Algunas cosas útiles sobre el TEX

(Sacadas de The Advanced TEX Book, de D. Salomon)

1 Categorías de caracteres

Cada carácter tiene asociada una categoría que determina su comportamiento. Hay un total de 16 categorías:

Categ.	Significado	Ejemplo	Categ.	Significado	Ejemplo
0	Carácter de escape	\	8	Subíndice	_
1	Inicio grupo	{	9	Ignorado	
2	Fin grupo	}	10	Espacio	␣
3	Modo matemático	$	11	Letra	a, b, ...
4	Tabulador	&	12	Otros	@
5	Fin de línea		13	Activo	
6	Parámetro	#	14	Comentario	%
7	Superíndice	^	15	Inválido	

Para cambiar la categoría de un carácter se usa `\catcode`. Por ejemplo,

```
\catcode98 = 9
Ejemplo bobo
\catcode98 = 11
```

produce 'Ejemplo oo', porque 98 es el código ASCII del caracter 'b'. En general no es necesario saberse el código de un carácter para cambiar su categoría. Una forma equivalente es `\catcode`\b=9`, aunque esto no sirve para reestablecer la categoría 11, porque la 'b' sería ignorada. (No sé qué pintan ahí exactamente el acento grave y la barra, pero funciona.)

Los comandos `\makeatletter` y `\makeatother` cambian la categoría de @ entre 11 y 12. El texto fuente LATEX asigna temporalmente a @ la categoría 11, lo que permite definir como comandos internos muchas palabras que contienen este signo. Después vuelve a asignarle la categoría 12, de manera que si el usuario intenta usar o modificar un comando que contenga @ obtiene un

error. Lo mismo sucede con la orden `\documentclass` (o `\documentstyle`), de modo que si copias alguna instrucción con @ del estilo book (por ejemplo) al preámbulo de un documento (para modificarla), no funcionará si no cambias temporalmente la categoría de @.

Los caracteres con categoría 13 funcionan como comandos aunque no empiecen por \. Por ejemplo, `\catcode`\ç = 13` permite definir a continuación `\defç{\c c}`, y así la ç puede entrarse desde el teclado como cualquier otra letra.

2 Clases de signos matemáticos

Los signos en modo matemático están divididos en ocho clases, con objeto de determinar adecuadamente la separación entre ellos. Las clases son

Clase.	Significado	Comando	Ejemplos
0	ordinario	\mathord	$\alpha\ \beta\ \aleph\ \forall$
1	operador	\mathop	$\sum\ \prod$ cos lím
2	operador	\mathbin	$+\ \circ\ \cup$
3	relación	\mathrel	$<\ \in$
4	izquierda	\mathopen	$(\ \{$
5	derecha	\mathclose	$)\ \}$
6	puntuación	\mathpunct	$,\ ;\ \cdots$
7	variable		$x\ y$

Así, por ejemplo, la diferencia entre spec$X$ y $\operatorname{spec} X$ es que el segundo está hecho con

```
$\mathop{\rm spec}X$
```

Más que en estos casos simples, el interés reside en los casos en que aparecen consecutivamente una relación y un operador, etc. Entonces las diferencias de espacios mejoran considerablemente el aspecto de la fórmula.

Los operadores tienen límites, es decir, los subíndices y superíndices los llevan abajo y arriba, en lugar de a la derecha. Por ejemplo

```
$$
\mathop{\mbox{ lím\,ind}}_{n\rightarrow\infty}X_n.
$$
```

produce

$$\lim_{n\to\infty}\operatorname{ind} X_n.$$

Si no queremos este efecto usamos `\nolimits`. Por ejemplo, la definición de sen es

```
\def\sen{\mathop{\rm sen}\nolimits}
```

Otro ejemplo: la diferencia entre $x \in]1, 2[$ y $x \in]1, 2[$ es que la segunda fórmula esta hecha con

```
$x \in \mathopen] 1, 2\mathclose[$
```

Algunos símbolos tienen distintos nombres según la clase que queramos que tengan. Por ejemplo, `|` es una barra ordinaria, mientras que `\mid` es el mismo símbolo como relación. Igualmente, ':' es relación, y `\colon` es puntuación.

Los comandos `\big, \Big, \bigg, \Bigg` aumentan gradualmente el tamaño de un delimitador. Si añadimos la letra 'l', 'r' o 'm' además el delimitador será tratado como izquierdo, derecho o relación. Por ejemplo

```
$$
\Bigl\{ (a, b) \Bigm| (a, b) < (c, d)\Bigr\}
$$
```

produce

$$\Big\{(a, b) \;\Big|\; (a, b) < (c, d)\Big\}$$

(`\big` no tiene efecto con la opción 12pt).

Comparar $||x| - |y|| \le |x - y|$ con $\Big||x| - |y|\Big| \le |x - y|$.

Un ejemplo en el que no interesa que un delimitador sea tratado como relación es

$$G(K/k)\Big/G\Big(K/(K \cap L)\Big),$$

escrito con

```
$$
G(K/k)\Big/G\Bigr(K/(K\cap L)\Bigl),
$$
```

3 Fantasmas

La instrucción `\phantom{texto}` deja el espacio que ocuparía el texto pero no escribe nada. Las instrucciones `\hphantom` y `\vphantom` sólo dejan el espacio horizontal y vertical respectivamente. Por ejemplo, para corregir el mal efecto de

$$\sqrt{a}\sqrt{d}$$

escribimos `\sqrt{a\vphantom{d}} \sqrt{d}` y resulta

$$\sqrt{a}\sqrt{d}$$

En general, `\mathstrut` deja el espacio vertical correspondiente a un paréntesis. Otro ejemplo, si queremos que en $A_2^2B_2$ los subíndices aparezcan alineados escribimos `$A_2^2B^{\phantom{2}}_2$` y queda $A_2^2B_2$.

Un uso de `\hphantom` es, por ejemplo, en una bibliorafía:

[1] Cervantes, M. *El ingenioso hidalgo Don Quijote de La Mancha.*
[2] *Las novelas ejemplares.*
[3] Quevedo, F. *Historia del buscón llamado Don Pablos.*

La instrucción `\smash` hace justo lo contrario: escribe con altura cero. Por ejemplo,

```
\font\grande=cmr10 at 30pt
un \hphantom{\grande T}ic, seguido de

un \smash{\grande T}ac.
```

produce

un Tic, seguido de
un ac.

Una aplicación más útil es, por ejemplo, evitar que esta línea se separe de ésta cuando explicas que el signo ⋮ se consigue con `\vdots`.

Si en lugar de `\smash{$\vdots$}` pusiéramos sólo `$\vdots$` quedaría así:

Una aplicación más útil es, por ejemplo, evitar que esta línea se separe de ésta cuando explicas que el signo ⋮ se consigue con `\vdots`.

4 Modos de escritura

En un momento dado, TEX puede encontrarse en uno de seis modos posibles de escritura:

Modo horizontal (hmode): cuando está escribiendo una línea.

Modo RH: cuando está escribiendo en una caja horizontal distinta de una línea.

Modo vertical (vmode): cuando está uniendo líneas (entre párrafos).

Modo vertical interno (IV): Cuando está escribiendo en una caja vertical distinta de una página.

Modo matemático: cuando está escribiendo una fórmula centrada (entre $$).

Modo matemático en línea: cuando está escribiendo una fórmula matemática insertada en el texto (entre $).

Los cambios a los modos matemáticos se producen sólo cuando el usuario pone dólares, y los cambios a modos internos (RH e IV) si se construyen cajas, pero los cambios entre hmode y vmode son automáticos y a veces hay que tenerlos en cuenta. El cambio a vmode se produce al acabar un párrafo, y el cambio a hmode se produce cuando se encuentra un carácter de texto. A veces es necesario forzar el cambio a hmode mediante la instrucción `\leavevmode`.

Por ejemplo, si modificamos el ejemplo anterior del Tic-Tac eliminando los dos 'un' hemos de escribir

```
\font\grande=cmr10 at 30pt
\leavevmode\hphantom{\grande T}ic, seguido de

\leavevmode\smash{\grande T}ac.
```

Si nos olvidamos de `\leavevmode` obtendremos

Tic, seguido de

ac.

La instrucción `\hphantom` es horizontal, y por lo tanto no tiene efecto en modo vertical (el espacio en blanco que vemos se debe al sangrado, no al `\hphantom`). La instrucción `\smash` sí tiene efecto, pero con el texto 'ac' se inicia una tercera línea.

Tendremos problemas de este tipo si empezamos una línea con una caja o similar.

5 Cajas

Todo texto ha de estar dentro de una caja. Las cajas pueden contener texto, espacios, instrucciones u otras cajas. Hay dos clases de cajas: horizontales (hbox) y verticales (vbox). Hay instrucciones que sólo hacen efecto en cajas horizontales, instrucciones que sólo causan efecto en cajas verticales e instrucciones con efecto distinto según el tipo de caja (algo importante a tener en cuenta). Las líneas de texto usuales son cajas horizontales, las páginas son cajas verticales compuestas de líneas.

Una caja tiene tres dimensiones anchura (width) altura (height) y profundidad (depth). La línea de altura 0 de una caja es su línea base (baseline).

La instrucción `\hbox{material}` crea una caja horizontal con el material dado. Su anchura será la del material y causará overfull si no cabe en una línea. En cambio `\vbox{material}` crea una caja vertical donde el texto contenido en el material se divide automáticamente en líneas (de anchura `\hsize`).

Por ejemplo, si escribimos

```
\hbox{\vbox{\hsize = 4cm \tolerance = 10000 Esto es
el texto de la primera columna. Los finales de línea
se fijan automáticamente}
\kern1cm
\vbox{\hsize = 5cm \tolerance = 10000 Esto es el texto
de la segunda columna.}}
```

obtenemos:

Esto es el texto de la primera columna. Los finales de línea se fijan automáticamente

Esto es el texto de la segunda columna.

La instrucción `\tolerance = 10000` sirve para aumentar la tolerancia en el espacio máximo entre palabras. Así se evita que aparezcan continuos overfulls debidos a la estrechez de las columnas. La instrucción `\kern` crea un espacio rígido, horizontal o vertical según si se usa en una caja horizontal o vertical.

La línea base de una caja vertical es la de su última línea. Por eso las columnas del ejemplo anterior están alineadas por abajo. Si cambiamos las dos órdenes `\hbox` por `\vtop` obtenemos cajas verticales cuya línea base es la de su primera línea, luego las columnas quedan alineadas por arriba.

La disposición relativa de las cajas se puede controlar con las instrucciones `\raise`, `\lower` (en cajas horizontales) y `\moveleft`, `\moveright` (en cajas verticales).

Por ejemplo,

```
A\raise2pt\hbox{B}\raise4pt\hbox{C}\raise2pt\hbox{D}E
```

produce A$^{\mathrm{B}}$C$_{\mathrm{D}}$E.

Las cajas se manipulan mejor con ayuda de registros. Existen 256 registros de cajas, llamados `\box0`, `\box1`, `\box2`, ...

`\setbox0=\hbox{A}` asigna al registro 0 la caja horizontal { A}.

`\copybox1` imprime la caja 1

`\box1` imprime la caja 1 y la borra (recomendado para ahorrar memoria)

`\ht0 = 10pt` `\dp0 = 12pt` `\wd0 = 20pt` modifican las dimensiones de la caja 0.

Como algunos registros pueden ser usadas por los macros del LaTeX, puede ser necesario asegurarse de estar usando un registro vacío. La instrucción `\newbox\micaja` asigna al comando `\micaja` el mínimo registro libre. Entonces todas las instrucciones anteriores valen igual cambiando el número 0, 1, etc. por `\micaja`.

Las instrucciones `\unhbox0` `\unvbox0` eliminan las cajas, es decir, si hacemos

```
\setbox0=\hbox{b}
\setbox1=\hbox{a\box1}
\setbox2=\hbox{a\unhbox1}
```

obtenemos en la caja 2 el contenido 'ab', y en la caja 1 la 'a' seguida de una caja con la 'b'.

Una precaución: si hacemos `\vbox{\unvbox1}` no se altera la longitud de las líneas de la caja 1 (no se reajustan a la `\hsize` en curso).

Podemos fijar la anchura de una caja horizontal mediante `to`, pero obtendremos mensajes de overfull/underfull si no incluimos un espacio flexible que haga que el material tenga realmente la anchura indicada.

Por ejemplo `\hbox to 5cm{A\hfil\hfil B \hfil C}` produce
A B C

El espacio `\hfil` se extiende lo necesario para cubrir la anchura prevista. Así el primer espacio resulta el doble de grande que el segundo.

El espacio `\hfill` anula a cualquier `\hfil`. Similarmente tenemos `\vfil` y `\vfill`.

Otro espacio interesante es `\hss`, que es como `\hfil` pero puede ser negativo. Por ejemplo, si escribimos

```
\hbox to0pt{x\hss}
```

obtenemos una caja de anchura 0, lo que en la práctica significa que escribimos la x sin desplazar el punto de inserción, con lo que cualquier cosa que escribamos a continuación se superpondrá a la x. Si escribimos `\hss` a la izquierda de la x, en lugar de a la derecha, el resultado es una x superpuesta al carácter anterior. Por ejemplo, `00\hbox to0pt{\hss x}` produce 0x0

6 Reglas

Las reglas son líneas horizontales o verticales. En realidad son rectángulos con altura, anchura y profundidad. La definición básica es

`\vrule height10pt width5pt depth0pt`,

que da lugar a .

Es importante recordar que ¡OJO! `\vrule` sólo se puede usar en cajas horizontales, mientras que `\hrule` sólo se puede usar en cajas verticales. Así, si en una línea se intercala una orden `\hrule` el párrafo termina en ese punto.

Si no se especifica alguna dimensión, ésta se fija según el contexto. Una regla horizontal tendrá por defecto altura y profundidad pequeñas y la anchura de la caja en la cual se encuentre.

Así pues, una simple orden `\hrule`, sin más, produce

---

porque la anchura de la caja en curso (la página) es `\hline`.

Por el contrario, una regla vertical tiene por defecto anchura pequeña y la altura y profundidad de la caja en curso.

La instrucción `\hrulefill` llena todo el espacio disponible con una recta horizontal (me parece que funciona tanto en modo horizontal como vertical).

Analicemos ahora la definición de ℂ:

```
\def\bbbc{\mathpalette{}{{\setbox0=\hbox{$\rm C$}\hbox{\hbox
to0pt{\kern0.4\wd0\vrule height0.9\ht0\hss}\box0}}}}
```

La orden `\mathpalette` sirve para que la definición funcione a distintos tamaños. Dejando eso de lado, primero se mete una C en la caja 0 y luego se construye una caja horizontal de anchura 0 que comienza por un espacio de 0.4 veces la anchura de la C, seguido de una regla vertical de altura 0.9 veces la de la C; luego `\hss` añade un espacio negativo para que la anchura de la caja coincida con el 0 previsto, y así el punto de inserción se queda donde al principio (como si no hubiéramos escrito la regla), luego se escribe la C, que se superpone a la regla.

7 Párrafos

Las magnitudes siguientes regulan el aspecto de los párrafos y, más en general, del contenido de cualquier caja vertical. Todas ellas pueden ser modificadas localmente en cada caja vertical.

`\hoffset`	margen izquierdo menos una pulgada.
`\size`	ancho de línea.
`\leftskip`	espacio adicional a principio de línea (0 por defecto)
`\rightskip`	espacio adicional a final de línea (0 por defecto)
`\parindent`	longitud del sangrado
`\parfillskip`	espacio de relleno en la última línea de un párrafo.
`\baselineskip`	distancia entre las líneas base de líneas consecutivas.
`\lineskiplimit`	mínima distancia admisible entre las líneas base de líneas consecutivas. Si no se respeta se añade espacio adicional usando `\lineskip`
`\lineskip`	espacio entre la parte inferior de una línea y la superior de la siguiente si no se respeta `\baselineskip`.
`\parskip`	espacio vertical entre dos párrafos.

La sintaxis para modificar estas magnitudes es `\baselineskip=.5cm`. Además se dispone de las instrucciones siguientes para hacerlo:

`\nointerlineskip` suprime el espacio adicional entre las línea anterior y posterior.

`\offinterlineskip` suprime el espacio adicional entre líneas en lo sucesivo.

`\noindent` suprime el sangrado en el párrafo siguiente.

`\hangafter = n` añade una sangría adicional (independiente de la que produce `\parindent`) a partir de la línea n del párrafo y cuya longitud se especifica mediante `\hangindent = 5pt`. Si n es negativo la sangría se aplica a las primeras líneas del párrafo. Si `\hangindent` es negativo la sangría se aplica a la derecha.

`\parshape =` $n\ i_1 l_1 i_2 l_2 \ldots i_n l_n$ produce un párrafo donde las n primeras líneas tienen sangría i_k y longitud l_k.

Estas instrucciones son útiles para acomodar figuras junto a párrafos y cosas similares. Por ejemplo:

En este espacio podría ir una figura, una inicial más grande, o lo que se quiera.

En un lugar de la mancha, de cuyo nombre no quiero acordarme, no ha mucho que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda ...

Este efecto está hecho con:

```
\setbox0=\vtop{\hsize=3.5cm \rightskip=6pt\tolerance=8000
\baselineskip=13pt {\footnotesize En este espacio
podría ir una figura, una inicial más gran\-de,
o lo que se quiera.}}

\setbox1=\vtop{\hangindent=\wd0 \hangafter = -4 \noindent
En un lugar de la mancha, de cuyo nombre no
quiero acordarme, ...}

\wd0=0pt
\hbox{\box0\box1}
```

Para acabar citamos `\abovedisplayskip` y `\belowdisplayskip`, que determinan el espacio por encima y por debajo de una fórmula centrada, así como `\displayindent` y `\displaywidth`, que determinan el sangrado y la anchura de una fórmula centrada, respectivamente. Por defecto son 0 y `\hsize`.

8 Condicionales

A la hora de modificar estilos resultan útiles los condicionales. La instrucción básica es `\newif\ifMiCondicion`. Esta instrucción crea una variable booleana `\MiCondicion`, cuyo valor se puede establecer mediante los comandos `\MiCondiciontrue` y `\MiCondicionfalse` y se puede consultar mediante la sintaxis

```
\ifMiCondicion
   instrucciones
\else
   instrucciones
\fi
```

Otros condicionales útiles son
`\ifnum contador1 < contador2` (también vale con > o con =)
`\ifdim longitud1 < longitud2` (idem)
`\ifodd contador`
`\ifvoid5` (determina si `\box5` está vacía.)
`\ifhbox5`, `\ifvbox5`

```
\ifcase contador
   instrucciones para contador = 0
\or
   instrucciones para contador = 1
\or
   instrucciones para contador = 2
\else
   intrucciones para cualquier otro valor
\fi
```