LatticeY (X,Y) coordinate in pts

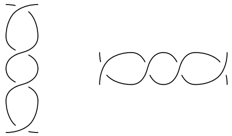
```
\def\latticebody{%
\ifnum\latticeA=1 \ifnum\latticeB=-1 %
\else \drop{\dir{+}}\fi\else
\ifnum\latticeA=0 \ifnum\latticeB=1\else
\drop{\dir{x}}\fi\else\drop{\circ}\fi\fi}
\begin{xy} +(2,2)="o",0*\xybox{%
  0;<3pc,1.5mm>:<0.72pc,1.65pc>::
 ,{"o" \croplattice{-4}4{-4}4{-2.6}{2.6}{-3}3}
 ,"o"+(0,1) ="a"*{\bullet}*+!D{\alpha}
 ,"o"+(1,-1)="b"*{\bullet}*+!L{\beta}
 ,"o"+(0,-1)="c","o"+(-1,1)="d"
 ,"a"."c"="e",!DR*{};"a"**\dir{.}
 ,"e",!UL*{};"c"**\dir{.}
 ,"b"."d"="f",!DL*{};"b"**\dir{.}
 ,"f",!UR*{};"d"**\dir{.}
 ,"e"."f"*\frm{.}}="L","o"."L"="L"
 ,{"L"+L \ar "L"+R*+!L{s^{(1)}}}
 ,{"L"+D \ar "L"+U*+!D{s^{(2)}}}
\end{xy}
```



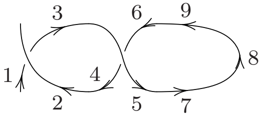
4.8. Knots and Links

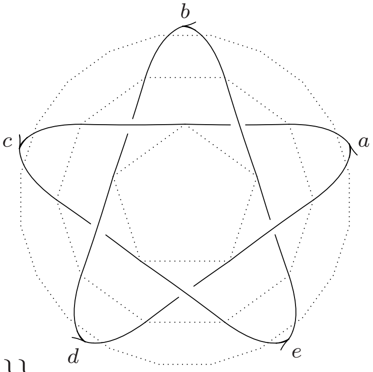
See Reference Manual for details.

```
$\xy
 0;/r1pc/:
 ,{\vunder\vtwist\vtwist\vunder-}
\endxy\qquad
\xy
 0;/r1pc/:+(0,-1.5)
 ,{\hover\hcross\hcross\hover-}
\endxy$
```



```
\begin{xy} /r9mm/:
 ,(0,0)
 ,{\hunder<><{1}|>|{2}>>>{3}%
    \htwist<<<{4}|>|{5}><>{6}%
     \hloop<><{7}|>|{8}>>>{9}}
\end{xy}
```





```
\[\knotholesize{2mm}
\xygraph{!{0;/r1cm/:}
 !P5"p"{~>{.}}
 !P10"d"{~:{(1.7,0):}~>{.}}
 !P20"D"{~={-9}~:{(2.2,0):}~>{.}}
 !{\xunderv~{"d3"}{"d2"}{"p2"}{"p1"}}
 !{\xunderv~{"d5"}{"d4"}{"p3"}{"p2"}}
 !{\xunderv~{"d7"}{"d6"}{"p4"}{"p3"}}
 !{\xunderv~{"d9"}{"d8"}{"p5"}{"p4"}}
 !{\xunderv~{"d1"}{"d10"}{"p1"}{"p5"}}
 !{\vloop~{"D3"}{"D2"}{"d2"}{"d1"}|>|{a}}
 !{\vloop~{"D7"}{"D6"}{"d4"}{"d3"}|>|{b}}
 !{\vloop~{"D11"}{"D10"}{"d6"}{"d5"}|>|{c}>}
 !{\vloop~{"D15"}{"D14"}{"d8"}{"d7"}|>|{d}}
 !{\vloop~{"D19"}{"D18"}{"d10"}{"d9"}|>|{e}}
}\]
```

4.9. Other options

Basic syntax of XY-pic

Top

Ex

pos	
pos`+`*cod*	*c* ← +*cod* (size of *cod* is kept)
pos`-`*cod*	*c* ← −*cod* (size of *cod* is kept)
pos`!`*cod*	*c* ← *pos*, skew *c* by *cod* (move reference point)
pos`.`*cod*	*c* ← *p* with covering *c* (size changes)
pos`,`*cod*	*c* ← *pos*, then *c* ← *cod*
pos`;`*cod*	*c* ← *pos*, swap *p* *c*, then *c* ← *cod*
pos`:`*cod*	set *x-base* and naturally arrange *y-base*
pos`::`*cod*	set *y-base*
pos`*`*obj*	*c* ← *pos*, then drop *obj* at *c*
pos`**`*obj*	*c* ← *pos*, connect from *p* to *c* using *obj*
pos`?`*place*	*c* ← *pos*, *c* ← *place*
pos`@`*stacking*	*c* ← *p*, do *stacking*
pos`=`*saving*	*c* ← *p*, do *saving*
cod	*c* ← *cod*
`c p x y`	*c*, *p*, axis intersections with $\overrightarrow{pc}$
`s`*digit*	stack position *digit* below the top (= `s`0)
`s{`*number*`}`	stack position *number* below the top
`"`*id*`"`	restored what is saved as `"`*id*`"` earlier
{*pos* *decor*}	the *c* resulting from interpreting the group
vector	*pos* is *vector* with zero size
`0`	zero
`<`*dimen*`,`*dimen*`>`	absolute
`<`*dimen*`>`	absolute with equal dimensions
`(`*factor*`,`*factor*`)`	in current base
`a(`*number*`)`	angle in current base (*number* is in degree)
corner	from reference point to *corner* of *c*
`L R D U CL CR CD CU C LD RD LU RU E P`	
corner(*factor*)	the *corner* multiplied with *factor*
/*direction* *dimen*/	vector *dimen* in *direction*
empty	
`@+`*cod*	push *cod*
`@-`*cod*	*c* ← *cod* then pop
`@=`*cod*	load stack with *cod*
`@@`*cod*	do *cod* for *c* ← stack
`@i`	initialize
`@(`	enter new frame
`@)`	leave current frame
`@:"`*id*`"`	`"`*id*`"` restores current base
`@`*cod*`"`*id*`"`	`"`*id*`"` reinterprets *cod*
`@@"`*id*`"`	`@="`*id*`"` reloads this stack

Ex

place	
<_place_	shave (0) to edge _p_, _f_←0
>_place_	shave (1) to edge _c_, _f_←1
(_factor_)_place_	_f_← _factor_
!{_pos_}_slide_	intercept with line setup by _pos_ and apply _slide_
slide	pick place ans apply _slide_
/_dimen_/	slide _dimen_ further along connection
empty	
obj	
modifier obj	apply _modifier_ to _obj_
objectbox	
{_text_}	build default object or text in TEX
library-obj	library object
@_dir_	(_dir_)ectional
\object _obj_	wrap up _obj_ as finished object box
\composite{_composite_}	build composit object box
\xybox{_pos decor_}	package entire XY-picture as object
composite	
_composite_*_obj_	add _obj_ to composite object box
obj	first object is required
modifier	
!_vector_	_obj_ has its reference point shifted by _vector_
!	_obj_ has the original reference point reinstalled
add-op size	change _obj_ size <_dimen_,_dimen_>
h i	_obj_ is given hidden, invisible
[_shape_]	_obj_ is given the specified _shape_
[=_shape_]	define _shape_ to reestabrish current object style
direction	set current _direction_ for this _obj_
diag	
l r d u ld rd lu ru	left diagonal, right diagonal,...
empty	last used direction
v_vector_	direction of _vector_
q{_pos decor_}	direction from _p_ to _c_ after _pos decor_
direction:_vector_	_vector_ relative to _direction_
direction orient	_orient_ to _direction_
add-op	
+	default size: +2×_objectmargin_
-	default size: -2×_objectmargin_
=	default size: =<_objectwidth_,_objectheight_>
+=	default size: +=<max($(L_c + R_c, D_c + U_c)$)>
-=	default size: -=<min($(L_c + R_c, D_c + U_c)$)>

orient	`_` (clockwise) `^` (conterclockwise)
shape	
`. o r l u d` etc.	many optional ones `|(num)` `@!number`
empty	
`\objectmargin`*add-op*{*dimen*}	set default object margin
`\objectwidth`*add-op*{*dimen*}	set default object width
`\objectheight`*add-op*{*dimen*}	set default object height
`\xy` *pos* *decor* `\endxy`	build a box with an XY-picture
`\everyxy={`*text*`}`	typed literally right after each `\xy` command
command	
`\save` *pos*	save state, and do *pos*
`\restore`	restore state by saved maching `\save`
`\POS` *pos*	interpret *pos*
`\afterPOS` {*decor*} *pos*	interpret *pos* and then perform *decor*
`\drop` *obj*	drop *obj* as *pos* `*`*obj*
`\connect` *obj*	connect with *obj* as *pos* `**`*obj*
`\relax`	do nothing
TEX-command	any TEX command
`\xyverbose \xytracing \xyquiet`	tracing command
`\xyignore`	ignore XY-command
`\xycompile`{*pos* *decor*}	compile to file *prefix no*.xyc
`\xycompileto`{*name*}{*pos* *decor*}	compile to file *name*.xyc
decor	
command decor	
empty	
c	current position
p	previous position
digit	one digit
number	an integer number
factor	a real number
dimen	a length in TEX
letter	a letter (a usual character) in TEX
id	a string in TEX
text	a text in TEX
space	a space
empty	empty

Kernel object library

`\dir` *dir*	a directional object (a kernel object)	Ex
dir		
variant{*main*}	see §1.7 for *main* (in kernel library)	
variant	`_ ^ 2 3`	
`\cir` *radius*{*cir*}	a circle object (a kernel object)	Ex
radius		
dimen		
vector	use X of the *vector* as radius	
empty	use R_c as radius	
cir	partial circle segment with *orient*ation	
diag orient diag	*diag*s are start/end directions along circle	
empty	full circle	
`\txt<`*width*`>` *style*{*text*}	*text* is typeset to *width* with *style*(a kernel object)	
width	`\\` can be used in *text*. All lines are centered.	
dimen		Ex
empty		
style	a font command etc. for each line	

Extensions

`**\crvs`{*dir*}	get *polist* from the stack	Ex
`\curve`*modifier*{*curve-object polist*}	as a *decor*ation	
`\crv`*modifier*{*curve-object polist*}	`\curve` equals `\connect\crv`	
modifier		
`~`*curve-option modifier*	set *curve-option*	
empty	default is `~C`	
curve-option		
`p P l L c C pc pC Pc PC lc lC Lc LC cC`		
curve-object		
`~*`*obj curve-object*	specify the drop object	
`~**`*obj curve-object*	specify connect pbject	
empty		
polist		
pos `&` *polist*	list f positions for control points	
`~@`	add the current stack to the control points	
`~@ &` *polist*		
empty		
`\xycircle`*vector*{*style*}		Ex
`\qspline`{*style*}		
`\frm`{*style*}		Ex

`\SelectTips{`*family*`}`

family

  `xy cm eu`

`\xyimport(`*width*`,`*height*`){`*graphic*`}`

`\xyimport(`*width*`,`*height*`)(`*x-off*`,`*y-off*`){`*graphic*`}`

Features

`\afterPATH {`*decor*`}`*path*	Interpret *path* and then run *decor*
`\PATH` *path*	Interpret *path* Ex
path	
`~`*action*`{`*stuff*`}`*path*	set *action* to *stuff*
`~`*which*`{`*labels*`}`*path*	add *labels* prefix for some segments
`~{`*stuff*`}`*path*	set failure continuaton to *stuff*
`'`*segment path*	make straight segment
`` ` ``*turn segment path*	make turning segment
segment	make last segment
path-pos slide labels	segment with *slide* and *labels*
action	`=` `/` use *stuff* before/after each segment
which	`<` `>` next (last) segment only `=` every segmant
turn	
diag turnradius	1/4 turn starting in *diag*
cir turnradius	explisit turn
turnradius	
`/`*dimen*	set turnradius to *dimen*
empty	use default turn radius
slide	
dimen	*dimen* in the “above” direction
empty	
labels	
`^`*anchor it alias labels*	label with *it*em above *anchor*
`_`*anchor it alias labels*	label with *it*em below *anchor*
`|`*anchor it alias labels*	break with *it*em at *anchor*
empty	
anchor	
`-`*anchor*	
place	
alias	
`="`*id*`"`	optional name for label object
empty	

it	
digit	0 1 2 3 4 5 6 7 8 9
letter	0 ··· A B ··· y z
{text}	usual text
cs	
*obj	object
@dir	directory
[shape] it	use [shape] for item
\labelmargin add-op{dimen}	
\turnradius add-op{dimen}	default: 10pt
\ar form* path	make arrow along path
form* represents form form ···	(may be empty)
form	
@variant	use variant of arrow
@variant{tip}	use standart stem and tip for head
@variant{tip conn tip}	use tip conn tip as tail, stem, head
@connchar	change stem by connchar
@!	dash the arrow stem by doubling it
@/direction dist/	curve arrow the distance towards direction
@(direction,direction)	curve fit with in-out directions
@‘{control-point-list}	curve setup with explicit control points
@[shape]	add [shape] to modifiers for all objects
@{modifier*}	add modifier for all objects
@<dimen>	slide arrow by dimen
|anchor it	break each segment at anchor with item
^anchor it	label each segment at ^anchor with item
_anchor it	label each segment at _anchor with item
@?	reverse meaning of ^ and _
variant	
^ _ 0 1 2 3	
empty	
tip	
tipchar*	
dir	directional
tipchar	< > ( ) | ’ ‘ + / letter space
conn	
conchar*	
dir	directional
conchar	- . ~ = :

Ex Top

```
\xymatrix setup*{
 entry & entry & ··· \\
 entry & entry & ··· \\
 ··· }
```

setup	
@R add-op dimen	change row spacing
@C add-op dimen	change column spacing
@ add-op dimen	change row and column spacing
@!R	every row spacing is the maximal in entries
@!C	every column spacing is the maximal in entries
@!	@!R @!C
@!0	entries have size 0, @!R0 @!C0 are possible
@!=dimen	entries have size dimen, @!R=dimen etc are OK
@1	suited for one-line matrices
@direction	orientation, r is the default of direction
@*[shape]	apply to every entries
@*add-op size	apply to every entries
@em add-op dimen	set size for entries
em	
M	entry margin
W	entry width
H	entry height
L	label separation for label
entry	
{text}	a math text, sometimes { } may be omitted
*obj pos decor	
**[shape] entry	
**[modifier*] entry	
"r,c"	entry row r and column c, top left is "1,1"
[Δr,Δc]	Δr rows below and Δc columns right
[hop*]	entry reached by hop*
[hop+place]	place on straight line to non-empty hop*
hop	
r l u d	right left up down
"prefix r,c" ["prefix" Δr,Δc]	entry from the matrix prefix
["prefix" hop*] ["prefix" hop+place]	entry from the matrix prefix
\xymatrixrowsep add-op dimen	change row separation (default 2pc)
\xymatrixcolsep add-op dimen	change column separation (default 2pc)
\entrymodifiers={modifier*}	
\everyentry={decor}	

\xygraph{graph}	Setup graph which equals step*	Ex
step		Top
-arrow node labels	draw line to node with labels	
:arrow node labels	draw arrow to node with labels	
(list)	map current node over list	
node	move to the node	
[move]	new node moved relative to current	
&	new node is next column	
\\	new node is next row	
"id"	previously saved node	
?	currently mapped node	
node it	node with it typeset and saved there	
node="id"	node saved as ”id”	
node!escape	node augment node with material in another node	
move		
hop*	hops (d u l r) from current node	
hop* place move	do hops but use place and move again	
list		
graph, list		
graph		
escape		
{pos,decor}	perform pos decor	
M matrix	insert matrix	
P polygon	insert polygon	
E ellipse	insert ellipse	
~ setup	setup paramaters	
!~setup		
!~:{arrow}	include with every : arrow	
!~-{arrow}	include with every - line	
!~*{modifier}	include with every non-* node	
!~letter{graph}	define new graph escape !letter	
\newgraphescape{letter}#1#2···{graph}		
\xypolygon number "prefix" {switches···}		Ex
~:{...}	for rescaling	
~*{obj}	obj at each vertex	
~={angle}	aligne first vertex	
~<{...}	directional for ”spokes”	
~<<{arrow}	use arrow for ”spokes”	
~<>{...}	labels and breakes on ”spokes”	
~>{...}	directional for ”slides”	
~><{arrow}	use arrow for ”slides”	
~>>{...}	labels and breaks in ”slides”	