

X_Y-pic (主に xymatrix) の使い方

より詳しくは, X_Y-pic の配布物に含まれる User's Guide (xyguide.dvi) 等を参照のこと.

目次

1	基本	2
1.1	読み込み	2
1.2	項目の配置	2
1.3	矢印	3
1.4	ラベル	3
2	矢印	4
2.1	ラベルの位置	4
2.2	いろいろな矢印	5
2.3	スライドする	6
2.4	穴を空ける	7
2.5	曲げる	7
2.6	曲げる (2)	8
3	サイズ等	9
3.1	行間列間のサイズを変える	9
3.2	強制的に行間列間をそろえる	9
3.3	項目の余白	11
3.4	項目やラベルのサイズ	11
3.5	回転	12
4	Tips	12
4.1	数式番号	12
4.2	平行な矢印	13
4.3	□ 等	14
4.4	包含射の一例	15
4.5	要素の対応	16
4.6	pullback, pushout	17
4.7	いろいろ	18
5	xy	18
5.1	xy	18

6	xypolygon	20
6.1	基本	20
6.2	回転と拡大縮小	21
6.3	通し番号等	21
6.4	その他	23
A	! TeX capacity exceeded	23

1 基本

1.1 読み込み

X_Y-pic を使用するためには, まず X_Y-pic を読み込む必要がある.

具体的には`\usepackage[all]{xy}`という一行を`\begin{document}`の前に書く.

また default では xymatrix 中の数式は文中数式モードになるようなので

`\def\objectstyle{\displaystyle}`を書いておいた方が使いやすいかもしれない.

```
\documentclass[12pt]{jarticle}
\usepackage{amsmath,amsfonts,amsthm,amssymb,amscd}
\usepackage[all]{xy}

\def\objectstyle{\displaystyle}

\begin{document}
```

1.2 項目の配置

```
\[
\matrix{
A & B & C \\
D & E & \\
& F & G
}
\]
```



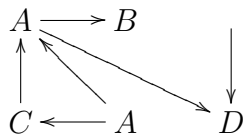
A	B	C
D	E	
	F	G

1.3 矢印

矢印は、その矢印の出発点である項目の後に `\ar[方向]`. 方向は `r,l,u,d` の組み合わせ.

```
\[
\xymatrix{
A \ar[r] \ar[rrd] & B & \ar[d] \\
C \ar[u] & A \ar[l] \ar[lu] & D
}
\]
```

↓



1.4 ラベル

矢印にラベルをつける. 位置は `^,_,|` で指定. 矢印の向きとラベルのつく位置に注意.

```

\[
\xymatrix{
A \ar[r]^f & B \ar[d]^f \\
& A \ar[r]_{g_1} & B \ar[d]_{g_1} \\
& A \ar[r]|h & B \ar[d]|h \\
D \ar[u]^f & C \ar[l]^f \\
& D \ar[u]_{g_1} & C \ar[l]_{g_1} \\
& D \ar[u]|h & C \ar[l]|h
}
\]

```

↓

$$\begin{array}{ccc}
\begin{array}{ccc} A & \xrightarrow{f} & B \\ f \uparrow & & \downarrow f \\ D & \xleftarrow{f} & C \end{array} &
\begin{array}{ccc} A & \xrightarrow{g_1} & B \\ g_1 \uparrow & & g_1 \downarrow \\ D & \xleftarrow{g_1} & C \end{array} &
\begin{array}{ccc} A & \xrightarrow{h} & B \\ h \uparrow & & \downarrow h \\ D & \xleftarrow{h} & C \end{array}
\end{array}$$

2 矢印

2.1 ラベルの位置

項目のサイズによっては矢印のラベルの位置がずれる。

```

\[
\xymatrix{
A \ar[r]^f & BCDEFGH
}
\]

```

↓

$$A \xrightarrow{f} BCDEFGH$$

ラベルの前に-をつけると自動で調整してくれる。特に位置を指定する必要の無い場合は常に-をつけるようにするとよいだろう。

```
\[
\xymatrix{
A \ar[r]^-f & BCDEFGH
}
\]
```



$$A \xrightarrow{f} BCDEFGH$$

ラベルの前に (. 数字) で位置を指定することも出来る.

```
\[
\xymatrix{
A \ar[r]^{(.2)f}_{(.25)g} & BCDEFGH
}
\]
```



$$A \xrightarrow[\substack{f \\ g}]{} BCDEFGH$$

2.2 いろいろな矢印

`\ar@{スタイル}[方向]` でいろいろな矢印が書ける. どのようなスタイルが使えるかについては `xyguide` 等を参考のこと.

```

\[
\xymatrix{
A \ar@{=} [r]
& A \ar@{.} [r]
& B \ar@{->>} [r] & C \ \
D \ar@{|-\hat{>}} [r]
& E \ar@{\hat{({}->}} [r]
& F \ar@{>->} [r] & G
}
\]

```

↓

$$\begin{array}{ccccccc}
 A & \xlongequal{\hspace{1cm}} & A & \xrightarrow{\hspace{1cm}} & B & \xrightarrow{\hspace{1cm}} & C \\
 D & \xrightarrow{\hspace{1cm}} & E & \xrightarrow{\hspace{1cm}} & F & \xrightarrow{\hspace{1cm}} & G
 \end{array}$$

2.3 スライドする

矢印を平行に移動するには`\ar@<移動量>[方向]`とする。上記のいろいろな矢印と併用するときは`\ar@<移動量>@{スタイル}[方向]`等とする。移動量とスタイルの順序はどちらでもよいようだ。

```

\[
\xymatrix{
A \ar[r]^-f
& B \ar@<-0.5ex>[r]_-f
& C \ar@<1ex>[r]^-f
& D \ar@<0.5ex>[d]^-f \ \
E \ar@<-0.3ex>@{\hat{({}->}}[r]^-f
& F \ar@{\hat{({}->)}@<0.3ex>[r]^-f
& G \ar@<0.5ex>[r]^-f \ar@<-0.5ex>[r]_-g
& H \ar@<0.5ex>[u]^-g
}
\]

```

↓

$$\begin{array}{ccccccc}
 A & \xrightarrow{f} & B & \xrightarrow{f} & C & \xrightarrow{f} & D \\
 & & & & & & \uparrow g \\
 E & \xrightarrow{f} & F & \xrightarrow{f} & G & \xrightarrow{f} & H \\
 & & & & & & \downarrow f
 \end{array}$$

2.4 穴を開ける

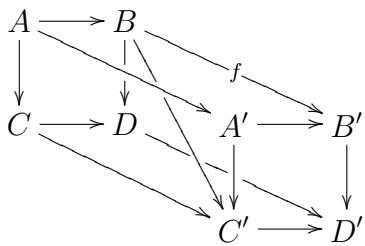
`\ar[方向]|\hole` で穴が開く。穴の位置の指定はラベルの位置の指定と同じ。

```

\l
\xymatrix{
A \ar[r] \ar[d] \ar[rrd]
& B \ar[rrd] | f \ar[d] |\hole \ar[rdd] | (.33)\hole & & \l \\
C \ar[r] \ar[rrd] & D \ar[rrd] | (.33)\hole |\hole \\
& A' \ar[r] \ar[d] & B' \ar[d] \l \\
& & C' \ar[r] & D'
}
\l

```

⇓



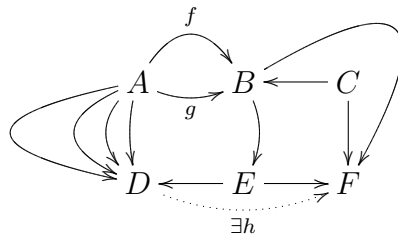
2.5 曲げる

`\ar@/^曲げる量/[方向]` で進行方向左側に、`\ar@/_曲げる量/[方向]` で進行方向右側にふくらんだ矢印になる。

```

\[
\xymatrix{
A \ar@/^18pt/[r]^f \ar@/_/[r]_g \\
\ar@/_3pt/[d] \ar@/_12pt/[d] \ar@/_24pt/[d] \ar@/_48pt/[d] \\
& B \ar@/^/[d] \ar@/^54pt/[rd] & C \ar[l] \ar[d] \\
D \ar@/_10pt/@{.}>[rr]_{\exists h} & E \ar[r] \ar[l] & F
}
\]

```

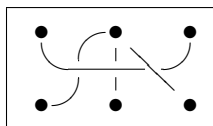


2.6 曲げる (2)

```

\fbbox{
\xymatrix@=5pt{
\bullet \ar@{-} 'd[dr] ' [rrrd] ' [rrrr] [rrrr] \\
& \bullet \ar@{-} ' [d] [dd] \ar@{-} [rrdd] \\
& \bullet \\
& \bullet & \bullet \\
\bullet \ar@{-} 'r[ruu] ' [ru] ' [rruu] [rruu] \\
& \bullet & \bullet & \bullet
}
}

```



3 サイズ等

3.1 行間列間のサイズを変える

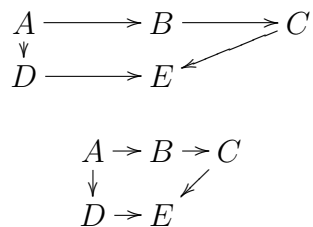
@R= で行間, @C=で列間のサイズを指定. @= で行間列間のサイズを一斉に指定.

```
\[
\xymatrix@C=36pt@R=6pt{
  A \ar[r] \ar[d] & B \ar[r] & C \ar[ld] \\
  D \ar[r] & E & \\
}
\]
```



```
\[
\xymatrix@=10pt{
  A \ar[r] \ar[d] & B \ar[r] & C \ar[ld] \\
  D \ar[r] & E & \\
}
\]
```

↓



3.2 強制的に行間列間をそろえる

特に何も指定しなければ, 項目のサイズに合わせて適当に行間列間を調整してくれるが, 行間列間をそろえたい場合は@!, @!R, @!Cを使うとよい. @!=サイズ等でサイズの指定も可.

```

\[
\xymatrix{
& A \ar[ld] \ar[d] \ar[rd] & \\
B & C & DDDDDDDD
}
\]

```

```

\[
\xymatrix@!C{
& A \ar[ld] \ar[d] \ar[rd] & \\
B & C & DDDDDDDD
}
\]

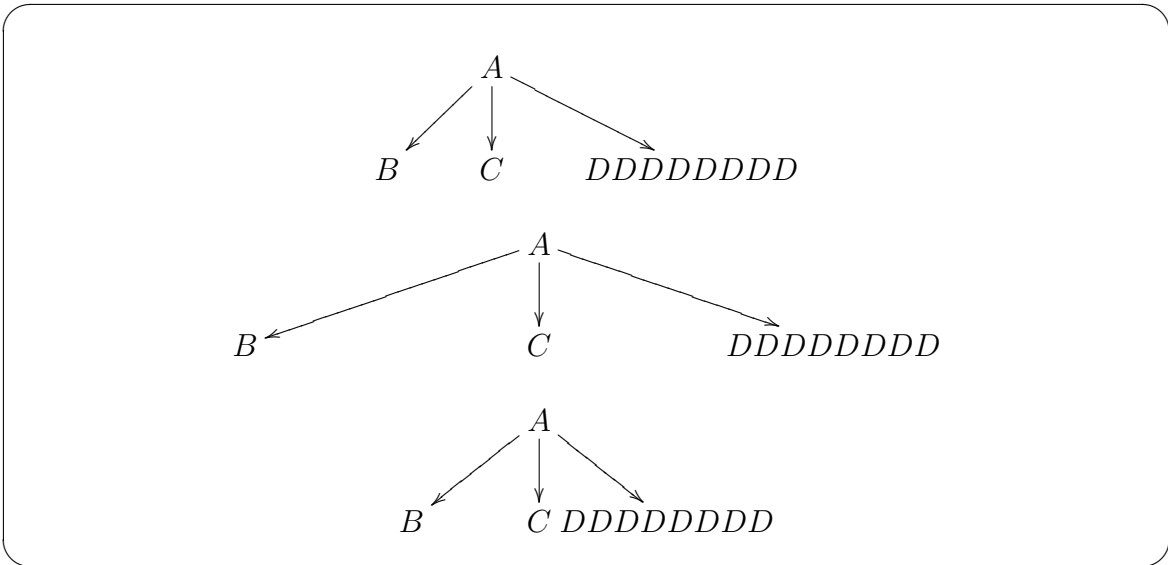
```

```

\[
\xymatrix@!C=24pt{
& A \ar[ld] \ar[d] \ar[rd] & \\
B & C & DDDDDDDD
}
\]

```

⇓



3.3 項目の余白

@M=で余白のサイズを指定.

```
\[
\xymatrix{
  A \ar[r] & B \ar@{>->}[r] & C
}
\]
```

```
\[
\xymatrix@M=8pt{
  A \ar[r] & B \ar@{>->}[r] & C
}
\]
```

↓

$$A \longrightarrow B \longrightarrow C$$

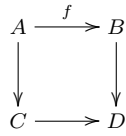
$$A \longrightarrow B \xrightarrow{\hspace{1cm}} C$$

3.4 項目やラベルのサイズ

`\def\objectstyle{サイズの指定}`で項目の、`\def\labelstyle{サイズの指定}`でラベルのサイズを変更出来る。サイズの指定にどのようなものが使えるのかはよくわからない。

```
\[
\def\objectstyle{\scriptstyle}
\def\labelstyle{\scriptscriptstyle}
\xymatrix{
  A \ar[r]^f \ar[d] & B \ar[d] \\
  C \ar[r] & D &
}
\]
```

↓



3.5 回転

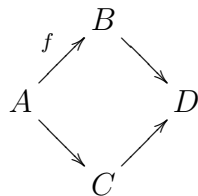
@方向で図式を、左上の項目を中心に、回転させることができる。方向は ur u ul l dl d dr で指定するようだ。

```

\[
\xymatrix@ur{
A \ar[r]^f \ar[d] & B \ar[d] \\
C \ar[r] & D
}
\]

```

⇓



4 Tips

4.1 数式番号

`xymatrix` でかいた図式に数式番号をつけるときは `\vcenter` を使うとよいかもしれない。

`\verb+\vcenter+`をつけない場合

```
\begin{equation}
\begin{matrix}
A \ar[r] \ar[d] & B \ar[d] \\
C \ar[r] & D \ar@{}[lu]|{\circlearrowright}
\end{matrix}
\end{equation}
```

つけた場合

```
\begin{equation}
\vcenter{
\begin{matrix}
A \ar[r] \ar[d] & B \ar[d] \\
C \ar[r] & D \ar@{}[lu]|{\circlearrowright}
\end{matrix}
}
\end{equation}
```

↓

`\vcenter` をつけない場合

$$\begin{matrix} A & \longrightarrow & B \\ \downarrow & & \downarrow \\ & \circlearrowright & \\ C & \longrightarrow & D \end{matrix} \quad (1)$$

つけた場合

$$\begin{matrix} A & \longrightarrow & B \\ \downarrow & & \downarrow \\ & \circlearrowright & \\ C & \longrightarrow & D \end{matrix} \quad (2)$$

4.2 平行な矢印

平行な矢印をたくさん使う場合は以下のようなマクロを用意すると便利.

```
\newcommand{\prarrow}[2]{\ar@<0.5ex>[r]^-{\#1} \ar@<-0.5ex>[r]_-{\#2}}
\newcommand{\plarrow}[2]{\ar@<0.5ex>[l]^-{\#1} \ar@<-0.5ex>[l]_-{\#2}}
\newcommand{\pdownarrow}[2]{\ar@<0.5ex>[d]^-{\#1} \ar@<-0.5ex>[d]_-{\#2}}
\newcommand{\puarrow}[2]{\ar@<0.5ex>[u]^-{\#1} \ar@<-0.5ex>[u]_-{\#2}}
```

```
\[
\xymatrix{
A \prarrow{f}{g} \pdownarrow{h}{k} & B \\
C & D \plarrow{u}{v} \puarrow{s}{t}
}
\]
```

↓

$$\begin{array}{ccc}
A & \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} & B \\
\begin{array}{c} \downarrow k \\ \downarrow h \end{array} & & \begin{array}{c} \uparrow s \\ \uparrow t \end{array} \\
C & \begin{array}{c} \xleftarrow{v} \\ \xleftarrow{u} \end{array} & D
\end{array}$$

本文中で使うには以下のようにマクロに`\xymatrix`を入れたものの方が便利だろう。1行からなる図式を書くときには、ここに出ている`@1`という指定をするとよいそうである。

```
\newcommand{\paar}[3]{\xymatrix@1{#1 \ar@<0.5ex>[r]^-{{#2}_1}%
\ar@<-0.5ex>[r]_-{{#2}_2} & #3}}
```

```
\newcommand{\paaar}[3]{\xymatrix{#1 \ar@<0.5ex>[r]^-{{#2}_1}%
\ar@<-0.5ex>[r]_-{{#2}_2} & #3}}
```

`\verb+@1+`を指定したものの`\paar{A}{f}{B}`と、
指定していないものの`\paaar{A}{f}{B}`
はこのようになるが、どうであろうか。

↓

`@1`を指定したものの $A \begin{array}{c} \xrightarrow{f_1} \\ \xrightarrow{f_2} \end{array} B$ と、指定していないもの $A \begin{array}{c} \xrightarrow{f_1} \\ \xrightarrow{f_2} \end{array} B$ はこのようになるが、どうであろうか。

4.3 \prod 等

\prod や \lim の添字が下だけにつく場合にバランスが悪くなる場合があるので以下のよ
うなマクロを使うとよいかもしれない。

```

\newcommand{\xyprod}{\prod^{\phantom{A}}}
\newcommand{\xycoprod}{\coprod^{\phantom{A}}}
\newcommand{\xydlimi}[1]{\varinjlim\limits_{#1}^{\phantom{A}}}
\newcommand{\xyilimi}[1]{\varprojlim\limits_{#1}^{\phantom{A}}}

```

```

\[
\matrix@1{
\prod_{i \in I} A_i \ar[r] & B \ar[r] & \varinjlim_{i \in I} C_i
}
\]

\[
\matrix@1{
\xyprod_{i \in I} A_i \ar[r] & B \ar[r] & \xydlimi_{i \in I} C_i
}
\]

```

⇓

$$\prod_{i \in I} A_i \longrightarrow B \longrightarrow \varinjlim_{i \in I} C_i$$

$$\prod_{i \in I} A_i \longrightarrow B \longrightarrow \varprojlim_{i \in I} C_i$$

4.4 包含射の一例

```

\[
\matrix@0=18pt{
S \times_X \ar[r] \ar@{}[d]|\{\bigcap\} & S \ar@{}[d]|\{\bigcap\} \\
Y \ar[r] & R \ar@{}[r]|\{\subset\} & X.
}
\]

```

⇓

$$\begin{array}{ccc}
 S \times_X Y & \longrightarrow & S \\
 \cap & & \cap \\
 Y & \longrightarrow & R \subset X.
 \end{array}$$

4.5 要素の対応

```

\[
\xymatrix@C=36pt@R=2.8pt{
  A \ar[r]^f & B \\
x \ar@{(-)}[u] \ar@{|->}[r] & y \ar@{(-)}[u]
}
\]

\[
\xymatrix@1{
  A \ni x \ar@{|->}[r]^f & y \in B
}
\]

\[
A \ni x \overset{f}{\longmapsto} y \in B
\]
\]

```

↓

$$\begin{array}{ccc}
 A & \xrightarrow{f} & B \\
 \downarrow & & \downarrow \\
 x & \longmapsto & y
 \end{array}$$

$$A \ni x \xrightarrow{f} y \in B$$

$$A \ni x \overset{f}{\mapsto} y \in B$$

これは `\rotatebox` 等を使った方がよいか。

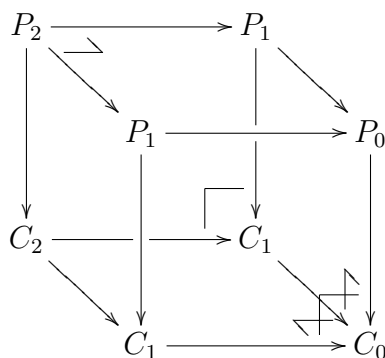
4.6 pullback, pushout

pullback や pushout の図式の角につけるマークの一例. web で見つけたものを少し変更したものだったと思うが, どこで見つけたものか分からなくなりました.

```
\newcommand{\pullbackmark}[2]{\save ;p+<.8pc,0pc>:(0,-1)::%
(#1) *{\phantom{Z}} %
;p+(#2)-(0,0) **@{-}%
;p-(#1)+(0,0) *{\phantom{Z}} **@{-} \restore}
```

```
\[
\xymatrix{
P_2 \pullbackmark{2,0}{1,1}
\ar[rd] \ar[dd] \ar[rr] & & \\
P_1 \ar[dd] |{\hole} \ar[rd] & & \\
& P_1 \ar[dd] \ar[rr] & & P_0 \ar[dd] \\
C_2 \ar[rr] |{\hole} \ar[rd] & & \\
\pullbackmark{-2,0}{0,-2} C_1 \ar[rd] & & \\
& C_1 \ar[rr] & & \\
\pullbackmark{-1,-1}{0,-2} \pullbackmark{-1,-1}{-2,0}
\pullbackmark{-2,0}{0,-2} C_0
}
\]
```

↓

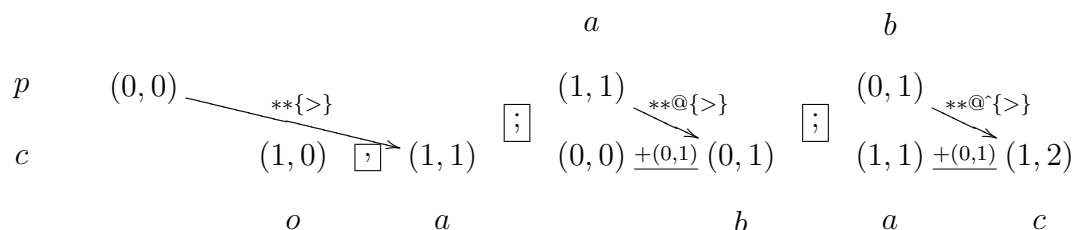


- コンマ , の後に位置を指定すると点 c がそこに移動する.
- セミコロン ; を書くと点 p と点 c が入れ替わる.
位置の指定は $\langle 1\text{pt}, 2\text{pt} \rangle$ 等とすると (xy 全体の基点を中心とした) 絶対的な指定になる. $(0, 1)$ 等とすると現在の基底による位置指定になる.
- : で点 p が原点, $c - p$ が第一ベクトル, それに直交するベクトルが第二ベクトルであるような基底が指定される.
- :: で $c - p$ が第二ベクトルになる.
- default の基底は $\langle 1\text{mm}, 0\text{mm} \rangle, \langle 0\text{mm}, 1\text{mm} \rangle$ である.
- +vector で点 c は vector の分だけ移動する. ただし vector として $(0, 1)$ 等を使うと, xy 全体の基点からみた点 $(0, 1)$ の位置ベクトルの分だけ点 c から移動することになるので, : で基底を指定している場合は注意が必要.
- *{hoge} で点 c の位置に hoge が置かれる.
- **{huga} で点 p と点 c が huga でむすばれる.
- **@{huga} は xymatrix の矢印と同じようなもの.

```
\xy <1cm,0cm>:
(1,0)*{o}, (1,1)*{a} **{>};
+(0,1)*{b} **@{>};+(0,1)*{c} **@^>
\endxy
```



上のコードで, 点 p と c がどのように移動し, どこに o, a, b, c がおかれ, どの点が $**\{>\}$ 等で結ばれるかを示した図.



6 xypolygon

6.1 基本

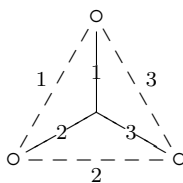
`\xypolygon` を使った例. これを使うためには`\xyoption{poly}` という 1 行をどこか (普通は preamble の部分だろうか) に入れる必要がある.

- `~*{hoge}` 頂点に hoge をおく.
- `~<{hoge}` spoke (中心と頂点を結ぶ線) として hoge を使う.
- `~>{hoge}` 辺として hoge を使う.

辺や spoke に label をつけるときは

- `~<<{arrow}` spoke として arrow で表される矢印を使う.
- `~<>{hoge}` spoke の矢に (xymatirx と同様な) label をつける.
- `~>>{arrow}` 辺として arrow で表される矢印を使う.
- `~<>{hoge}` 辺の矢に (xymatirx と同様な) label をつける.

```
\[
\xy /r3pc/:
\xypolygon3{~*{\circ}~<<{@{-}}~<>{|-{\xypolynode}}
~>>{@{--}}~>>{_-{\xypolynode}}
\endxy
\]
```

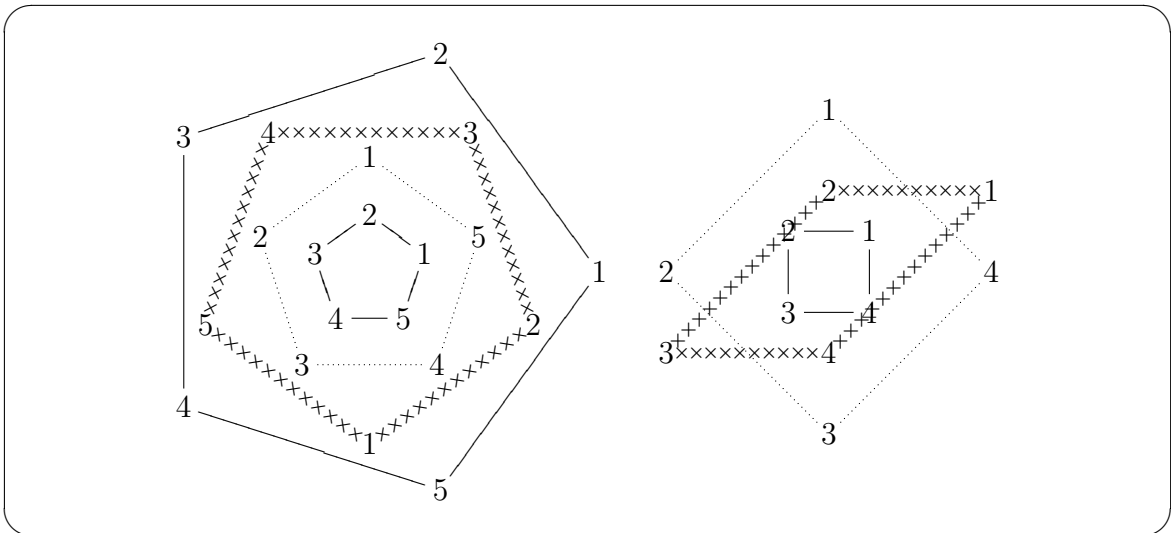


6.2 回転と拡大縮小

図形の回転は `~={}` で角度を指定. 回転の角度は, 水平右向きを 0 度, 反時計回り. 何も指定しなければ, 12 角形までは, 底が水平かつ 1 番目の頂点の偏角が最小になる向きに描かれる. 頂点の数が 13 以上になると, 1 番目の頂点の偏角が 0 になるように描かれる.

拡大縮小は `~:{}_` で指定.

```
\[
\xy /r1.8pc/:
{\xypolygon5{~*{\xypolynode}}},
{\xypolygon5{~={90}~:{(2,0):}~*{\xypolynode}~>{.}}},
{\xypolygon5{~={-90}~:{(3,0):}~*{\xypolynode}~>{x}}},
{\xypolygon5{~={0}~:{(4,0):}~*{\xypolynode}}},
+(8,0),
{\xypolygon4{~*{\xypolynode}}},
{\xypolygon4{~:{(2,2):}~*{\xypolynode}~>{.}}},
{\xypolygon4{~:{(2,0):(1,1):}~*{\xypolynode}~>{x}}}
\endxy
\]
```



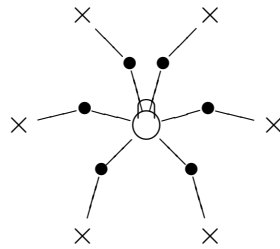
6.3 通し番号等

`polygon` に名前をつけてその頂点等を利用出来る.

`\xypolynode` は各 `polygon` ごとにリセットされる. 以下のようにすると頂点に通し番

6.4 その他

```
\[
\xy /r1pc/:
(0,0.25)*{\cap},(0,-0.25)*{\bigcirc}="0",
{\xypolygon12"C"~:{(2,0):}~*{}~>{}}},
{\xypolygon6"D"~:{(4,0):}~*{\times}~>{}}},
"C1"*{\bullet};"D1"*@{-},"C3"*{\bullet};"D2"*@{-},
"C4"*{\bullet};"D3"*@{-},"C6"*{\bullet};"D4"*@{-},
"C8"*{\bullet};"D5"*@{-},"C11"*{\bullet};"D6"*@{-},
"C1";"0"*@{-},"C3";"0"*@{-},"C4";"0"*@{-},"C6";"0"*@{-},
"C8";"0"*@{-},"C11";"0"*@{-}
\endxy
\]
```



A ! TeX capacity exceeded

最近の T_EX であればはじめから十分なサイズがとってあるので問題無いとは思いますが、コンパイルの際

! TeX capacity exceeded, sorry [main memory size=数字]

というエラーが出る場合は以下のように main_memoy の値を変更すればよい。

通常 T_EX の様々なファイルは texmf という directory(フォルダ)に入っている。texmf は、unix 系の OS(Mac の OS X も)であれば多分

/usr/local/share/

あるいは

/usr/local/teTeX/share

に、Windows であれば多分

C:\usr\local\share\

に入っているのではないかと思う。

以下が main_memory の値の変更方法である.

1. main_memory の値は texmf 内の web2c に入っている texmf.cnf というファイルに記述されている. このファイルの
main_memory = "数字" % words of inmemory available;
という行を探し, ここの”数字”を 700000 ~ 1000000 に変更する.
2. platex.ini というファイルが, 多分
texmf/ptex/platex/config
に入っているのので, それを -ini オプションをつけて ptex にかける.
ptex -ini platex.ini
platex.fmt というファイルが出来るのでこれを
texmf/web2c
に入っている platex.fmt とおきかえる. 念のためもとの platex.fmt は名前を付け
変えて (platex.fmt.org とか platex.fmt.bak とか) おいておく.