

El paquete Xy-pic

1 Introducción

Xy-pic¹ es un macro T_EX diseñado por K.H. Rose y R. Moore que permite construir fácilmente diagramas en T_EX. Desde L^AT_EX 2_ε se carga mediante

```
\usepackage[all]{xy}
```

La forma más cómoda de crear un diagrama es mediante una matriz. Las matrices se crean mediante la orden `\xymatrix`. La sintaxis básica es similar a la del L^AT_EX. Por ejemplo, la instrucción

```
\xymatrix{ A & B & C \\ D & E }
```

produce

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

Vemos que no es necesario completar las filas. Podemos incluir diagramas en modo texto o en modo matemático, pero en cualquier caso las entradas de las matrices se procesarán en modo matemático. Es recomendable no empezar nunca una entrada mediante un comando T_EX, usando llaves si es preciso en casos en que en L^AT_EX serían redundantes.

2 Flechas

Veamos ahora cómo introducir flechas que conecten distintas entradas de una matrix. La orden básica es `\ar` (de *arrow*). En su forma más simple, la flecha partirá de la entrada donde figura la orden y llegará a la entrada que se especifique mediante una combinación de letras

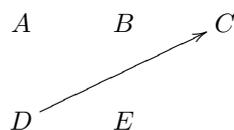
l = left, r = right, u = up, d = down.

Por ejemplo, mediante

¹El logotipo Xy-pic se obtiene mediante `\Xy-pic`

```
\xymatrix{ A & B & C \\ D & E }
```

Se obtiene



La orden crea una flecha que apunta a dos lugares a la derecha y un lugar hacia arriba. El orden de las letras `rru` es irrelevante, `rur` produce el mismo efecto. Alternativamente, podríamos sustituir `[rru]` por `[-1,2]`, donde -1 indica una posición hacia arriba y 2 dos posiciones hacia la derecha.² Aquí el orden sí importa. Con ambas notaciones, el final de la flecha se especifica en términos relativos a la entrada donde figura la orden `\ar`. También pueden usarse coordenadas absolutas, para lo cual hemos de cambiar los corchetes por comillas. Por ejemplo, la instrucción

```
\xymatrix{ A & B & C \\ D & E }
```

produce el mismo efecto, pero ahora "1,3" indica que la flecha ha de acabar en la fila 1 columna 3. También es posible especificar explícitamente el origen de la flecha, de modo que la posición que ocupa la orden dentro de la matriz se vuelve irrelevante. Para ello separamos el origen del destino mediante un punto y coma. La instrucción

```
\xymatrix{ A \ar[d]; [rr] & B & C \\ D & E }
```

produce el mismo efecto una vez más. La orden `\ar` se encuentra en la posición "1,1" e indica que la flecha debe partir de la entrada de abajo `d` a la entrada situada dos lugares a la derecha `rr`. No es necesario que el origen y el destino se especifiquen mediante la misma notación. Por ejemplo, también serviría

²Es importante no dejar espacios en blanco en estos puntos. Si escribimos `[rr u]` o (lo que es más fácil) `[-1, 2]` provocaremos un error.

`\xymatrix{A\ar[1,0];[rr] & B & C \ \ D & E}` `\xymatrix{ A \ \ \ar@{>--}[r] & B }`

Si la dirección de destino es la propia entrada donde está la instrucción, puede especificarse simplemente como `[]`.

No es necesario apuntar hacia una de las entradas de la matrix. Un destino puede ser modificado sumándole o restándole un vector. Así, en vez de `[rr]` podemos poner `[rr]+<3mm,6mm>`. El efecto es que la flecha va 3mm más a la derecha y 6 mm más arriba. Las flechas se calculan teniendo en cuenta las dimensiones del contenido de cada entrada de la matriz. Al trasladar un destino mediante un vector se considera que el nuevo destino tiene dimensiones nulas. Si se desea conservar las dimensiones de la entrada en el cálculo de la flecha se puede usar `!` en lugar de `+`.

Existen varias formas de representar vectores. Por ejemplo, las letras U, D, R, L, UR, UL, DR, DL representan vectores que trasladan el destino del centro a su parte superior, inferior, etc. Por ejemplo `\xymatrix{ A \ar[r]+DL & B }` produce

$$A \longrightarrow B$$

2.1 Estilo de las flechas

Nos ocupamos ahora de la modificación del aspecto de una flecha. Una flecha consta de tres partes que podemos llamar cola, cuerpo y punta. El estilo de una flecha se determina mediante la sintaxis:

`@variante{cola cuerpo punta}`

Las colas y las puntas varían en un mismo juego de caracteres que llamaremos *extremos*. Son los siguientes:

<<	>>	(.....(
<<<<	>>>>))
//		< <
////		>>
oo	xx	++

Las posibilidades para el centro son

-	_____	--	- - - -	~	~~~~~
~	~ ~	:
=	=====				

Así, por ejemplo, la instrucción

produce

$$A \succ - \rightarrow B$$

Podemos omitir cualquiera de las tres partes (o dos de ellas). Si especificamos sólo un extremo se entiende que es la punta de la flecha y no la cola. Si ponemos `@{}` obtenemos una flecha invisible, cuya posible utilidad veremos después.

La *variante* puede ser:

^	(arriba),	-	(abajo),
2	(doble),	3	(triple).

Los ejemplos siguientes muestran su efecto:

`\xymatrix{A\ar@^{<->}[r]&B}` $A \longleftrightarrow B$
`\xymatrix{A\ar@_{<->}[r]&B}` $A \longleftrightarrow B$
`\xymatrix{A\ar@2{<->}[r]&B}` $A \longleftrightarrow B$
`\xymatrix{A\ar@3{<->}[r]&B}` $A \longleftrightarrow B$

Si queremos que la variante afecte sólo a una de las partes de la flecha podemos ponerla justo ante ella. Por ejemplo, la instrucción

`\xymatrix{ A\ar@{^(->)}[r] & B }`

produce

$$A \overset{\curvearrowright}{\longrightarrow} B$$

2.2 Flechas curvas

Para obtener flechas curvas podemos añadir `@/^/` si la queremos curvada hacia arriba o `@/_/` si la queremos hacia abajo. Por ejemplo,

`\xymatrix{ A\ar@{>--}@/^/[r] & B\ar@{>--}@/_/[l] }`

produce

$$A \overset{\curvearrowright}{\succ} B$$

Es importante recordar que todos los conceptos referentes a flechas son relativos a su sentido. Por ejemplo, “curvada hacia arriba” significa “curvada hacia arriba cuando se la mira de izquierda a derecha”. Así, en el ejemplo anterior ambas flechas están curvadas hacia arriba.

Si queremos modificar la curvatura podemos añadir una longitud. Por ejemplo,

`\xymatrix{A\ar@/^/[r] & B\ar@/^5mm/[l]}`

produce



Otra forma de curvar flechas es mediante

`@(salida, entrada)`

donde son direcciones de la forma `u`, `d`, `r`, `l`, `ur`, `dr`, `ul`, `dl`, que expresan en qué dirección ha de salir la flecha y en que dirección a de entrar en su destino. Por ejemplo,

`\xymatrix{A\ar@{ul}[d] \ar@{ur}[r] & B}`

produce

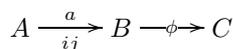


2.3 Etiquetas

Podemos situar texto encima, debajo o en medio de una flecha. Basta anteponerle un signo de superíndice, un signo de subíndice o bien `|` para superponer:

`\xymatrix{A\ar[r]^a_{ij} & B\ar[r]|\phi & C}`

produce



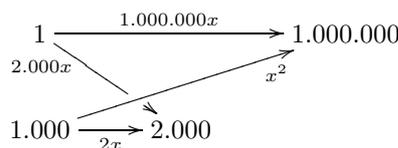
Las etiquetas se sitúan por defecto en el punto medio del segmento que une el centro de la entrada matricial de salida con el centro de la entrada matricial de llegada (que no es necesariamente el centro de la flecha). Entre el signo `^`, `_` o `|` y la etiqueta se puede poner una especificación de dónde debe situarse ésta. Las posibilidades son las siguientes:

- `-` sitúa la etiqueta en el medio de la flecha.
- `<` sitúa la etiqueta al principio de la flecha. Pueden ponerse varios seguidos `<<<`. Cuantos más haya, más se alejará la etiqueta del principio.
- `>` sitúa la etiqueta al final de la flecha. También pueden ponerse varios
- `(número)` sitúa la etiqueta en proporción al factor dado, de modo que 0 representa el origen y 1 corresponde al final. Por ejemplo, `\xymatrix{A\ar[r]^{(.6)\bullet} & B}` produce $A \xrightarrow{\bullet} B$

- `!{d1;d2}` donde d_1 y d_2 son posiciones en la matriz, pone la etiqueta en el punto donde una hipotética flecha que fuera de d_1 a d_2 cortaría a la flecha actual. Sólo funciona con flechas rectas. Entre otras cosas sirve para evitar que dos flechas se corten. Para ello intercalamos como etiqueta la orden `\hole`, que simplemente deja un espacio. Por ejemplo,

`\xymatrix{1\ar[rr]^-{1.000.000 x} \ar[dr]_{(.2){2.000 x}!{d};[rr]}\hole & & 1.000.000 \\ 1.000 \ar[r]_{2x} \ar[urr]_{>>>}{x^2} & & 2.000}`

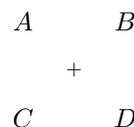
produce



Una flecha invisible puede ser útil para situar algo en una posición que no corresponda con ninguna entrada. Por ejemplo,

`\xymatrix{A\ar@{}[dr]|{+} & B \\\ C & D}`

produce



También podemos escribir en el destino de una flecha sin más que incluir el texto en `*{}` Por ejemplo,

`\xymatrix{A\ar[]+D;[r]+U*{o} & B\ar[]+D;[r]+U*{o} & C}`

produce

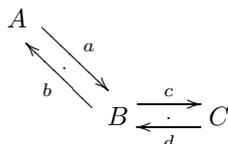


2.4 Flechas paralelas

Podemos desplazar una flecha paralelamente a sí misma mediante `@<>`, intercalando una distancia. Por ejemplo,

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

produce



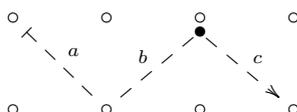
La distancia 1ex suele ser apropiada.

2.5 Flechas consecutivas

Es posible crear una flecha quebrada que visite varias entradas. Cada etapa intermedia (pero no la última) ha de ir precedida por un apóstrofo. Por ejemplo,

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

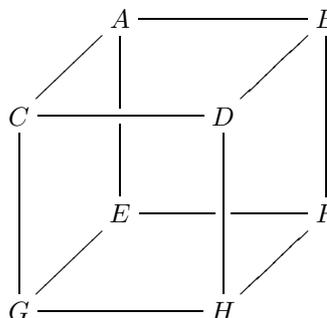
produce



Esto puede usarse a veces para conseguir flechas que se crucen sin cortarse de forma más sencilla, como es el caso de

```
\xymatrix{ & A \ar@{-}[rr] \ar@{-}' [d] [dd] \\
& & B \ar@{-}[dd] \\
C \ar@{-}[ur] \ar@{-}[rr] \ar@{-}[dd] \\
& & D \ar@{-}[ur] \ar@{-}[dd] \\
& E \ar@{-}' [r] [rr] & & F \\
G \ar@{-}[rr] \ar@{-}[ur] \\
& & & & H \ar@{-}[ur] }
```

que produce

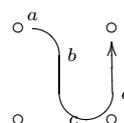


2.6 Flechas enlazadas

Las flechas de la sección anterior son segmentos que se interrumpen al llegar a cada destino parcial (a no ser que éste tenga dimensiones nulas). Ahora construiremos flechas que pasan por varios destinos intermedios pero que en lugar de cortarse se doblan adecuadamente. Las órdenes para lograr esto consisten en especificar en primer lugar en qué dirección ha de partir la flecha y hacia dónde ha de girar, después, en cada paso intermedio se indica hacia dónde ha de apuntar la flecha tras el nuevo giro y, por último se establece el destino definitivo. Consideremos como ejemplo la instrucción

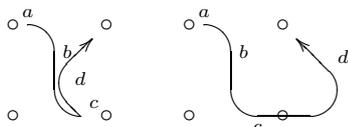
```
\xymatrix{{\circ} \ar' r [d]^a
' [dr]^b ' [r]_<c [r]_<d
& {\circ} \\\
{\circ} & {\circ} }
```

que produce



La orden `'r[d]` comienza con un acento grave, que indica un objetivo parcial. La `r` indica que la flecha ha de partir apuntando a la entrada de la derecha y la `[d]` que ha de girar hacia la entrada de abajo. El resultado es un giro de 90 grados hacia abajo. El siguiente tramo es `'[dr]`, que indica otro objetivo parcial, aunque ahora ya no es necesario especificar la dirección de salida, que se sobrentiende igual a `d` (pese a ello podríamos especificar otra, como veremos enseguida). La flecha gira ahora 90 grados hacia la entrada `[dr]`. Notemos que la flecha avanza primero en línea recta y luego gira. `\Xy-pic` determina si conviene avanzar primero

y girar después o viceversa para apuntar mejor al objetivo. En el paso anterior no se produjo ningún avance en línea recta porque no ayudaba en nada a apuntar hacia [d]. El siguiente paso hace que la flecha gire 90 grados hacia el objetivo [r] (sin necesidad de avande) y el último paso (sin acento grave) indica que dicha entrada es el objetivo final. Las dos figuras siguientes se han obtenido cambiando ‘[r]_<c por ‘ul[r]_<c y ‘^ul[r]_<c respectivamente.



En el primer caso hemos forzado a que el tramo c parta en dirección noroeste, lo que ha provocado un pico muy brusco en la flecha. En el segundo caso, el signo ^ fuerza a que el giro se produzca en sentido antihorario. Como Xy-pic no encuentra una solución posible, realiza primero un avance y luego gira para acabar en la dirección indicada. Otras órdenes imposibles pueden acabar con un mensaje de error. Podemos forzar giros en sentido horario mediante el signo de subíndice. Así mismo, podemos determinar el radio de giro mediante la sintaxis ‘/4pt[r]

3 Objetos

Un objeto es una porción de texto o de diagrama que Xy-pic puede tratar unitariamente para modificarlo o relacionarlo con otros objetos. La forma general de un objeto es

$$*modificadores\{texto\}$$

Ya nos apareció esta construcción en 2.3, al explicar cómo podemos especificar el texto del objetivo de una flecha. Basta adjuntarlo en forma de objeto, con la sintaxis anterior. También podemos usar objetos como etiquetas o en la especificación del estilo de una flecha.

Los modificadores más útiles son:

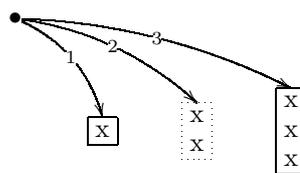
- =vector Establece el tamaño del objeto. Un vector tiene el formato <3mm, 5mm> Si no se especifica nada el tamaño del objeto estará determinado por el texto. El modificador =0 hace el objeto puntual.

- + aumenta un poco el tamaño del objeto (pueden ponerse varios seguidos). +<5mm> aumenta en la cantidad estipulada. Puede usarse - para disminuir el tamaño.
- += hace cuadrado el objeto, aumentando la menor de sus dimensiones, con un -= se disminuye la mayor de las dimensiones. Es posible combinar esto con los modificadores anteriores. Por ejemplo, +=+ hace cuadrado el objeto y lo aumenta un poco.
- [F], [F=], [F.], [F--], [F-], [F-:<2mm>], enmarca el objeto con un marco continuo, doble, a puntos, a rayas, sombreado o con esquinas redondeadas. Si se antepone [o] el marco es circular.

Mediante \txt{ } se pasa a modo texto (se sale del modo matemático). Por ejemplo,

```
\xymatrix{*{\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}}
```

produce



Mediante \txt<1cm>\{texto\} se especifica el ancho de la línea.

4 Órdenes globales

Para determinar el espaciado entre filas y columnas de una matriz contamos con las instrucciones siguientes, que se ponen justo después de \xymatrix y antes de la primera llave.

@=5mm	Espaciado general.
@R=5mm	Espacio entre filas.
@C=5mm	Espacio entre columnas.
@M=5mm	Margen por defecto de cada entrada.
@W=5mm	Anchura por defecto de cada entrada.
@H=5mm	Altura por defecto de cada entrada.
@L=5mm	Margen para las etiquetas.
@!	Todos los espacios iguales.
@!0	Idem y además ignorar el tamaño de las entradas.
@!R	Todos los espacios entre filas iguales.
@!C	Todos los espacios entre columnas iguales.
@1	Ajusta el espaciado para diagramas insertados en una línea de texto.

Por ejemplo, si en el diagrama cúbico de la sección 2.5 ponemos una entrada más larga que las demás se distorsionará el cubo, lo cual se evita con `@!0`

Si ponemos `@ur` el diagrama girará de modo que las líneas horizontales pasarán a tener sentido ur. En realidad sólo gira la red de entradas. Las entradas en sí mismas aparecen siempre horizontales.

Con `\objectstyle{\scriptstyle}` hacemos que todas las entradas tengan por defecto tamaño de subíndice. `\labelstyle{\scriptstyle}` hace lo mismo con las etiquetas.

`\entrymodifiers{++[o][F-]}` hace que todas las entradas sean circulares y aparezcan recuadradas.