TeXemplares

Boletín de CervanT_EX, Grupo de Usuarios de T_EX Hispanohablantes

Año 3.º, número 3, tercer trimestre 2002

T_FXemplares

TEXemplares es el boletín de CervanTEX, el grupo de usuarios de TEX hispanohablantes. Queda prohibida cualquier reproducción total o parcial por cualquier medio, sea convencional o electrónico, de su contenido. Ni TEXemplares ni CervanTEX comparten necesariamente la opinión de los colaboradores. Nada en TEXemplares debe entenderse como una postura oficial del grupo.

Las colaboraciones deben ser creadas con la clase TeXemplar.cls, disponible por CTAN en una versión preliminar. Se debe usar, siempre que sea posible, caracteres de 7 bit y acentuar con el método de abreviaciones de babel. Las figuras externas en eps se deben crear a 600 pt.

Este ejemplar se creo con las siguientes aplicaciones: pdfTeX, Version 3. 14159, LaTeX2e <1999/12/01> y teTeX0.9 (aunque actualizado en algunos puntos). Se imprimió a 600 pt. con una ampliación de 1,414 y luego reducido a su tamaño real en las copias.

Notación. Por necesidades de composición y contrariamente al comportamiento normal de \verb, el código puede ser dividido a final de línea. Las divisiones en un símbolo no se indican mientras que aquellas entre letras se hace con un guión estilístico. Eso quiere decir que un guión a final de línea nunca es parte del código mientras que los que haya al comienzo sí lo son.

Redacción. Javier Bezos, Pascual Lucas, Enrique Meléndez, Luis Seidel.

Depósito Legal y otros registros en trámite.

Editorial

Estimado lector,

Este tercer número, como se anunció, corresponde a la segunda parte de la edición de las ponencias presentadas en el primer encuentro del grupo de usuarios de TEX hispanohablantes, el EGUTH'99 que se celebró en Madrid los días 13 y 14 de septiembre de 1999. Este número ha sido posible gracias a la labor de los editores, Javier Bezos, Pascual Lucas, Luis Seidel y Enrique Meléndez, quienes han dedicado mucho tiempo a que la calidad en su edición sea la adecuada.

El agradecimiento va también a los autores, que han aportado lo esencial del número, sus contribuciones. Se quiere animar desde aquí a todos los usuarios de TEX y amigos de TEXemplares a mantener vivo este boletín remitiendo contribuciones para próximos números.

Tipografía española con T_EX

Javier Bezos

1. Introducción

Si consultamos el Diccionario de la Real Academia y saltamos de una definición a otra, podemos concluir que la tipografía es el arte de confeccionar una obra impresa; una definición bastante genérica que no revela todas las ramicaciones que comporta. De forma más precisa, la tipografía no sólo incluye el diseño de letras y la maquetación de libros, que son los temas que tratan casi exclusivamente los libros más recientes, sino el conjunto de normas que se aplican para usar de forma adecuada las letras y signos ortográficos del alfabeto latino o cualquier otro (*ortografía*), las abreviaciones y símbolos de diversas disciplinas científicas (*ortografía técnica*) y la estructuración de elementos en cuanto no forman parte directa del texto, como llamadas de nota, bibiografías, disposición de títulos, etc. (*ortotipografía*).¹

La forma escrita de cada lengua se organiza a partir de una serie de signos que le son propios y característicos. Naturalmente, las lenguas que se basan en el alfabeto latino, además de los signos propios, comparten una serie de elementos comunes —todas tienen la letra a, por ejemplo—aunque cada una de esas lenguas le ha atribuido un sentido distinto. Así, el inglés posee unos signos como & o – que no forman parte del repertorio del español; lo mismo se puede decir de β y " en alemán o la «thorn» en islandés. Por otra parte, la interrogación y la exclamación de abrir son casi

¹Definiciones que, en todo caso, no deben tomarse demasiado literalmente.

exclusivas del español, mientras que la \tilde{n} es muy rara.²

Uno de nuestros objetivos es definir el uso de estos signos en el español. Con ello no pretendemos más que establecer unas pautas que nunca pueden ser rígidas ni implicar la renuncia al enriquecimiento con otras ideas que pueden ser útiles. Sin embargo, eso no se ha de hacer simplemente porque se ha visto en otra lengua, particularmente el inglés, sino teniendo en cuenta la lógica interna de la nuestra.

Por poner un ejemplo, cada día es más frencuente ver & con el sentido de *y*: ¿hay alguna razón para ello que no sea la de haberlo visto en inglés? Ahí puede ser una abreviatura útil de *and*, pero en español *y* ¡difícilmente puede ser más breve! ³ Se ha usado en ocasiones, tal vez simplemente porque estaba ahí, pero no ha tenido auténtica tradición en nuestra lengua. ⁴

Un buen ejemplo de un posible enriquecimiento lo tenemos en la distinción entre el signo menos - (\$-\$) y la raya - (---) que expondremos más adelante. El objetivo claro, en todo caso, debe ser producir un resultado lo más inteligible posible, evitando notaciones o formatos que a nosotros nos pueden parecer muy ocurrentes e ingeniosos pero que tal vez no sean evidentes a quien los lea.

Hay que resaltar que las normas tipográficas no sólo obedecen a una lógica y una tradición, sino también a la conveniencia de los tipógrafos en una época en la que era complicado realizar cierto tipo de correcciones. Un ejemplo muy característico es la prohibición radical de numerar las notas por páginas, ya que un cambio en la compaginación obligaba a retocar las planchas de imprimir; hoy tal precaución no es necesaria, pero todavía aparece esta norma en los libros de tipografía. Otro buen ejemplo es el sistema de citas conocido como *autor-fecha*, que tambien es independiente de la compaginación final.⁵ Por supuesto, con TEX también aparecen conveniencias, sobre todo cuando se aceptan sin más unos estilos que fueron concebidos según la tipografía del inglés americano (ni siquiera se adaptan bien al inglés británico).

También parece que algunos profesores de computación piden que en sus libros se sustituya la coma decimal por el punto de los miles, ya que en inglés su significado está invertido. La razón es que en muchos lenguajes

²Las primeras se han incorporado al catalán. En cuanto a la \tilde{n} , también se usa en el friulano, otras lenguas próximas de Italia y Suiza, y el bretón.

³En realidad, & es más ancho que y por lo que es más bien una antiabreviatura.

⁴Sobre las letras usadas para significar *y*, véase la colaboración de Menéndez Pidal en el libro de Morato, p. 85.

⁵Léase el comentario al respecto de Oren Patashnik en el manual de BIBTEX, que da de lleno en la diana.

de programación se usa el punto con sentido decimal (en TEX se pueden usar indistintamente). Creo que esto es medir todo exclusivamente según la conveniencia de una asignatura —despreciando lo que se sigue en todas las demás y en cualquier escrito de la naturaleza que sea—, inducir a la confusión a los alumnos por introducir un criterio contrario al generalmente aceptado, y ocultar la naturaleza intrínsicamente arbitraria de la sintaxis de los programas, cuyas normas se basan en unas necesidades distintivas. Según este razonamiento, también debería usarse == en lugar de = en las fórmulas.

Nos centraremos en la ortotipografía, ya que sobre ortografía hay bastantes tratados en librerías y bibliotecas mientras que de ortotipografía las referencias son muy escasas. Por fuerza tenemos que dejar fuera algunos aspectos más especializados, como la composición de poemas, obras de teatro, números, cuadros o diccionarios. Tampoco nos detendremos demasiado en las cursivas o versalitas, porque casi todo se reduce a enumerar los casos en que se aplican.

Debido a la importancia que tiene la literatura científica inglesa, también destacaremos algunas diferencias significativas con ella, en especial en aquellas normas que han sido adoptadas por La Y que por tanto requieren introducir modificaciones. Varios de los ejemplos aprovechan el nuevo estilo spanish de babel, que incluye muchas utilidades para poder seguir las normas tipográficas españolas.

2. Estudio de algunos signos

Dejaremos de lado las normas para el uso en el propio texto de signos como el punto, la coma o las mayúsculas, ya que se trata más de un problema de estilo ortográfico o literario que tipográfico, y por tanto me remito a los libros sobre gramática y redacción actualmente disponibles. Tampoco consideraremos signos de uso muy esporádico como el apóstrofo,⁶ el calderón, la manecilla, el cristus, etc.

Guión, menos, raya y semirraya

A menudo todos ellos se engloban bajo la denominación de *guiones* y se emplean los términos de *guión largo* para la raya y *guión corto* para el guión, pero se trata de signos distintos, tanto en función como en diseño: el guión, aparte de ser más corto, suele ser más de trazo más grueso que la raya.

⁶Y no el *apóstrofe*, que es una figura retórica.

También conocido por los tipógrafos como *división*, el guión sirve para dividir una palabra a fin de línea y para indicar números correlativos: 2-6 (2-6). En ese último caso, conviene no separar la expresión; para ello spanish proporciona la abreviación ~- por lo que mejor escribimos 2~-6.

La raya se emplea principalmente para incisos —como este— y diálogos. También sirve como sustituto de alguna cosa, como un dato en un cuadro, una palabra en una serie de entradas en un índice alfabético o un autor en una bibliografía. Nótese que digo uno, ya que el resto de los autores sí se escriben:

Luis Sánchez y Juan García: Fuentes y Fontanería, Madrid, 1968.

—y Juan García: Tipogrifería, Bogotá, 1987.

Que yo sepa, no existe ningún estilo de BIBTEX que siga esta norma.

Como T_EX se pensó sobre todo para el inglés, nos encontramos con que puede haber división tras una raya; la abreviación ~--- de spanish impide esa división posterior. En inglés hay incluso rayas de dos y tres cuadratines para situaciones muy concretas.

Los términos *menos* y *raya* se han utilizado siempre como sinónimos aunque en ámbitos distintos: *menos* por los tipógrafos, *raya* por los lingüistas. Tal equivalencia no es conveniente, ya que son dos signos completamente diferentes sujetos a normas distintas; por ejemplo, el menos ha de ser siempre del mismo ancho que los signos más e igual del tipo usado para matemáticas mientras que la raya tiene una medida y una forma que depende del tipo en el texto.

En inglés y otros idiomas se usa también una raya algo más corta llamada *en-dash* y que nosotros llamaremos *semirraya* a falta de un nombre normalizado. Nunca parece haber hecho falta, aunque por influencia del inglés se está extendiendo como sustituto del guión en números correlativos o de la raya en apartados, sin que exista razón alguna; aquí el problema es que algunos estilos de BIBTEX usan por omisión esta semirraya.

Comillas

Las españolas (también llamadas latinas) son estas: «, ». Eso no quiere decir que las inglesas (", ") no se usen: su función es la de comillas dentro de comillas.

Puntos suspensivos

Tanto por su forma como por su uso, hay diferencias notables con el inglés. En español los puntos (que siempre son tres) no están espaciados y

sirven para dejar algo en suspenso y en ocasiones como sustituto de etc.:

```
él dijo... Bueno, déjalo. él dijo... Bueno, déjalo.
```

Hay que distinguir si los puntos suspensivos cierran la oración o no, ya que en ocasiones TEX puede aprovecharse de ello. La opción spanish para babel dintingue entre . . . (con los tres puntos escritos tal cual) cuando cierra una oración y \ . . . cuando no, como en:

```
¿Qué c|.. era eso?... ?'Qué c\... era eso?...
```

Nótese que \... no descarta los espacios que le siguen.

Para indicar la supresión de parte de una trascripción hay que escribir [...], que Martínez de Sousa llama *puntos encorchetados* y, con mucha lógica, considera un signo ortográfico aparte. En definitiva, \dots (que espacia los puntos) carece de uso en castellano.

Reunión de signos

Un problema interesante se plantea cuando se reúnen varios signos, y en algunos casos hay disparidad de opiniones. A menudo es un problema más tipográfico que ortográfico y su razón de ser es, casi siempre, la de la uniformidad. Bien mirado, ni siquiera es algo tan importante como a veces se ha sugerido, pero eso no excluye que una tipografía cuidada establezca unos criterios.

En este aspecto, el inglés (al menos el americano) tiene una norma muy simple: los signos volados siguen a los signos «bajos» como la coma, el punto, etc. Un criterio sencillo y de facil aplicación, como en "that's all, folks." Precisamente, esta es la regla que hace que las comillas sean «transparente al *spacefactor* responsable de aumentar en TEX el espacio tras el punto o la coma.

En cuanto al castellano, nos detendremos únicamente en dos casos. El primero es muy simple: si tras cerrar interrogación o exclamación tuviera que ir punto, éste se suprime.

El segundo caso se refiere a la concidencia de comillas de cerrar con algún otro signo. La regla tradicional es la siguiente, formulada para el caso concreto de una cita textual:

 En general, se omiten los signos de puntuación que pudieran quedar al final de la cita, salvo si hay que cerrar un signo que se ha abierto (paréntesis, interrogación, etc.). Ese signo de cerrar tan solo cumple la función de cierre. 2. Como excepción, si la cita está al comienzo de párrafo o tras punto y, además, tras ella tuviera que ir un punto, se terminará la cita con ese punto, al tiempo que se reemplaza cualquier signo de puntuación que pudiera haber y se aplican otras reglas que pudieran ser relevantes. Tras las comillas de cerrar no se añade nada.

Realmente, la excepción no solo introduce una complicación innecesaria, sino que además supone una modificación no indicada del original (lo que debe evitarse siempre en citas textuales). Por tanto, suprímase sin más y colóquese el punto siempre tras las comillas.

3. Partes del texto

Principios y finales

Se llaman así las partes que van al principio y al final del escrito, respectivamente, como índices, prólogos, derechos, portada, dedicatorias, etc. Las posibles disposiciones de estas partes son muchas, y sólo la observación y el buen gusto pueden proporcionar las pautas para crear nuestros propios estilos. En todo caso, normalmente debemos dar por buenos los formatos del LATEX estándar a menos que exista una necesidad real de introducir algún cambio.

En lo que respecta al *índice*, los paquetes tocloft, para pequeños y sencillos reajustes en el formato estándar, y titletoc, para diseñar un nuevo esquema con total libertad a partir de una idea previa, pueden ser de gran ayuda. Solía ir en los finales, aunque hoy en obras técnicas se prefiere en los principios, antes de los prólogos (porque si se hace después de ellos, como se ve a menudo en obras anglosajonas, pierde su utilidad). La costumbre de escribir «Capítulo» o el signo de párrafo (§) ha caído en desuso ya que no aportan más claridad a un índice bien diseñado. Por lo demás, no hay mucha diferencia con relación al inglés, aunque se prefiere añadir un punto tras el número de sección.

El *índice alfabético* es más problemático, ya que *MakeIndex* no admite más ordenaciones que las del inglés y el alemán, a menos que recurramos casi continuamente al @ para dar una alfabetización distinta. Para solucionarlo, he escrito un pequeño paquete que hace ese trabajo en la trastienda y cuya idea es, en principio, muy simple: que cuando escribamos la orden \esindex{cañón} se convierta en \index{can^ffon@cañón}. Su realización resultó no ser tan simple y, aunque ahora funciona bastante bien, no

estoy por completo satisfecho, por lo que es probable que en un futuro intente otra aproximación.⁷ La otra posibilidad es xindy, pero es algo complicado de usar y todavía faltan implementaciones para varias plataformas. Con todo, ninguno de esos sistemas permite establecer diferentes criterios para la alfabetización de partículas.

Los *glosarios* no son muy frecuentes, y hasta hace poco todavía estaba por llegar un sistema mínimamente adecuado. Sin embargo, ahora hay un paquete llamado gloss que apunta en la dirección correcta: usar BIBTEX, que es una utilidad disponible en todos los sistemas, con una variante para 8 bits, programable y que permite tener bases de datos externas. Este es, sin duda, mi sistema preferido.

En cuanto a las *bibliografías*, de nuevo ha de pensarse en un formato que sea más claro que ingenioso. En ocasiones, el número de edición se indica con un número volado, pero quien se encuentre por primera vez con esa disposición se preguntará qué significa esa cifra. Personalmente, prefiero la fórmula tradicional con el número de edición seguido de *ed.*, como en «2.ª ed. rev. y aum.». Las bibliografías pueden tener una gigantesca variedad de disposiciones, sobre todo en las obras cuya finalidad principal es la documentación.

Capítulos y secciones

Probablemente, este sea uno de los aspectos en el que menos diferencias hay entre las diferentes tradiciones tipográficas nacionales, al menos en la actualidad. Pero podemos señalar dos características del español: 1.º, tras el número de la parte sigue un punto, y 2.º, el primer párrafo tras el título no se compone de forma especial. Esto último implica que, a menos que se use el nuevo spanish, conviene cargar el paquete indentfirst.

4. Otros elementos

Aunque sería interesante tratar los *cuadros* o los *folios* (la linea que puede haber al pie o a la cabeza de la página para indicar su número o alguna otra información que se considere necesaria), no hay mucho que decir específico para el español.

⁷CTAN:macros/latex/contrib/supported/bezos/esindex.*

Notas

Las llamadas en el texto se han de hacer de forma que no interrumpan significativamente la lectura en mitad de una oración; un buen sitio es detrás de una coma o un punto. Normalmente se dice que las llamadas han de ser muy discretas, para no dar la sensación de fragmentación en el texto, pero también conviene recordar que hay que facilitar el retorno al punto donde estábamos leyendo y un signo demasiado discreto puede no ser suficiente. Algunos de los sistemas usados son:

(1) 1 $^{(1)}$ * (*) a $^{(a)}$

En inglés es muy frecuente designar las notas con una sucesión de símbolos como *, †, ‡, ¶, etc. Personalmente, nunca he sido capaz de recordar su orden —suponiendo que haya uno normalizado—, sobre todo cuando no van en el texto sino en cuadros, donde el orden no es evidente. Lo mejor es usar números volados y, si es necesaria otra serie de notas (en cuadros, por ejemplo), letras minúsculas de cursiva; sólo si se sabe que van a haber pocas puede resultar interesante emplear uno, dos, tres asteriscos tal y como ofrece spanish, comenzando de nuevo en cada página con la ayuda de footmisc.

Apartados

Este es uno de los puntos donde las clases básicas de LATEX son completamente inapropiadas. Y lo son por dos razones:

- a) Se emplean las minúsculas para formar numeros romanos; en inglés «lix» es cincuenta y nueve, pero en castellano las minúsculas siempre se «leen», es decir, que tenemos 'lics' (cualquiera que pueda ser su significado). Nótese que «LIX» es quincuagésimo noveno, con sentido ordinal. Naturalmente, «LIX» puede resultar muy ostentoso en muchos contextos, lo que se puede solucionar con versalitas, como «LIX»; para ello se puede usar en el nuevo spanish la orden \lsc{LIX}.
- *b*) En «(a)» hay tanto paréntesis de abrir como de cerrar, con la letra de redondo. La forma correcta es la que estamos usando aquí, con la letra de cursiva y un paréntesis de cerrar de redondo.

No hay una norma rígida para identificar los apartados, aunque son frecuentes las siguientes variantes:

a) 1. 1.° 1) A. A) I. I)

Las más usadas son las tres primeras. El punto se suele emplear cuando se encabezan varios párrafos o uno de cierta extensión; el paréntesis, en apartados cortos o para subdivisiones en apartados «con punto». A un ordinal no le sigue ningún signo (ya incluye un punto), aunque en medio de un párrafo le sigue una coma.

5. División de palabras

Este es un punto que por sí mismo requiere un examen mas detallado del que podemos dar aquí. El problema radica en la falta de un criterio unitario, aunque se pueden distinguir dos modalidades:

- a) la silábica, en la que siempre se divide por sílabas de forma que nunca se divida antes de una vocal o h ni quede suelta una letra de la palabra, y
- *b*) la llamada *etimológica* (no muy correctamente), en la que las palabras formadas de prefijo + raiz —y cuyo significado sea ese— se dividen según el método silábico como si fueran palabras sueltas.

Se considera que hay una frontera silábica antes de una consonante o, si hay varias agrupadas, antes de la última de ellas, excepto con *bl*, *br*, *cl*, *cr*, *fl*, *fr*, *gl*, *gr*, *pl*, *pr*, *tr*, *ch*, *ll* y *rr* que nunca se dividen.

Vamos con algunos ejemplos de ambos métodos, con las posibles divisiones indicadas con |:

Silábica	Etimológica	Silábica	Etimológica
1) de sa pa re cer	des apa re cer	8) pa ra psi có lo go	pa ra psi có lo go
2) clor hi dra to	clor hi dra to	9) ist mo	ist mo
3) con traa ta que	con tra ata que	10) cor taú ñas	cor ta úñas
4) in te rre la cio nar	in ter re la cio nar	11) su bri ga da	su bri ga da
5) cohe cho	co he cho	12) su bra yar	sub ra yar
6) ex tra rra dio	ex tra rra dio	13) cohor te	cohor te
7) reo ler	re oler	14) ciu dad rea le ño	ciu dad rea le ño

La división etimológica produce divisiones más naturales (1, 3 y 4) —lo que es importante en irregularidades prosódicas (12 y 14) y para que se vea el sentido de palabras formadas *ad hoc* (7)—, no obliga a introducir excepciones (2, para consonante + h; 8, para el grupo ps que silábicamante es p|s, y 14, por razones evidentes), evita situaciones antiestéticas (la aa en 3), y facilita algunas divisiones que no tendrían que tenerse por incorrectas

(cohe en 5 es una sílaba y no se puede dividir, pero en la etimológica es un punto válido). En todo caso, siempre hay que evitar dividir palabras como cohorte ya que su significado no se deriva de co + horte.

Sólo los patrones de Julio Sánchez dan una división silábica correcta, y no hay ninguna serie de patrones preparados para la división etimológica. El problema es hacer una lista de prefijos, ya que cualquier verbo en la tercera persona singular del presente puede ejercer ese oficio: alguien que destroza armarios es un *destrozaarmarios*, algo que abre cartas es un *abrecartas*, etc.

Los ejemplos 11 y 12 presentan algunos problemas a los que la Real Academia no parece dar una solución satistactoria: de las normas de ortografía se puede deducir que la *b* nunca se debe pronunciar larga (como se alarga la *n* en *innoble*), aunque de hecho todos los prefijos se suelen pronunciar como si se escribieran separados (como *semieje*) e incluso uno, *ex*, no se une. Eso es particularmente cierto en *subrayar*, que debería escribirse *subrrayar*. En cuanto a *subrigada*, la Real Academia sostiene que el prefijo en este caso es *su*- y no *sub*-.

Un último problema que apuntaremos es el de las divisiones que se producen en el punto donde será necesario algún cambio ortográfico; su solución depende de si consideramos que los elementos que inicialmente estaban divididos, se unen y se vuelven a dividir, o si directamente permanecen sin unir. En el primer caso se deben respetar los posibles cambios y así lo hemos hecho en los ejemplos; en el segundo, se preserva la grafía original de los elementos de forma que tenemos *extra-radio* (6), *corta-uñas* (10) y, si se considera que el prefijo en realidad es *sub-*, *sub-brigada* (11). Pero ya nos desviamos mucho del tema y por tanto me remito a la literatura especializada.

6. Lomos

Hoy los lomos se suelen componer de forma vertical, y sólo si el libro es grueso y el título corto aparecen en posición horizontal. La disposición «a la inglesa» es de arriba abajo para que se pueda leer bien si el libro está sobre una mesa; la disposición «a la española» es de abajo arriba, para que se lea bien si el libro está boca abajo y por tanto no tenemos la cubierta a la vista. También se dice que el lomo a la española es más facil de leer en una estantería, aunque personalmente no encuentro diferencia entre inclinar la cabeza a un lado u otro.

7. Unidades de medida tipográficas

Este es un punto interesante y que rara vez se trata. Por razones históricas y prácticas, en tipografía no se sigue el sistema internacional de unidades sino uno propio, mejor adaptado a las medidas con las que se trabaja. Ese sistema variaba de una pais a otro aunque, incluso antes de la generalización de las computadoras, se redujo a dos sistemas: ⁸

- a) el Didot, en el que se llama cícero a los 12 puntos, es el adoptado en la Europa continental y buena parte del mundo, y
- b) el inglés, en el que se llama pica a los 12 puntos, ha sido adoptado en Inglaterra, parte de Europa y la mayoría de paises americanos. Su punto es ligeramente menor que el Didot.

TEX admite ambos sistemas, aunque el inglés está rectificado de forma inapreciable para adaptarse mejor a los sistemas de computación. No es casual que incluso en países con tanto apego a la tradición como Francia se emplea el inglés porque la diferencias son mínimas y las ventajas grandes. Pero no cabe duda de que la inclusión del cícero entre las unidades de TEX es toda una señal de respeto de Knuth a otras tradiciones tipográficas.

8. Modificar babel

El sistema babel proporciona un conjunto de utilidades para facilitar la escritura en diferentes lenguas, aunque su desarrollo es lento y no siempre muy acertado. La versión en el momento de escribir esto es la 3.6x con un «bicho» importante que impide desarrollar un tema que inicialmente tenía previsto: los signos diacríticos en la escritura del sefardí.

En lugar de ello, hablaremos del bable en el que se acostumbraba a escribir \ddot{x} para un sonido similar a la sh inglesa. Ahora solamente apuntaremos que existe tal posibilidad, y mostramos el código que tendría que añadirse al archivo spanish.cfg:

\declare@shorthand{spanish}{"x}{\"x}

⁸A los que cabría añadir el sistema italiano y algún otro.

9. Bibliografía

La referencia más útil en general para la presentación de trabajos es el Diccionario de ortografía técnica, de Martínez de Sousa. Mucho más limitado, pero también excelente, es el libro de Ramón Sol.

- Carlos Arroyo y Francisco José Garrido, *Libro de estilo universitario*, Madrid, Acento, 1997.
- Javier Bezos, Estilo spanish para el sistema babel, documento electrónico.
- Johannes Braams, Babel, a multilingual package for use with LaTeX's standard document classes, documento electrónico.
- The Chicago Manual of Style, Chicago, University of Chicago Press, 14.^a ed., 1993, esp. págs. 333-335.
- Prudenci Comes, Guía para la redacción y presentación de trabajos científicos, informes técnicos y tesinas, Barcelona, Oikos-Tau, 1971.
- Umberto Eco, Cómo se hace una tesis, Barcelona, Gedisa, 1982.
- José Fernández Castillo, *Normas para correctores y compositores tipógrafos*, Madrid, Espasa-Calpe, 1959.
- Michel Gossens, Frank Mittelbach y Alexander Samarin, *The LaTeX Companion*, Reading, Addison Wesley, ed. rev., 1994.
- Iacob. M. Hassan, «Transcripción normalizada de textos judeoespañoles», *Estudios Sefardíes*, n.º 1, 1978, págs. 147-150.
- Javier Lasso de la Vega, *Cómo se hace una tesis doctoral*, Madrid, Fundación Universitaria Española, 1977.
- Vicente Martínez Sicluna, *Teoría y práctica de la tipografía*, Barcelona, Gustavo Gili, 1945.
- José Martínez de Sousa, *Diccionario de ortografía de la lengua española*, Madrid, Paraninfo, 1996.
- Diccionario de ortografía técnica, Madrid, Germán Sánchez Ruipérez/Pirámide, 1987. (Biblioteca del libro.)
- Diccionario de tipografía y del libro, Madrid, Paraninfo, 3.ª ed., 1992.
- Manual de edición y autoedición, Madrid, Pirámide, 1994.
- Juan José Morato, *Guía práctica del compositor tipográfico*, Madrid, Hernando, 2.ª ed., 1908 (1.ª ed., 1900, 3.ª ed., 1933).
- Marion Neubauer, «Feinheiten bei wissenschaftlichen Publikationen», *Die T_EXnisches Komödie*, parte I, vol. 8, n.° 4, 1996, págs. 23-40; parte II, vol. 9, n.° 1, 1997, págs. 25-44.
- José Polo, Ortografía y ciencia del lenguaje, Madrid, Paraninfo, 1974.
- Manuel Seco, Diccionario de dudas y dificultades de la lengua española, Madrid, Espasa-

Calpe, 9.a ed., 1986.

— *Gramática esencial del español*, Madrid, Espasa-Calpe, 3.ª ed., 1995 (4.ª ed., 1996) Ramón Sol, *Manual práctico de estilo*, Barcelona, Urano, 1992.

Real Academia Española, *Esbozo de una nueva gramática de la lengua española*, Madrid, Espasa-Calpe, 1973.

Pedro Valle, Cómo corregir sin ofender, Buenos Aires, Lumen, 1998.

«Webster's Style Manual», Webster's New Encyclopedic Dictionary, Colonia, Könemann, 1994, págs. 1323-1395.

Typesetting Books with Special Needs

Yannis Haralambous
Atelier Fluxus Virus
http://www.fluxus-virus.com

What do we call a *book with special needs*? It is a book whose preparation demands special efforts by the author (or typist), either because its structure is complex (dictionaries and critical editions are typical examples of books with complex structure), or because it contains information which must be presented in a special way (mathematics, phonetics, multiple languages and scripts) or simply because it has to obey to a certain number of constraints which go father than the standard technical document (whose needs are almost entirely handled by standard LATEX).

We will cover a few cases, taken from real world challenges we had to face in our typesetting work. Our goal is to achieve the typographical quality of traditional typography, while using the most appropriate IT methods and standards.

1. Visual Quality

It may sound strange (or at least pessimistic) to start a list of "special needs" by "visual quality." After all, every book should have visual quality, this is the least to ask. Actually this is not always true: in real life many publishers are rather inclined to produce books in the cheapest possible way, and this leaves little room for any esthetic consideration.

Nevertheless there are cases where the book needs to comply to a certain visual quality standard. But how do we transform a standard LATEX document into a typographical masterpiece? Maybe we could start with

something more realistic: how do we obtain a decently typeset book? Here is a tentative checklist of things to consider:

- 1. choose the right font and page layout (according to book format and paper nature);
- 2. apply basic typesetting rules (for the specific language and country);
- 3. apply principles of good typography with a maximum of good taste;
- 4. print the result and show it to a specialist.

Making the Right Choices

Fortunately LATEX allows an easy change of typographical and page layout parameters. But still we need a starting point: we have to choose the font, its size and leading, the dimensions of the printed page, the vertical space between paragraphs, text blocks, headers, footers and main text, the presentation of titles and subtitles, footnotes, captions, etc. In LATEX occurric terms: everything that makes a class file (and often much more).

The choice of the font is a typical example: it has to do with the contents of the book: is it a modern text? an old text? in that case the font should not be significantly more recent that the text: you don't typeset Montaigne in Helvetica, nor Cervantes in Palatino; is it literary or technical? in the latter case, you should choose a neutral font (see § 1); in what extent should typographical choices be apparent? in a "luxury edition" you have more freedom because people expect to find a "nice" font, while a critical edition concentrates on contents and again the global appearance must be neutral (just like in the case of technical documents).

To illustrate the right choice of font, we have chosen to innovate by not using the Roman script, but fraktur. On fig. 1 the reader can see an example of font choices: three texts have been typeset in fraktur, the first one is from Carl Philipp Emanuel Bach's "Versuch über die wahre Art das Clavier zu spielen" (1753), the second from E.T.A. Hoffmann's "Der goldene Topf" (1822) and the third from H.K. Stein's "Lehrbuch der Geschichte" a schoolbooks published in 1903. All three have been typeset in fraktur and for most readers—even German speaking ones—all fraktur looks the same. This idea is also reinforced by modern font foundries which often include only a single fraktur font in their catalogs ("Wittenberger Fraktur" for Linotype, "Walbaum Fraktur" for URW, etc. where both foundries also provide "Fette Fraktur," which is purely decorative). This is certainly not true: just

Man hat ausser vielen Arten der Claviere, welche theils wegen ihrer Mångel unbekant geblieben, theils noch nicht überall eingeführt sind, hauptsächlich zwen Arten, nemlich die Flügel und Clavicorde, welche bis hieher den meisten Benfall erhalten haben. Jene braucht man insgemein zu starcken Musicken, diese zum allein spielen. Die neuern Forte piano, wenn sie dauerhaft und gut gearbeitet sind, haben viele Vorzüge, ohngeachtet ihre Tractirung besonders und nicht ohne Schwierigkeit ausstudiret werden muß. Sie thun gut benm allein spielen und ben einer nicht gar zu starck gesetzen Music, ich glaube aber doch, daß ein gutes Clavicord, ausgenommen daß es einen schwächern Ton hat, alle Schönheiten mit jenem gemein und überdem noch die Bebung und das Tragen der Tone voraus hat, weil ich nach dem Anschlage noch jeder Note einen Druck geben kan. Das Clavicord ist also das Instrument, worauf man einen Clavieristen auß genaueste zu beurtheilen sähig ist.

Mit Recht draf ich zweifeln, daß du, günstiger Leser, jemals in einer gläsernen Flasche verschlossen gewesen sein solltest, es sei denn, daß ein lebendiger neckhafter Traum dich einmal mit solchen seeischen Unwesen befangen hätte. War das der Fall, so wirst du das Elend des armen Studenten Anselmus recht lebhast fühlen; hast du aber auch dergleichen nie geträumt, so schließt dich deine rege Phantasie mir und dem Anselmus zu Gefallen wohl auf einige Augenblicke in das Aristall ein. — Du bist von blendendem Glanze dicht umflossen, alle Gegenstände rings umher erscheinen dir von strahlenden Regenbogensarben erleuchtet und umgeben — alles zittert und wankt und dröhnt im Schimmer — du schwimmst regungsend bewegungslos wie in einem sestgefrorenen Äther, der dich einpreßt, do daß der Geist vergebens dem toten Körper gebietet.

Die Geschichte sucht als Geschichtsforschung zu ergründen, was geschehen ist. Aber nicht alle Geschehnisse sind für sie von gleicher Bedeutung. Da der Mensch von den geschaffenen Wesen das wichtigste ist, so nimmt von aller Geschichte die Geschichte der Menschheit den ersten Rang ein. Nun ist aber nur da, wo es eine Entwicklung, ein Werden, eine Veränderung gibt, Geschichte möglich. Darum beschäftigt sich die Geschichte hauptsächlich mit denjenigen Völkern, die sich weiter entwickelt haben, deren Werse und Taten für den Werdegang der Menschheit wichtig waren oder noch sind. Das Material das dem Geschichtsforscher die Erkenntnis des Geschehenen ermöglicht, nennen wir die Quellen. Diese können von zweierlei Art sein. Entweder sind es Berichte von Zeitgenossen oder von Späteren, die den Zweck haben, die Kenntnis der Ereignisse der Mits und Nachwelt zu übermitteln.

Figure 1. Three texts typeset in different fraktur fonts, according to their history and contents.

like there are many Roman typefaces, drawn in different times and places, so there also many fraktur typefaces. The ones we have chosen for these texts are "Wartburg" (a 18th century typeface), "Unger" (drawn in 1794), "Normale Fraktur" ("normal type," the most common typeface of the 19th century).

We have chosen this example to show that often a little historical research is a better solution for choosing the right font ("what would choose a printer contemporary to the author?") than just personal taste ("I like Garamond very much, let us use it for all books").

"Neutral" typography

But what about technical documents or, more generally, documents needing a neutral appearance, where the reader's attention must be concentrated on the contents and only? This is a very common problem in the TEX world, since TEX is generally more often used for technical and scientific documents than for literary ones.

By definition, a "neutral" typeface is a typeface that doesn't attract attention, an "ordinary" typeface. But "ordinary" does not necessarily means "ugly." The typefaces seen most often today in technical documents are Computer Modern for TeX documents, and Times for the rest of the world. Computer Modern is a nice, a bit old-fashioned font. If it hasn't been chosen by D.E. Knuth for TeX, we probably would not call it "neutral" today (the closest commercial fonts are ITC Modern and Monotype Modern: these are certainly not "neutral" fonts, and give an old-fashioned look to a book), but so many documents have been typeset in Computer Modern that our eyes have become used to it.

If you want to use a Times-like font then Adobe Times (on the Mac) and Monotype New Times Roman (on Windows) are not your only choices: there are other, very nice "neutral" fonts, like Times Ten, Utopia, Le Monde, ITC Century or Linotype Centennial. On fig. 2 the reader can see a few lines taken from a technical document, typeset in the five typefaces we mentioned: Times Ten, Utopia, Le Monde, ITC Century and Linotype Centennial. These are only a few examples from the long list of typefaces suitable for technical documents.

 $^{^9}$ While Ω provides Times-like fonts as default text fonts.

But freshwater resources are limited and unevenly distributed. In the high-consumption countries with rich resources and a highly developed technical infrastructure, the many ways of conserving, recycling and re-using water may more or less suffice to curb further growth in supply. In many other regions, however, water availability is critical to any further development above the present unsatisfactorily low level, and even to the mere survival of existing communities or

But freshwater resources are limited and unevenly distributed. In the high-consumption countries with rich resources and a highly developed technical infrastructure, the many ways of conserving, recycling and re-using water may more or less suffice to curb further growth in supply. In many other regions, however, water availability is critical to any further development above the present unsatisfactorily low level, and even to the mere survival of existing

But freshwater resources are limited and unevenly distributed. In the high-consumption countries with rich resources and a highly developed technical infrastructure, the many ways of conserving, recycling and re-using water may more or less suffice to curb further growth in supply. In many other regions, however, water availability is critical to any further development above the present unsatisfactorily low level, and even to the mere survival of existing

But freshwater resources are limited and unevenly distributed. In the high-consumption countries with rich resources and a highly developed technical infrastructure, the many ways of conserving, recycling and re-using water may more or less suffice to curb further growth in supply. In many other regions, however, water availability is critical to any further development above the present unsatisfactorily low level, and even to the mere survival of existing communities

But freshwater resources are limited and unevenly distributed. In the high-consumption countries with rich resources and a highly developed technical infrastructure, the many ways of conserving, recycling and reusing water may more or less suffice to curb further growth in supply. In many other regions, however, water availability is critical to any further development above the present unsatisfactorily low level, and even to the

Figure 2. A few lines from a technical document typeset in five different "neutral" fonts, suitable for technical documents: Times Ten, Utopia, Le Monde, ITC Century and Linotype Centennial.

Basic Typesetting Rules

These are the very fundamental rules for typesetting a given language, in a given country. You can find them described in books such as the *Chicago Manual of Style* [1] for US English or *Hart's Rules for Compositors and Readers* [3] for UK English, the *Lexique des règles typographiques en usage à l'Imprimerie Nationale* [4] for French, the *Duden-Taschenbuch Band 5 Satzund Korrekturanweisungen, Richtlinien für die Texterfassung* [2] for German, and so on.

Wether they deal with punctuation, the use of italics or small caps, abbreviations, the use of lower and upper case, etc. most of these rules are clear and unambiguous. Some of them can be applied directly by TEX itself, others can be applied by pre-processor scripts and finally others need the typesetter's attention.

Principles of Good Typography

Besides being a technique, typography is also an art. And an art is harder to learn than just a few rules. Of course there are very good books on typography around: books by Tschichold [10], Gill [7], Rogers [8], Wilson [11], Bringhurst [5], Solomon [9], Willberg/Forssman [12] (in German), Duplan/Jauneau [6] (in French) and many others will give you some insight (see also our Web site [14] for an additional list of interesting books on various topics). But still you only really learn by doing it, by seeing how others have done it, and by discussing with specialists.

2. Characters to Draw: Letters, Symbols, Diacritics,...

It happens often that an author needs a special symbol, wether this is a mathematical symbol, a pictogram, a phonetic character, a special accent to place upon or beyond an ordinary letter, etc.

The best solution is to *draw* this character and to include it into a new or an already existing font. This is the best solution, for the following reasons:

- 1. by including the character into the same font, you can kern it with other characters;
- 2. by making a regular character out of it (and not some weird construction made out of boxes, \accents, \raises, rules, etc.) it can participate in hyphenation;

- 3. only by drawing a character you have complete control on his form and can make it fit with the surrounding characters;
- 4. (this applies only to accented letters) accents often change form when placed on different letters: for example accents placed on uppercase letters are lower than the ones placed on lowercase letters; in Greek, the circumflex accent is narrower when placed on the iota than when placed on the omega, etc. Inversely, sometimes the shape of a letter may change because an accent has been placed upon it: in the case of the Esperanto 'h' with circumflex accent, we can very well imagine the vertical stroke of the 'h' being lower so that the accent doesn't get too high.

Drawing characters also allows us to apply rules of optical correction, that is the variation of shape according to the size of printed letters. This very important method has been successfully applied to the Computer Modern family: CMR5, CMR7, CMR10, CMR12, CMR17 (Computer Modern Roman at 5–17 points) are all the same font, but the shape of letters is (slightly) different. This method can also be applied to non-Metafont fonts, in a more manual and less precise manner: by redrawing characters for each important size. See fig. 3 for an example of optical correction of Greek letter alpha (at 10 and 7 points size).

How do I Draw Characters

All depends on the kind and use of the characters. If you need to draw a geometrical symbol, you will get better results using Metafont or Metapost and then converting the result to PostScript. Especially if it is a mathematical symbol which has to fit with other Metafont-generated mathematical symbols, like those in CMSY or in the AMS fonts.

If it is not geometrical, then you'll better draw it in a font drawing program like Fontographer or FontLab or Ikarus, and then save it in PostScript form. With these programs you can also open existing fonts and redraw, compose or re-arrange the glyphs (of course you can do this only for personal use, since copyright of the original font applies).

You can also ask a specialist to do it for you: fonts are complex objects and there is a long way from the first sketch of a character to the accomplished real-world usable font.

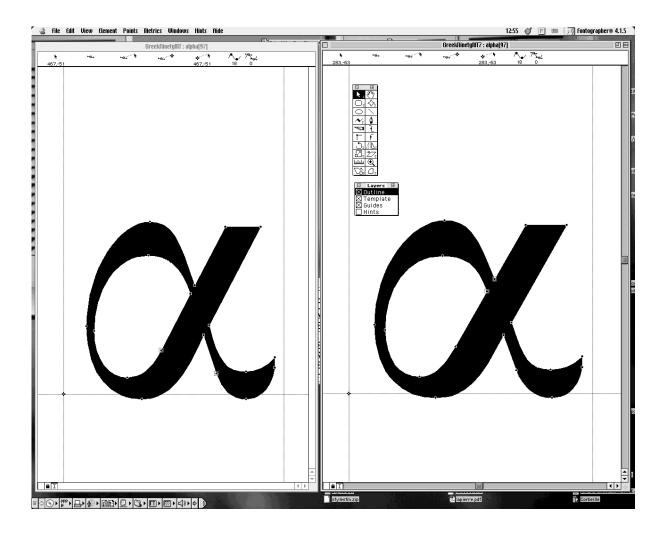


Figure 3. Working on the optical correction of letter alpha for Greek text: on the left, the original letter for 10 points typesetting; on the right the same letter corrected for 7 points typesetting.

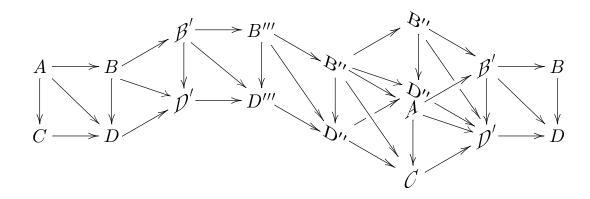


Figure 4. A mathematical construction made under Illustrator out of pieces made by XYPic.

3. Mathematics

Typesetting of mathematics is a very interesting challenge. It is especially important for the TeX community, since D.E. Knuth has developed TeX in the first place to typeset the mathematics of his own books. One could expect that after D.E. Knuth's work, and twenty years of collective efforts, all problems of mathematical typesetting should be resolved. Also it is a very pleasant fact that TeX has slowly become a universal method for communicating and storing mathematical texts, so that mathematicians conform their style of writing to TeX's range of possible mathematical constructions.

But still, sometimes mathematics can be so complex that even the combined efforts of T_EX, AMS-L^AT_EX, XYPic and YHMath cannot typeset them conveniently. These are rare cases, and for them one can always use graphical tools like Metapost or Adobe Illustrator to create the desired construction out of existing pieces. For example, the commutative diagram on fig. 4 has been made under Illustrator using pieces from constructions made by XYPic. The fact that some letters are shown under perspective is supposed to increase the legibility of this "2.5-dimensional" mathematical expression.

When a mathematical symbol is missing, it has to be drawn (see § 2). Sometimes this involves using existing glyphs, and making precise measurements. For example, to draw the symbols on fig. 5, the author had to first make precise measurements of the distance of integral glyphs in AMS-LATEX constructions \iint and \iiint, then combine two (resp. three) integral glyphs with that precise distance into a single character, and finally draw the ellipses.

$$\frac{mNq^2}{2} = \frac{P}{2} \oiint \vec{r} \cdot d\vec{S}$$

$$\frac{mNq^2}{2} = \frac{3P}{2} \oiint dV = \frac{3PV}{2}$$

Figure 5. Double and triple curvilinear integral signs: An example of mathematical characters specially drawn.

This has been done in a PostScript font editor, because these symbols are part of the SMF Baskerville family of math fonts (see § 3). But when the surrounding math symbols are designed in Metafont, then it is easier to stay with that font design language. The YHMath package does exactly that: to produce larger delimiters and wider accents, it adds another 128 glyphs to the CMEX font. Here is an (hypothetical) example of the use of YHMath:

$$\det\begin{pmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \\ u & v & w & x & y \end{pmatrix} = \operatorname{area}(\widehat{A_{ay}B_{ay}C_{ay}}) - \begin{pmatrix} y & x & w & v & u \\ t & s & r & q & p \\ o & n & m & l & k \\ j & i & h & g & f \\ e & d & c & b & a \end{pmatrix}.$$

Typesetting Mathematics in Typefaces other than Computer Modern or Times

In 1 we have discussed the possibility of using fonts other than Computer Modern or Times for technical documents. What happens with mathematics included in such documents?

The only mathematical typefaces widely available for TEX are Computer Modern, Mathtime and Lucida: a very small number of typefaces compared to the large number of existing typefaces for text. The author has had the opportunity to design other mathematical typefaces for specific projects: the mathematics had to fit with the text typefaces Stempel Garamond, Sabon and ITC New Baskerville. The latter is called "SMF Baskerville" and has be drawn especially for the French Mathematical Society. On fig. 6 the reader can see an example of mathematics typeset in SMF Baskerville.

Drawing a mathematical typeface using a text typeface as model is an in-

If a function ρ belongs to $L^1(\mu_K)$ then for the conditional expectation of ρ with respect to the inverse image of the whole σ -field under f^k , we have the formula:

$$E_{\mu_K}(\rho | f^{-k}(\mathcal{B}))(x) = \sum_{y \in f^{-k}(f^k(x))} \frac{\rho(y)}{\psi(y)}.$$

Hence:

(8.5)
$$\int \left(\frac{\psi}{\varphi} - 1\right) d\mu_{K} = \int E_{\mu_{K}} \left(\frac{\psi}{\varphi} - 1 \mid f^{-k}(\mathcal{B})\right) d\mu_{K}$$
$$= \int \sum_{y \in f^{-k}(f^{k}(x))} \left(\frac{1}{\varphi(y)} - \frac{1}{\psi(y)}\right) d\mu_{K}(x) = \int (1 - 1) d\mu_{K} = 0.$$

Assume now that μ_K is a measure with maximal entropy for $f^k|_K$, and that $x \in K \setminus B$, $f^{kn}(x) = x$. By (8.4) and (8.5), we have $\varphi = \psi$, μ_K -almost anywhere. The whole trajectory of x (under f^k) is disjoint from B and hence, in view of (8.1), (8.2) and Theorem (6.2) (c), φ and ψ are contiguous in some neighbourhood of the trajectory of x. Thus, there are equal on the whole trajectory of x. Therefore by (8.1) and (8.2) we obtain:

$$\left| (f^{nk})'(x) \right| = \prod_{i=0}^{n-1} \left| (f^k)'(f^{ik}(x)) \right| = \prod_{i=0}^{n-1} \psi(f^{ik}(x)) = \prod_{i=0}^{n-1} \varphi(f^{ik}(x)) = \beta^n.$$

Figure 6. Mathematics typeset in SMF Baskerville, according to the tradition of the *Publications mathématiques de l'I.H.E.S.*

teresting experience. Latin letters have to be redrawn (they have to be larger and their position in the character box often changes entirely). Geometrical symbols (that is: mathematical symbols made out of geometric forms, like $+,-,\times,\otimes,\triangle$ etc. contrarily to "alphabetical symbols" like ∇,\int,\sum,\coprod , etc.) have to be redrawn with a different width of stroke, different extremities and corners (sharp or round) and a different mathematical axis.

But the most interesting is the drawing of Greek letters: these must keep their very specific form (Greek letters used in mathematics have fixed forms and cannot adopt the design characteristics of their Latin counterparts like in the case of Greek text fonts) but at the same time must be homogenized with Latin letters. To achieve this, their heights and widths, and widths of thin and fat strokes must be modified so that they optically fit with Latin letters.

Mathematics and the Web

Sometimes we have an additional constraint: the mathematics of a given book must be coded in such a way that the book can be easily ported to the Web. In other words, one must choose a method of publishing mathematics on the Web (\LaTeX TEX2HTML, TEX4HT, IBM TechExplorer, TEXSpider, WebEQ Java applets, MathML produced by Ω or by WebEQ, Acrobat PDF, etc.) and write (or re-write) the code in such a way that it is compatible with the chosen method. This issue is out of the scope of this article.

4. Multilingual Documents

Various issues arise when we are dealing with multilingual documents. In some cases we want to make different languages look similar, in other cases we want them to look different. Independently of how we present them, languages have their own rules, grammatical, or typographical. In the latter case we may want to impose the typographical rules of one language (the "background language") to others, or keep the rules of each language (which often leads to weird problems when we have to decide to which language belong neutral elements such as digits or punctuation).

On fig. 7 the reader can see an illustration of the two methods: (a) making scripts look different, this can be useful when we have short insertions of text in a different script and we want the reader's eye to immediately notice the difference. To achieve this we take the most traditional form of each script: in the example, French is typeset in Didot, Greek in Monotype

Making scripts look different. A little French: un peu de français dans une police Didot, and Greek: λίγα έλληνικὰ γιὰ νὰ δοῦμε τὴν διαφορά, Russian: Βce шрифты κлассифицируют по группам, Armenian: ԳոՀուխինն Աստուծոյ, and finally Georgian: მხელრული ასოები.

Making scripts look similar. A little French: un peu de français dans la même police Times, and Greek: λίγα έλληνικὰ γιὰ νὰ δοῦμε τὴν διαφορά, Russian: Βce шрифты κлассифицируют по группам, Armenian: Գոhութի΄ιն Ասsուծոյ and Georgian: მხელრული ასოები.

Figure 7. The two methods of mixing different scripts.

Greek 90, Russian in an old-fashioned Russian font, Armenian in traditional oblique (with uppercase letters straight) and Georgian in a nice traditional font. Method (b) is the exact opposite: we want all scripts to look as close as possible (for example, Latin 'o', Greek omicron, Cyrillic 'o' are all identified, also identified are Latin 'n' and Armenian 'a', etc.). In the example we have used an ordinary Times font. This method is more efficient if we are typesetting well separated text blocks in different languages and want to keep a globally homogeneous presentation; a typical example are publications of the European Union, where everything is translated in all languages of the Union.

The French/Greek dictionary page we are showing on fig. 8 is also a typical example of method (a): since Greek and Latin scripts are very close and since in such a dictionary French and Greek words constantly alternate, the two scripts have to be presented in a clearly different way. The author has chosen the notorious French sans serif typeface Univers for the

¹⁰In Greece the author has often seen graffiti accidentally mixing the two scripts, like in 'NO ΠΑΡΚΙΝG', that is 'NO PARKING' where the first three letters of the second word are Greek, and while writing the next three (who are indeed common to the two scripts) the writer has forgotten he was writing in Greek—after all these are English words—and finished the inscription with a Latin 'G.'

Α

- **A,a** [a] όν. αρσ. πρώτο γράμμα του αλφαβήτου. De A à Z, από το A ως το Ω . Prouver par A + B, αποδεικνύω τελειωτικά.
- à [a] $\pi \rho \delta \theta$. I. $\sigma \epsilon$. 1. Je vais à Paris. $\Pi \eta \gamma a \acute{u} \nu \omega$ $\sigma \tau \sigma$ $\Pi a \rho \acute{u} \sigma \iota$. 2. Je pars à quatre heures. $\Phi \epsilon \acute{v} \gamma \omega$ $\sigma \tau \iota s$ $\tau \acute{e} \sigma \sigma \epsilon \rho \iota s$. ii. $\mu \epsilon$. 1. Ce moteur marche à l'essence. $A \upsilon \tau \acute{\eta} \eta \mu \eta \chi \alpha \upsilon \acute{\eta} \delta \sigma \upsilon \lambda \acute{e} \iota \iota \mu \epsilon \beta \epsilon \upsilon \acute{\zeta} \acute{\iota} \upsilon \eta$. 2. Il y avait environ de mille à mille cinq cents manifestants. $Y \pi \acute{\eta} \rho \chi \alpha \upsilon \tau \pi \epsilon \rho \acute{\iota} \tau \sigma \upsilon \chi \acute{\iota} \lambda \iota \iota \iota \iota \mu \epsilon \chi \acute{\iota} \lambda \iota \iota \sigma \upsilon s$ $\pi \epsilon \upsilon \tau \alpha \kappa \acute{\sigma} \iota \sigma \upsilon s$ $\delta \iota \alpha \delta \eta \lambda \omega \tau \acute{e} s$. iii. une machine écrire, $\mu \iota \alpha \gamma \rho \alpha \phi \rho \iota \eta \chi \alpha \upsilon \tau \acute{o}$. iii. une machine écrire, $\mu \iota \alpha \gamma \rho \alpha \phi \rho \iota \eta \chi \alpha \upsilon \acute{v} \acute{v}$. iv. 1. Il obéit à ses parents. $Y \pi \alpha \kappa \sigma \acute{\iota} \iota \tau \sigma \upsilon s$ $\gamma \sigma \upsilon \kappa \acute{\iota} s$ $\tau \sigma \upsilon \upsilon$. Il est fidèle à ses promesses. $K \rho \alpha \tau \acute{a} \tau \iota s$ $\upsilon \sigma \sigma \chi \acute{e} \sigma \iota s$ $\tau \upsilon \iota \iota \sigma \upsilon s$. Le livre est à moi. $T \sigma \beta \iota \beta \lambda \acute{\iota} \circ \acute{\iota} \nu \alpha \iota \delta \iota \kappa \acute{o} \mu \sigma \upsilon$.

abaissant — $eta \lambda$. abaisser 2.

- 1. abaisser [abese] ρ. μτβ. Α. (3). κατεβάζω, χαμηλώνω. 1. la vitre de la voiture, κατεβάζω το τζάμι του αυτοκινήτου. 2. les impôts, κατεβάζω τους φόρους. abaisser (s') Ε. (5). à la fin du spectacle, le rideau s'abaisses. Στο τέλος του θεάματος, η αυλαία χαμηλώνει. abaissement όν. αρσ. 1. κατέβασμα n. 2. ελάττωση, μείωση f. (= baisse, diminution).
- abaisser ρ. μτβ. Α. (3). υποβιβάζω, ταπεινώνω. qqn, υποβιβάζω κπ. abaisser (s') Ε. (5). ταπεινώνομαι. Il s'est abaisse à me demander de l'argent. Ταπεινώθηκε με το να μου ζητήσει χρήματα. abaissant, -ante επ. ταπεινωτικός, τή, -ό.
- abandonner [abādɔne] ρ. μτβ. Α. (3). I. εγκαταλείπω, παρατώ (-άω), αφήνω (= laisser). 1. un lieu, un bateau, εγκαταλείπω ένα μέρος, ένα πλοίο. 2. sa famille, εγκαταλείπω / παρατώ την οικογένειά μου. 3. une théorie, αφήνω μια θεωρία. ii. un nouveau-né, εκθέτω ένα βρέφος. iii. J'abandonne! Tα παρατώ ! abandonner (s') E. (5). 1. εγκαταλείπομαι. au désespoir, εγκαταλείπομαι στην απελπισία. 2. εξομολογούμαι (= se confier). abandonné, -ée επ. εγκαταλειμμένος, τ η, -ο. maison —, εγκαταλειμμένος τάλειψη f. de biens, de poste, εγκατάλειψη περιουσίας, θέσης. 2. νωθρότητα f. (= nonchalance). abandon (à L') επίρρ. Le parc est à l'abandon. Το πάρκο είναι εγκαταλειμμένο.
- **abaque** [abak] ($\mu\alpha\theta\eta\mu$. κλπ.) όν. αρσ. άβακας m.
- abasourdir [abazurdir] ρ . $\mu\tau\beta$. A. (23). 1. ξεκουφαίνω, ζαλίζω. 2. ($\mu\tau\phi$.) καταπλήσσω / καταπλήττω. Cette nouvelle m'a abasourdis. $A\nu\tau\eta$ η είδηση $\mu\epsilon$ κατέπληξε. abasourdissant, -ante $\epsilon\pi$. 1. εκκωφαντικός, -ή, -ό. 2. κατάπληκτος / κατάπληχτος, -η, -ο. abasourdissement όν. αρσ. κατάπληξη f. σάστισμα n.
- **abâtardir** [abotardir] ρ. μτβ. Α. (23). μπασταρδεύω, νοθεύω. **abâtardir** (**s**') Ε. (25). νοθεύομαι. **abâtardissement** όν. αρσ. μπαστάρδεμα n. **abat-jour** — [abaʒur] όν.αρσ. αμπαζούρ n. inv.

- abats, abattage $\beta\lambda$. abattre 1. / abattement $\beta\lambda$. abattre 1 κ . 2.
- **abattis** [abati] όν. αρσ. πληθ. 1. (πουλερικών) πλατάρια n. pl. **2.** \circ Numérote tes abattis ! Πρόσεχε, θα σε σκοτώσω !
- 1. abattre [abatr] ρ. μτβ. Α. (69). Ι. ρίχνω κάτω, καταρρίπτω, κόβω (= couper). 1. un mur, ρίχνω κάτω έναν τοίχο. 2. un avion, καταρρίπτω / κατεβάζω ένα αεροπλάνο. ΙΙ. ξεκάνω, σκοτώνω (= tuer). 1. un animal, ξεκάνω ένα ζώο. 2. qqn. ξεκάνω κπ. ΙΙΙ. 1. du travail, βγάζω πολλή δουλειά. 2. (χαρτοπ.) son jeu, κατεβάζω τα χαρτιά μου. abattre (s') Ε. (69. 5). καταπέφτω. abattage όν. αρο. 1. (ζώων) σφάξιμο n. σφαγή f. 2. ° avoir de l' —, είμαι γεμάτος δυναμισμό. abattement όν. αρσ. (φόρων) ελάττωση π. habattoir όν. αρσ. σφαγείο n. abatts όν. αρσ. πληθ. (ζώων) εντόσθια n. pl.
- abattre ρ. μτβ. Α. (69). (μτφ.) τσακίζω, σπάζω / σπάω (= briser).
 1. Cet échec l'a abattu. Αυτή η αποτυχία τον τσάκισε.
 2. le courage de qqn, τσακίζω το κουράγιο κάποιου. abattu, -e. επ. 1. τσακισμένος, εξασθενημένος, -η. -ο. 2. αποθαρρημένος, -η, -ο. (= découragé). abattement όν. αρα 1. εξασθένηση, εξάντληση f. (= épuisement).
 2. αποθάρρυση, κατάπτωση (= prostration) f.
- abbé [abe] όν. αρσ. 1. ιερωμένος, κληρικός m. 2. (μοναστηριού) αβάς, ηγούμενος m. l' — Pierre, ο αβάς Πιερ. abbatial, -e, aux. επ. 1. ηγουμενικός, -ή, -ό. 2. μοναστηριακός, -ή, -ό. abbaye όν. θηλ. αβαείο n. abbesse όν. θηλ. ηγουμένη f.
- A B C [abese] όν. αρσ. 1. αλφαβητάριο n. 2. C'est l'a b c du métier. Είναι το άλφα βήτα της τέχνης. abécédaire όν. αρσ. αλφαβητάριο.
- abcès [apse] $\delta \nu$. $\alpha \rho \sigma$. $\alpha \pi \delta \sigma \tau \eta \mu \alpha$ n.
- **abdiquer** [abdike] ρ . $\alpha\mu\tau\beta$. $\mu\tau\beta$. A. (3). $\pi\alpha\rho\alpha\iota$ τού $\mu\alpha\iota$. 1. Le roi a abdiqué. O $\beta\alpha\sigma\iota\lambda\iota\acute{a}s$ $\pi\alpha\rho\alpha\iota$ τήθηκε. ii. $(\mu\tau\phi.)$. J'abdique! Π αραιτού $\mu\alpha\iota$! abdication όν. θηλ. $\pi\alpha\rho\alpha\iota$ τηση f.
- **abdomen** [abdɔmɛn] όν. αρσ. κοιλιά f. (= ventre). **abdominal**, -**ale**, -aux επ. κοιλιακός, -ή,
- **abécédaire** $\beta\lambda$. A B C.
- **abeille** [abεj] $\acute{o}\nu$. $\theta\eta\lambda$. $\mu\acute{e}\lambda\iota\sigma\sigma\alpha$ f.
- aberrant -ante [aberã, -ᾶt] επ. 1. εξωφρενικός, -ή, -ό. C'est aberrant! Είναι αφάνταστο! 2. εξαιρετέος, -έα, -έο. aberration όν. θηλ. εξωφρενικότητα f.
- abêtir [abetir] ρ. μτβ. Α. (23). αποβλακώνω. abêtir (s') Ε. (25). αποβλακώνομαι. abêtissant, -ante επ. αποβλακωτικός, -ή, -ό. abêtissement όν. αρσ. αποβλάκωση f. αποβλάκωμα Π.
- abhorrer [abɔre] ρ . $\mu\tau\beta$. A. (3). $a\pi\epsilon\chi\theta\acute{a}\nu\rho\mu\alpha\iota$. la grossièreté, $a\pi\epsilon\chi\theta\acute{a}\nu\rho\mu\alpha\iota$ $\tau\eta$ $\chi\nu\delta\alpha\iota\acute{o}\tau\eta\tau\alpha$. abîme $\beta\lambda$. abîmer 2.
- 1. abîmer [abime] ρ. μτβ. Α. (3). χαλώ (-άω), στραπατσάρω, καταστρέφω. 1. L'humidité a abîmé les murs. Η υγρασία χάλασε / κατάστρεψε το τοίχο. 2. qqn, στραπατσάρω κπ.

Figure 8. A page from a French/Greek dictionary, typeset in Univers and Monotype Porson.

1

French text, and the equally notorious scholarly Greek typeface Porson (of English origin) for the Greek text. These two typefaces contrast in almost every extent: the first is sans serif, straight, thin or fat, of French origin; the second is serifed, slanted, of medium weight and of English origin.

For the phonetic description of French words we have taken a condensed version of Stone Phonetic. Not everyone is interested in the phonetic description, this is why we want to keep it as small as possible; on the other hand the choice of a sans serif font makes it more legible, although it is condensed and brings it closer to the French text. The font chosen for the entries and sub-entries is a bold Univers: this makes these subentries immediately identifiable inside the text. In this dictionary Greek is never typeset in bold.

Another problem that arises when we typeset multilingual documents is related to hyphenation. For most languages we have already hyphenation algorithms. Nevertheless one should always be ready to modify these algorithms. Here is what can happen:

- 1. New characters can get involved, needing special patterns (long and short 's' for German, pre-Leninian Cyrillic letters for Russian, etc.)
- 2. In dictionaries and primers, additional signs are often used to simplify the learning of the given language; for example, in Russian, acute accents are used to show which vowel must be stressed. These signs should not interfere with hyphenation.
- 3. Sometimes we use transcriptions instead of the original alphabet, for example Greek is often transcribed in the Latin alphabet. In that case the hyphenation algorithms and the transcriptions must be compatible.
- 4. Hyphenation rules can vary according to the location where a book is published. For example, ancient Greek is not hyphenated in the same way in Greece and in the United States. This means that the language and dialect information is not always sufficient for choosing properly a document's hyphenation rules.

In some cases people even argue if hyphenation should be applied or not: this the case of Ivrit (modern Hebrew). Classical Hebrew is never hyphenated (just as Arabic and Syriac) and as a derivative of classical Hebrew, Ivrit should not be hyphenated as well; but Ivrit has often long words (of

foreign origin) and these could very well be hyphenated. Wether an algorithm is able to recognize these words is another (open) question. We have the same problem with Yiddish: being a German dialect, there is no reason why Yiddish shouldn't be hyphenated, and indeed it is. The Yiddish Institute (YIVO) even provides hyphenation rules; but one of the rule is *not to hyphenated Hebrew words*. Can an algorithm distinguish these from German ones?

We see that often to do proper typesetting we need first to apply natural language processing methods.

5. Natural Language Processing Methods Needed for Typesetting

Many minor grammatical phenomena can be handled automatically by the typesetting software, for example the fact that in Greek or Dutch, when a broken diphthong is hyphenated, the dieresis disappears (for example the Dutch word "poëeten" is hyphenated "po-eeten"), in German when 'ck' gets hyphenated, it becomes 'k-k' and another similar phenomena.

But in some cases a real morphological analysis is needed, which of course cannot be carried out by the software alone.

Our first example is an ongoing project of the author, dealing with the long and short 's' in German, as well as with the use or non-use of certain ligatures. In broken script, there are two kinds of 's' the "short s" \$\sigma\$ and the "long s" \$\int\$. In German, the former is used at the end of the word, and at the end of word components inside a composite word; the latter is used in all other cases. So, for example, 'das' is written with a short s: \dash, while 'dasein' is written with a long one: \dashafeti. The word 'aus' is also written with a short s: \ausi; when this word is used to make a composite, it retains the short s: 'ausmachen' \ausmachen; but a word can start with 'aus' without being a composite: 'ausser' \auffer. Furthermore any German word can be combined with others to produce composite ones; no word list, or dictionary can predict all possible combinations. And German words get declined: such a word list should also take into account all possible forms.

The long and short 's' are not the only cases where the morphology of words plays a rôle in typesetting: there is also a rule saying that there should be no ligatures formed by letters belonging to different components of a composite word. For example, in the word 'Auflage' there should be no f-l ligature: Auflage and not Auflage. These rules affect also the

typesetting of German in Roman type, at least for the ligatures 'fi', 'ff', 'ffi', 'ffl', 'ffl' that are still in use in Roman type.

All of these problems can be solved automatically by using Natural Language Processing tools. Currently the author is working upon adapting Oliver Lorenz's DMM [13] (Morphological Analyzer of German, based on the programming language Malaga) to this task. By preprocessing texts (or by using this tool as an external Ω Translation Process) one can detect and treat all cases of short and long 's' and broken ligatures.

A similar problem arises in Thai: in South East Asian languages (besides Vietnamese), words are not separated inside a sentence. Nevertheless type-setting software must be able to separate words because sentences can be very long. Unicode provides a word-separator character, but Thai people are not accustomed into using it. The Thai TEX package provides a pre-processor which analyzes text and inserts the necessary word-separator marks (in the forthcoming Ω Thai package this pre-processor will be included internally as a Translation Process).

6. Parallel Texts and Critical Editions

A typical example of parallel texts is a literary text and its translation(s). In the Renaissance many polyglot Bibles have been printed, featuring sometimes up to six parallel versions of the same text (in different languages, scripts and writing directions). A more general case are critical editions: in this case, not only we have two or more parallel texts, but these also produce several bodies of footnotes, endnotes and critical apparati.

The example on fig. 9 is such a critical edition: it has 5 major blocks of text (not counting headers): the original text (ancient Greek) on the upper even page, translation (French) on the upper odd page, and three blocks underneath: critical apparatus, linked to the Greek text by line number, footnotes and references linked to the translation by numbers or lowercase letters.

All five blocks grow simultaneously: the Greek text and French translation blocks always stay on a single page (even for the former, odd for the latter); the other three blocks spread to occupy the remaining space underneath (even if this space is not symmetric, or is only on the even side or only on the odd side). Any word of Greek text you add may produce entries in the critical apparatus or words in the translation, which again may require footnotes and references.

SUR L'INCOMPRÉHENSIBILITÉ

Je vous exhorte donc à éviter leur folie, car c'est le

comble de la folie que s'acharner à connaître Dieu dans son essence. Et pour que vous compreniez que c'est bien en effet le comble de la folie, je vous le montrerai à l'évidence par le témoignage des écrivains sacrés¹: non seulement ceux-ci ignorent manifestement ce qu'il est dans son essence, mais encore ils ne savent que dire de l'étendue

sagesse, mais la sagesse de l'essence. Quand donc les écrivains sacrés ne peuvent pas même délimiter celle-là avec

exactitude, quelle est la folie de ceux qui croient pouvoir soumettre son essence elle-même à leurs propres raisonnements³ ? Écoutons donc ce que dit l'écrivain sacré à ce

sujet : « La connaissance que tu as de moi m'a été un ob-

de sa sagesse²; or ce n'est pas l'essence qui dérive de la

τὴν οὐσίαν ἐστὶν ὁ Θεός. Καὶ ἵνα μάθης ὅτι μανίας εσχάτης τοῦτο, ἀπὸ τῶν προφητῶν ὑμῖν τοῦτο ποιήσω φανερόν' οί γὰρ προφήται οὐ μόνον τί τὴν οὐσίαν ἐστὶν τῆς σοφίας, ἀλλ' ή σοφία ἐκ τῆς οὐσίας. ʹΌταν δὲ μηδὲ ταύτην δύνωνται καταλαμβάνειν οί προφήται μετὰ 10 ἀκριβείας, πόσης ἂν εἴη μανίας τὸ τὴν οὐσίαν αὐτὴν ὅτι φοδερῶς ἐθαυμαστώθης.» Τί ἐστι «φοδερῶς»; Πολλὰ Διὸ παραινῶ φεύγειν αὐτῶν τὴν μανίαν· μανίαν γὰρ ἀγνοοῦντες φαίνονται, ἀλλὰ καὶ περὶ τῆς σοφίας αὐτοῦ πόση τίς ἐστιν ἀποροῦσι' καίτοι γε οὐχ ή οὐσία ἀπὸ νομίζειν δύνασθαι τοῦς οἰκείοις ὑποδάλλειν λογισμοῖς; Ακούσωμεν τοίνυν τί φησιν ὁ προφήτης περὶ αὐτης΄ ἐξ ἐμοῦ.» Μᾶλλον δὲ «Έξομολογήσομαί σοι, θαυμάζομεν μόνον, ἀλλ' οὐ μετὰ φόδου, οἷον κιόνων φιλονεικεΐν φημι είναι ἐσχάτην τὸ άνωτέρω τὸν λόγον άγάγωμεν «Ἐθαυμαστώθη ή γνῶσίς σου 15

1 μανίαν 2 : μανίας EL DG OVX \parallel 2 έγιογε om. DG \parallel φημι B: transpost. ἐσχάτης Ε O om. cett. \parallel εΐναι : οίμαι VX om. DG \parallel ἐσχάτης ΕL DG OVX om. B \parallel τὸ om. DG \parallel 3 μανίας ἐσχάτης : μανίαν ἐσχάτην AC om. B \parallel 9 μετ' corr. Due \parallel 16 μόνον : νῦν Ε om. CL G VX.

a. Ps. 138, 6. b. Ps. 138, 14. Le mot προφήτης a des nuances diverses dans l'Ancien et le Nouveau Testament et à l'intérieur de chacun d'eux, mais ces sens divers ont un caractère commun : le prophète est l'homme qui parle sous l'inspiration de l'Esprit. Cf. hom., III, li. 150 s. C'est ainsi que Jean va citer successivement des textes du Psalmiste, d'Isaïe et de Paul.
 On remarquera que Jean utilise très fréquemment la tournure

οὸ μόνον... ἀλλὰ καί. C'est sans doute une habitude de style enseignée par la rhétorique mais elle correspond chez lui, croyonsnous, à une tendance profonde. Son dynamisme naturel ne se contente pas d'une constatation pure et simple; il la renforce soit

Figure 9. Sample of critical edition (John Chrysostome, *On the non-understandability of God*).

jet d'admiration^a. » Mais suivons plus loin son propos : « Je te bénirai, parce qu'on t'admire avec crainte^b. » Que signifient ces mots : « avec crainte »? Nombreuses sont les choses que nous nous contentons d'admirer⁴, mais non

dans un sens positif, soit dans un sens négatif. Ici, la tournure marque un nouveau point gagné sur l'adversaire, grâce au témoignage d'hommes inspirés. Loin de prétendre connaître l'essence de Dieu, ils ont un mouvement de recul, de confusion, même lorsqu'il s'agit de parler des manifestations de la sagesse.

3. Jean oppose ici la saisie claire du mystère de Dieu, dont se prévalent les Anoméens, καταλαμβάνευν μετὰ ἀκριδείας, à la connaissance conjecturale, forcément imparfaite lorsqu'elle ne s'appuie que sur des raisonnements humains. En fait, Eunome affirme qu'il ne s'appuie pas uniquement sur α ses propres raisonnements », mais grâce à une théorie du langage qu'il tire de

l'Écriture (Gen. 1, 3), il attribue à Dieu l'origine du nom qui le désigne exactement. Voir Apologia, chap. 7, PG 30, 841.

4. Lei encore, le choix des textes permet à Jean de faire progresser son argumentation. L'usage du verbe θανμάζειν dans le premier texte suggère l'admiration dans plus. Mais dans le second, l'adjonction de φοδερᾶς montre l'homme saisi d'une crainte révèrentielle en présence de Dieu, devant l'océan infini de sa sagesse: τὸ ἄπευρον... πέλαγος τῆς τοῦ Θεοῦ σοφίας.

But of course the typesetting system is not a translation engine, and hence is not capable of knowing exactly what part of the translation corresponds to each word of Greek text. It is the authors responsibility to mark up the Greek text and the translation so that the typesetting engine can then interpolate between corresponding tags and typeset the two blocks of text as parallel as possible.

How can T_EX do that?¹¹ There is certainly a way, but the author decided to do it otherwise. He developed a system, called PARALLEL, which extracts lines from DVI files and reassembles them according to a given setup. That way it is easy to control the growth of each one of the text blocks. As for the critical apparatus, which is special because it contains line numbers (and these may be initialized at each new even page), it is recompiled again at each new (double) page.

Footnote and reference numbers are also added automatically, since these also get initialized at every new double page. Hence, the system has to rearrange lines from five different DVI files, to complete them by adding some automatically generated material (line numbers, etc.) and to restart compilation and line extraction of the critical apparatus at every new double page. It is written in Perl, backed up by a MySQL database, the two communicating through the DBI module: the choice of Perl makes PARALLEL a bit slower, but much easier to debug and maintain.

How do you prepare the data for such a critical edition? Every author may find his/her own way; to be able to be compatible with any author's method, and to store the data in a unique and efficient way we have adopted XML as the intermediate format between data preparation and type-setting. Hence the system takes data from five XML files, converts it to T_EX code and runs T_EX (actually Ω , since there is ancient Greek involved) upon it to produce the five DVI files.

In this way, T_EX (Ω) is entirely responsible for what happens on a single line, and PARALLEL is responsible for making paragraphs and pages out of these lines. This would be absurd in any other case (since LATEX for example knows very well how to make paragraphs and pages, even if it sometimes swallows a few floating figures) but in this case it has proven to work efficiently. The author is investigating another possibility where PARALLEL would ask T_EX (Ω) to recompile the different blocks of text many times until a perfect double page is obtained, and then move to the next double page: one of the problems is that once T_EX has made the DVI

¹¹Note that a good T_EXist never asks "can T_EX do that?" but "how can we do that with T_EX?" since, by definition, T_EX can do everything.

68

8. ...Non facile declaratur quænam res sint vere relativæ. [II,II]

من عرف ان هذا احسن فقد عرف اشيء الذي هو احسن منه الا ان تكون و المعرفة توهما لا يقينا فانه ان لم يعرف الشيء الذي به قيل فيه انه احسن قد الله يكن ان لا الله يكون شيء دونه في الحسن فيكون 12 قوله فيه انه احسن كذبا ومن 5 هذا يظهر ان الراس واليد ليست من المصاف الحقيق فانه قد d يعرف 13 ماهية كل واحد 14 منها من حيث هما في الجوهر على التحصيل من غير ان يعرف الشيء الذي هو له راس ولا الشيء الذي هو له يد قال الا ان بالجملة الحكم بالحقيقة على e ما هو من المضاف من سائر المقولات وماليس من المضاف 10 هو مما يصعب 15 ما لم يتدبر مرارا كثيرة فاما التشكك 16 فيها فليس فيه صعوبة 17.

Kitāb al-Maqoūlāt

[Κατηγ. p. 8 b, 9-24]

صرورة ان يكون يعلم ايضا ذلك الشي الذي هذا احسن منه محصلا فانه ليس يجوز 357 ان يكون انما يعلم انَّ هذا احسن مما دونه في الحسن فان ذلك انما يكون توهما لا علما وذلك انه ليس يعلم يقينا انه احسن مما هو دونه فانه ربما اتفق الا ليكون شي* دونه فيكون قد ظهر انه واجب ضرورة متى علم الانسان احد المضافين محصلا ان يكون [15] يعلم ايضا[†] ذلك الاخر الذي اليه اضيف محصلا. فاما الراس و اليد وكل*[†] واحد مما a t* يجري مجر اهما مما هي جواهر فان ماهياتها t انفسها قد تعرف محصلة فاما ما يضاف t* اليه فليس واجبا ان يعرف وذلك انه لا سبيل الى ان يعلم على التحصيل راس من هذا ويد من هذه فيجب من ذلك ان هذه ليست من المضاف واذ لم تكن هذه من $^{ au}$ يكون قد يصعب التقحم على اثبات الحكم على امثال هذه الامور ما لم تتدبّر مرارا كثيرة فاما التشكك فيها فليس مما لا درك فيه. LECT. $\mathbf{t}:$ — (359) ان \mathbf{y} ان لا (361) ماهیتها (362) ایضا یعلم (361) ایضا یعلم (361) -یتُدبر (366) — واذا (364) — تضاف

Figure 10. Sample of critical edition (Averroes, Comments of Aristoteles' Book of Categories).

 $^{^{8}}$ C repetit ان هذا احسن فقد عرف $-^{9}$ F یکون $+^{9}$ F یقد $+^{10}$ $+^{10}$ $+^{10}$ $+^{10}$ $+^{10}$ $+^{10}$ $+^{10}$ $+^{10}$ Cو يكون j 'jam'. — الا j 'jam'. — الا j 'jam'. j 'jam'. j 'jam'. j 'jam'. j الا j 13 C,L,h,j يضعف F : يصعب 15 واحدة F : واحد C,L,h واحدة E : يصعب واحدة واحدة واحد E- وصعوبة F,L التشكك : التشكك - 17 Ita C,F,L,h

REFERENCES 37

file, it is difficult to find out where you are exactly in the source file, especially if the text is in Greek and has been transformed several times by Ω Translation Processes (from input encoding to Unicode and then to the 16-bit font encoding).

Another interesting example is the one on fig. 10: it is a critical edition of Averroes' Comments on Aristoteles' Book of Categories. There are four blocks of text, on the same page: the text of Averroes (from right to left), the critical apparatus (entries from left to right), the original Arabic text of Aristoteles (from right to left) and the critical apparatus on that text (also from left to right).

7. Conclusion

The issues covered in this paper may seem somehow unrelated. The author has tried to present some of the problems he had to solve in his daily typesetting work of the last few years. Their common point is the effort to respect by all possible means, and even re-invent in some cases, typographical tradition. At the same time the constraints of real life forced him to prefer practical solutions that work, to more elegant solutions that may have taken longer to realize. In any case, what counts is the result, and this is why this paper is full of examples, which we hope the reader has enjoyed.

The reader can find more information on past and ongoing projects, on the "research" page [14] of the Atelier's Web site.

References

- [1] *The Chicago Manual of Style*, 13th Edition, The University of Chicago Press, Chicago and London, 1982.
- [2] Duden-Taschenbuch Band 5, Satz- und Korrekturanweisungen, Richtlinien für die Texterfassung, Bibliographisches Institut, Mannheim, 1986.
- [3] Hart's Rules for Compositors and Readers at the University Press Oxford, 39th Edition, University Press, Oxford, 1996. (First edition: 1893)
- [4] Lexique des règles typographiques en usage à l'Imprimerie nationale, Imprimerie Nationale, Paris, 1990.

- [5] Robert Bringhurst, *The Elements of Typographic Style*, Hartley & Marks, Vancouver, 1992.
- [6] Pierre Duplan and Roger Jauneau, *Maquette et mise en page*, Éditions du Moniteur, Paris, 1994.
- [7] Eric Gill, An essay on typography, Lund Humphries, London, 1988 (reprint).
- [8] Bruce Rogers, *Paragraphs on Printing*, Wiliam E. Rudge's Sons, New York 1943 (reprint Dover, New York, 1979).
- [9] Martin Solomon, *The Art of Typography*, Watson-Guptill Publications, New York, 1986.
- [10] Jan Tschichold, *The New Typography, Handbook for Modern Designers*, University of California Press, Berkeley, 1995 (English translation).
- [11] Adrian Wilson, *The Design of Books*, Peregrine Smith, Salt Lake City, 1979.
- [12] Hans Peter Wilberg and Friedrich Forssman, *Lesetypographie*, Verlag Hermann Schmidt, Mainz, 1997.
- [13] http://www.linguistik.uni-erlangen.de/~orlorenz/DMM/DMM.html
- [14] http://www.fluxus-virus.com/en/research.html
- [15] http://www.fluxus-virus.com/en/books.html

Utilidades para la documentación de programas

Manuel Gutiérrez Algaba, Manuel Perera Domínguez

Facultad de Informática y Estadística Universidad de Sevilla España

e-mail: irmina@ctv.es

Resumen

Este trabajo presenta un conjunto de programas pensados para la elaboración de documentación, tanto de naturaleza textual como gráfica, durante el proceso de producción del software. Los programas generan como salida código LATEX y se han desarrollando utilizando el lenguaje python.

1. Introducción

El ciclo de vida clásico o modelo en cascada es el paradigma o marco común y más utilizado para el desarrollo de software. Este ciclo de vida parte de un enfoque secuencial en el que se consideran las fases de análisis, diseño, codificación, prueba y mantenimiento y constituye la base sobre la que se han propuesto diversos modelos alternativos y complementarios en ingeniería del software [3]. En la actualidad existen multitud de herramientas CASE que permiten una ingeniería del software asistida por ordenador de aspectos parciales de las diferentes fases del ciclo de vida, particularizando para las diversas metodologías y técnicas existentes.

Los aspectos de documentación han ido adquiriendo con el tiempo mayor importancia, sobre todo en las fases de análisis y diseño del ciclo de vida clásico. En la fase de codificación (traducción de una representación del software dada por un diseño detallado a una realización concreta en un lenguaje de programación) el desarrollo en este aspecto ha sido mucho menor y en gran medida se ha limitado a consejos de carácter práctico en cuanto al establecimiento de comentarios en el código fuente de los programas. En la actualidad se intenta interrelacionar documentación y código fuente de manera que se generen en paralelo de manera incremental. Los programas así desarrollados ganan en robustez e inteligibilidad y son más fáciles de probar y mantener. En esta línea se engloban la programación literaria [1] y el sistema CWEB [2].

En este trabajo comentamos una posible aplicación de LATEX y otros típicos programas de los sistemas UNIX para la documentación del proceso de desarrollo de software, proporcionando una respuesta potente y sencilla a las necesidades de documentación surgidas durante la elaboración de un proyecto fin de carrera en los estudios de Ingeniería Informática.

2. Descripción

Se tenía el problema de documentar un proyecto fin de carrera consistente en la elaboración de un programa simulador de un mundo virtual con técnicas de inteligencia artificial, utilizando la conocida metodología orientada a objetos OMT [5]. La documentación a generar integra tanto texto como los diversos tipos de diagramas que la metodología contempla (diagramas de clases y de estados, en nuestro caso). Como solución se optó por escribir un conjunto de utilidades independientes redactadas en el lenguaje python.

La complejidad estructural de la aplicación generada ocasionaba el problema de no saber exactamente a qué nivel había de estar la documentación que se estaba generando. Además, por la extensión de dicha documentación resultaba difícil trabajar con ella. Incluso algunas veces no se estaba seguro de la estructura de la documentación; se tenía claro que algunas cosas dependían o eran partes de otras, pero había partes que por sí solas podían constituir una unidad jerárquica superior o que podían combinarse con otras partes de la documentación. El desarrollo del programa sectioner permitió reducir bastante el problema. Trabajar con la documentación se convirtió en un proceso de seguir la pista a dos o tres pequeños ficheros, con la ventaja de que la disposición final de las distintas partes no hay que definirla hasta el final.

Uno de los problemas que siempre se encuentran con LATEX es la com-

plejidad que conlleva. Al apartarse un poco del estándar e intentar, por ejemplo, exponer una estructura o un dibujo de un diseño aparecen los problemas: ¿Qué paquete elegir? ¿Qué macros usar? En este caso concluimos que la solución satisfactoria más fácil era una de naturaleza intermedia. Usando elementos de LATEX, macros simples y corrientes, pero controlando sus parámetros desde un programa más flexible, se conseguía no sólo lo que se quería sino además un esquema ilimitado de posibilidades, en el cual LATEX dejaba de ser un programa con el que tratar directamente para convertirse una biblioteca de herramientas de formateo de textos. Siempre que el problema sea crear una serie de dibujos o esquemas muy parecidos se puede aprovechar este enfoque: LATEX en conjunción con un lenguaje de programación como python.

El número de clases de nuestro diagrama OMT era muy elevado. Dado que era esencial tener una documentación desde el principio y ante la posible necesidad de modificar los diagramas varias veces antes de alcanzar una configuración definitiva, se pensó recurrir a una automatización de la creación de los diagramas de clases para lo cual se desarrolló el programa consdiag.

3. Tratamiento de código L⁴T⊑X

En nuestro caso, las características más interesantes que ha de poseer un lenguaje de programación para tratar código LATEX son la orientación a objetos, la facilidad de trabajar con listas y las facilidades para expresiones regulares. Se realizó una comparación cualitativa de varios lenguajes de programación que se estimaron buenos candidatos para procesar código de la manera que nos interesaba. Como resultado optamos por el lenguaje python [4].

Característica	tcl	perl	python	С	Lisp
Listas	Bueno	Bueno	Excelente	Pésimo	Excelente
Objetos	Malo	Malo	Excelente	Malo	Bueno
Usabilidad	Bueno	Malo	Excelente	Malo	Regular
Exp. Regulares	Bueno	Excelente	Bueno	Malo	Bueno
Biblioteca	Bueno	Excelente	Bueno	Ilimitada	Excelente
Popularidad	Bueno	Excelente	Aceptable	Universal	Excelente

4. Programas desarrollados

El programa sectioner permite utilizar seccionamiento relativo. En vez de usar \section , por ejemplo, se indica el nivel del nuevo ámbito, respecto a los anteriores. Un ejemplo típico de código es el siguiente:

```
Tectónica de placas
\s3
\+
El periodo jurásico
... Texto 1 ...
Dorsales oceánicas
... Texto 2 ...
\c3
Corrientes oceánicas
... Texto 3 ...
/minput{mascodigotex.tex}
El código LATEX generado por sectioner es:
\section {Tectónica de placas}
\subsection {El periodo jurásico}
... Texto 1 ...
\subsection {Dorsales oceánicas}
... Texto 2 ...
\section {Corrientes oceánicas}
... Texto 3 ...
```

El programa boxerer produce rectángulos o cajas que representan la estructura típica de una función de programación (entradas, salidas, variables internas, ...) y cajas que muestran de manera cómoda y sencilla la estructura de un diseño de programación, a la manera del ejemplo de la figura 11.

El programa consdiag crea diagramas de Rumbaugh que representan clases y sus interrelaciones. Su utilidad deriva de la necesidad de hacer 50 ó 60 diagramas más o menos similares y de la «pereza» de modificar y crear manualmente con algún programa de dibujo todos esos diagramas de clases. En nuestra opción LATEX es simplemente una capa de software que se encarga de dibujar; de hecho consdiag podría adaptarse para generar código PostScript. Este programa pone más de manifiesto la faceta de

Lritaunas Peki	
Este programa pretende enseñar vocabulario y gramática de diferente	es
lenguajes.	
Representación gráfica	
GUI, principalmente	
tkinter.py	
Módulo principal	
men_prin.py Aquí se define el menú principal	
Estructuras de los lenguajes	
Configuración	
Base de datos de palabras	
Etiquetas	

Figura 11. Ejemplo de gráfico producido por boxerer.

«biblioteca gráfica» de TEX. La figura 12 es un ejemplo del tipo de diagramas generados.

Por último el programa grapher permite, utilizando gpic, crear diagramas de estados como el de la figura 13.

El uso de sectioner y boxerer facilita un método de trabajo incremental en el cual la documentación final puede generarse a la vez que se codifican y prueban los módulos funcionales de programa. No se trata exactamente de programación literaria, puesto que la documentación es independiente del código y además se ejecutan pequeños módulos. Los programas pueden encontrarse en el URL:

http://www.ctv.es/USERS/irmina/texpython.htm.

5. El futuro

Es conocido que T_EX, como producto en desarrollo, ha tocado techo. Sin embargo las necesidades de una gran parte de sus usuarios requieren un esquema más flexible. Y sobre todo existe la espada de Damocles de los procesadores de texto comerciales. Se impone la imperiosa de avanzar e intentar mejorar T_EX. Veamos dos posibilidades:

¿Merece la pena enriquecer LATEX con características propias de los lenguajes de programación? Se perdería la estabilidad que tiene ahora TEX y además un tiempo precioso (3, 4 años), con la amenaza constante del avance de los procesadores de texto comerciales. Por otro lado ya existen muy buenos lenguajes de extensión (tcl, python). Quizá el tiempo de transición fuera corto. Desde el punto técnico es un gran reto. En cualquier caso se produciría una división en el mundo TEX.

¿Es bueno «contaminar» CTAN con un número creciente de programas que no están escritos en TEX? Esta opción presenta fuertes ventajas a favor:

- Posibilitar el rápido desarrollo de multitud de utilidades.
- Facilitar el uso de T_EX y aprovechar todas sus potencialidades sin llegar a ser un T_EXnico.
- Permitir el procesamiento distribuido de documentos. Esto es factible con la tecnología actual. Básicamente se trata de que máquinas remotas traten con especificaciones que se mandan vía Internet. Esta tecnología posibilitaría que cualquier TEXero pudiera usar todas las aplicaciones sin llegar a poseerlas.

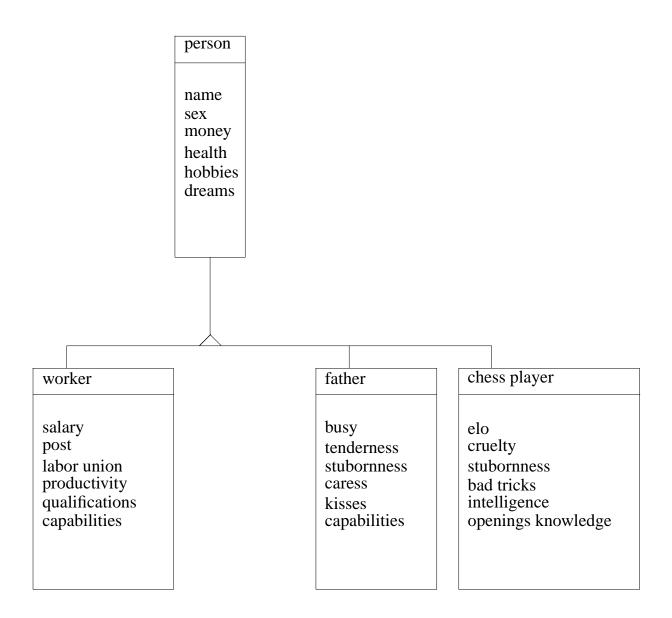


Figura 12. Ejemplo de diagrama de clases realizado con consdiag.

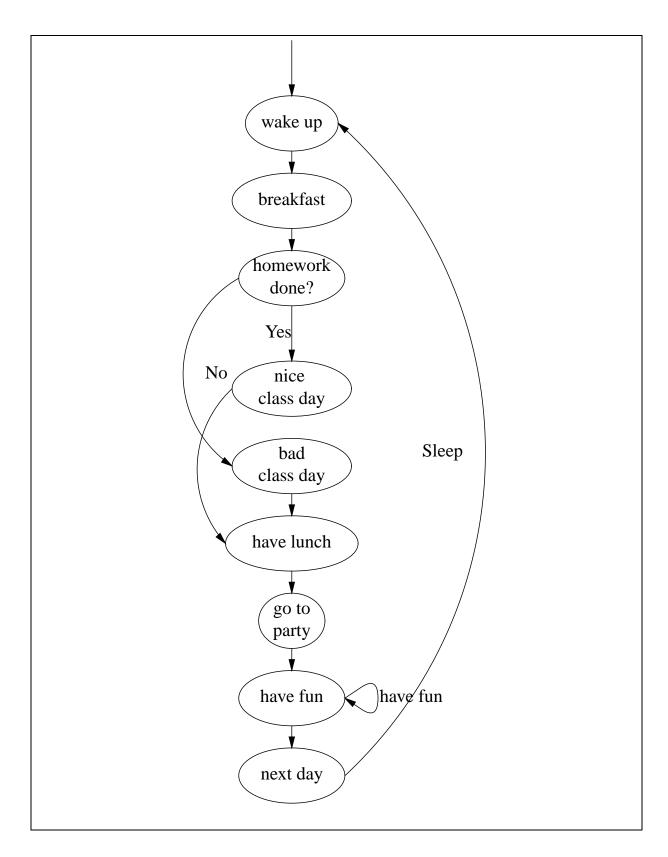


Figura 13. Ejemplo de diagrama de estados realizado con grapher.

REFERENCIAS 47

En contra presenta el problema de que se rompería en gran medida la universalidad de los documentos T_FX.

Referencias

- [1] D.E. KNUTH *Literate Programming*. Center for the Study of Language and Information (CSLI), 1992.
- [2] D.E. KNUTH *The CWEB System of Structured Documentation*. Addison-Wesley, 1993.
- [3] R.S. Pressman Ingeniería del software: un enfoque práctico. McGraw-Hill, 1997.
- [4] http://www.python.org
- [5] J. RUMBAUGH et al. Modelado y diseño orientados a objetos. Prentice-Hall, 1996.

xindy: un procesador de índices alternativo

Agustín Martín Domingo Departamento de Física ETS Arquitectura Madrid E-28040 Madrid email: agmartin@aq.upm.es

Resumen

xindy es un sistema de procesamiento de índices desarrollado por Roger Kehr [1], similar a makeindex, pero que se caracteriza por una serie de características adicionales, como internacionalización (puede procesar índices en muchos idiomas con distintos juegos de caracteres y reglas de ordenación) y un diseño modular altamente configurable. Se darán algunos ejemplos de uso haciendo especial hincapié en lo relativo al desarrollo de un estilo para el castellano.

1. Limitaciones de makeindex

makeindex es el programa clásico de indexación presente en todas las distribuciones LATEX, pero presenta algunas limitaciones para su uso en castellano y en general para los idiomas que no pueden describirse mediante la tabla de caracteres de 7 bits. Aunque a pesar de ello es posible utilizar makeindex en idiomas como el castellano, ello no se hace de forma natural, con todas las incomodidades que acarrea. Por ejemplo, para los acentos aunque no es difícil, desde luego no es inmediato,

\index{optica@óptica}
\index{indice@índice}

ordenaría la palabra índice como indice (sin acento) y la palabra óptica como optica, mientras que para la ñ sería algo más rebuscado y sería necesario algo como

```
\index{can^^ffa@caña}
```

para que caña se ordene adecuadamente. Si las palabras son compuestas la incomodidad crece de forma alarmante, por ejemplo, con

```
\index{optica electromagnetica@óptica electromagnética} \index{indice@indice!de refraccion@de refracción}
```

y no digamos ya si se quiere utilizar el orden alfabético tradicional en el castellano, en el que ch y 11 son consideradas como letras con personalidad propia y su sitio en el alfabeto, o diferenciar palabras como emphsólo y emphsolo.

Aunque se ha realizado un esfuerzo considerable en internacionalizar makeindex, [3, 4] lo cierto es que las versiones internacionalizadas de makeindex ni siquiera forman parte de la mayor parte de las distribuciones de LATEX y siguen presentando importantes carencias.

2. xindy, una posible solución

Algunas características de xindy

xindy es un sistema de indexación desarrollado por Roger Kehr [1], del Instituto de Informática Teórica de Darmstadt, que presenta una serie de peculiaridades,

Internacionalización xindy puede ser configurado fácilmente para adaptarse a las normas ortográficas de distintos idiomas, cada uno de ellos con un alfabeto distinto y con unas reglas de ordenación también distintas. xindy puede definir, para cada carácter del alfabeto, dónde debe ser éste ordenado (reglas de ordenación), así como a qué carácter o cadena es equivalente (reglas de equivalencia). Permite incluso definir reglas primarias y secundarias de ordenación que puedan resolver problemas como los presentados por las palabras *válido* y *valido*.

Formato de línea de índice definible por el usuario xindy permite definir páginas, rangos de páginas, secciones, o combinaciones de las mismas con gran facilidad, desde el clásico número de página a formatos del tipo enero-20, y de ordenarlos adecuadamente.

Altamente configurable y modular permite configurar los estilos de indexación para ser utilizado en distintos entornos de preparación de documentos, como TEX, SGML, u otros. Además permite reutilizar de forma fácil otros estilos ya hechos, además de los que ya vienen incluidos en la distribución.

xindy desde el punto de vista del usuario

Una diferencia a tener en cuenta es que el formato de entrada de xindy no es el normalmente creado por LATEX (con extensión .idx), sino un formato propio. Esto implica que es necesario procesar el fichero .idx creado por LATEX mediante un preprocesador. Este es tex2xindy y se usaría de la forma

```
tex2xindy fichero.sta < fichero.idx > fichero.raw
```

donde la extensión .sta corresponde a un fichero con datos del proceso, la extensión .idx al fichero de índice creado por LATEX y la extensión .raw al fichero de índice ya preprocesado que servirá de entrada a xindy, cuando se le llama de la forma

```
xindy -o fichero.ind -l fichero.ilg estilo.xdy fichero.raw
```

donde estilo.xdy es el estilo de creación de índices para xindy. Todo esto se habría podido hacer de una única vez llamando a tex2xindy como filtro en la llamada a xindy,

```
xindy -f tex2xindy estilo.xdy fichero.idx
```

Si ya está hecho el estilo adecuado, desde el punto de vista del usuario no hay más que saber, salvo que si el estilo lo contempla puede olvidarse de las complejas piruetas para introducir caracteres acentuados en los índices.

Si no es así, será necesario crear un estilo adecuado para xindy, de lo que trata la sección siguiente.

3. Creando un estilo para xindy

Definiendo alfabetos genéricos y el alfabeto base

Para evitar confusiones con la nomenclatura utilizada por xindy denominaremos alfabeto genérico a un vector de datos mientras que utilizaremos la palabra alfabeto base para describir el conjunto de caracteres base

susceptibles de ser ordenados y que forman el alfabeto del idioma en cuestión.

Se pueden definir alfabetos no ortográficos, por ejemplo con meses del año

```
(define-alphabet "meses"
  ("enero" "febrero" "marzo" "abril" "mayo" "junio"
  "julio" "agosto" "septiembre" "octubre" "noviembre"
  "diciembre"))
```

o el alfabeto digits que se encuentra definido por omisión,

En general no será necesario utilizar lo anterior salvo que se quiera crear un formato especial de línea de salida.

Lo que sí es necesario es definir el alfabeto base, salvo que sea suficiente el alfabeto base *latin* que cargaríamos con

```
(require "lang/latin/letgroup.xdy")
```

del fichero XINDY_BASE://lang/latin/letgroup.xdy. Si no es así, debemos definirlo. Por ejemplo, para utilizar el alfabeto base moderno del castellano, introduciríamos

```
(define-letter-groups
    ("a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m"
    "n" "ñ" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"))
```

De esta forma hemos definido el alfabeto base , con la $\|$ después de la $\|$ y antes de la letra o. Partiendo de un alfabeto base *latin*, también podríamos haber introducido la $\|$ mediante la línea

```
(define-letter-group "\tilde{n}" :after ''n'' :before "o")
```

que indica al procesador que las palabras comenzando por ñ forman un grupo con entidad propia, ordenadas después de las que empiezan por n y antes de las que empiezan por o. Las mismas reglas se aplicarían a las apariciones de la letra ñ en el interior de las palabras. Esto último permitiría una definición incremental de un alfabeto.

xindy no sólo permite introducir caracteres simples en un alfabeto, sino también caracteres compuestos que deben considerarse como letras únicas. por ejemplo, al alfabeto tradicional del castellano se definiría mediante

que diría al procesador de índices que los conjuntos de caracteres ch y 11 deben ser considerados como una única letra, y ordenados según se indica en la definición. Son necesarias algunas reglas adicionales para que el funcionamiento sea por completo el deseado.

Definiendo reglas de equivalencia y ordenación

Es también necesario tener en cuenta que hay caracteres que aun siendo distintos deben ordenarse como otros. Este es el caso por ejemplo de las mayúsculas, que deben ordenarse igual que si fueran minúsculas. Esta regla de equivalencia se definiría en xindy en la forma

```
(merge-rule "A" "a")
```

en el estilo. Esto se puede hacer también utilizando expresiones regulares, pero es bastante menos claro y eficiente que hacerlo de forma explícita. Estas definiciones también resultan útiles para que los caracteres acentuados escritos en formato TEX sean tratados de forma equivalente a los caracteres acentuados normales, como por ejemplo, en

```
(merge-rule "{\',\i}" "i" :string :again)
```

Definiendo el formato de las entradas en el índice

xindy también permite definir de forma fácil el formato de las líneas que han el índice, a las que denomina location-classes, por ejemplo

aceptaría entradas del tipo A-I, A-II o B-I que pudieran aparecer en un fichero de índice, y las ordenaría adecuadamente. Lo mismo se podría hacer si en vez de un alfabeto ordenado alfabéticamente como *ALPHA* se tuviera un alfabeto ordenado de otra forma como *meses*.

4. Un estilo para el castellano

En las normas de ordenación alfabética del castellano [5] en una primera aproximación no deben tenerse en cuenta ningún tipo de acentos, de forma

que las letras acentuadas sean equivalentes a sus correspondientes letras sin acentuar. Asimismo deben ignorarse los guiones y los apóstrofes. Estas reglas de equivalencia se expresarían mediante

```
(define-rule-set "es-basic"
        (("á" "a" :again :string)
 :rules
           ("é" "e" :again :string)
           ("í" "i" :again :string)
           ("ó" "o" :again :string)
           ("ú" "u" :again :string)
           ("ü" "u" :again :string)
           ("Á" "a" :again :string)
           ("É" "e" :again :string)
           ("Í" "i" :again :string)
           ("Ó" "o" :again :string)
           ("Ú" "u" :again :string)
           ("Ü" "u" :again :string)
           ("-" "" :again :string)
           ("'" "" :again :string)
           ))
```

la presencia de : again hace que se reordene todo después de la aplicación de estas reglas de equivalencia.

Estas reglas pueden dar lugar a palabras con letras acentuadas y no acentuadas que son equivalentes al tener las mismas letras base y cuya ordenación debe resolverse. La norma en este caso es que en segunda aproximación, para palabras equivalentes alfabéticamente en primera vuelta, las letras acentuadas deben ir detrás de las letras sin acentuar. Este comportamiento se conseguiría mediante

```
("o" "o0" :string)
("ó" "o1" :string)
("u" "u0" :string)
("ú" "u1" :string)
("ü" "u2" :string)
("A" "a0" :string)
("Á" "a1" :string)
("E" "e0" :string)
("É" "e1" :string)
("I" "i0" :string)
("Í" "i1" :string)
("0" "o0" :string)
("Ó" "o1" :string)
("U" "u0" :string)
("Ú" "u1" :string)
("Ü" "u2" :string)
("-" "~e" :string)
("'" "~e" :string)
))
```

Esto haría que las palabras acentuadas vayan detrás de las correspondientes sin acentuar, por ejemplo, que sólo vaya después que sólo, y que las que llenan guión o apóstrofe vayan detrás de las palabras equivalentes que no lo llevan, por ejemplo, que anti-virus vaya detrás de antivirus.

Además de estos caracteres es frecuente encontrar, especialmente en literatura científica, palabras con letras acentuadas que no existen en el castellano actual, provenientes de otros idiomas como el francés o el alemán. No existe una norma clara acerca de cuál debe ser el orden a seguir en estos casos, salvo el ya mencionado, en primera aproximación no se tienen en cuanta los discríticos, en segunda aproximación las letras acentuadas deben ordenarse después de las letras sin acentuar. Esto se conseguiría con la siguiente regla,

```
      (define-rule-set "es-extras"

      :rules (("à" "a" :again :string)

      ("ä" "a" :again :string)

      ("â" "a" :again :string)

      ("Ä" "a" :again :string)

      ("Â" "a" :again :string)

      ("Ç" "c" :again :string)

      ("Ç" "c" :again :string)

      ("B" "ss" :again :string)

      ("B" "ss" :again :string)
```

a añadir a la anterior es-basic. Como en los textos en castellano la letra ß alemana se escribe como se se las ha hecho equivalentes en cuanto a ordenación. Esto también dará lugar a palabras alfabéticamente equivalentes, cuyo orden deberemos resolver. No se ha encontrado una norma clara acerca de que tipo de acento predomina sobre cuál. Si nos limitamos a que estas letras estén después de las letras acentuadas que sí existen en el castellano, podríamos escribir una regla de equivalencia del tipo

("c" "c0" :string)

```
("ç" "c1" :string)
("C" "c0" :string)
("Ç" "c1" :string)
("s" "s " :string)
("ß" "ss" :string)
))
```

Definiendo el alfabeto y las etapas de ordenación

Definiremos ahora el alfabeto castellano y las características de cada etapa de ordenación, requiriendo las reglas anteriores que habremos incluido en el fichero es-rules.xdy. Como corresponde al alfabeto castellano moderno, lo he denominado es-moder.xdy, y básicamente sería

```
(define-letter-groups
    ("a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m"
     "n" "ñ" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"))
(require "lang/spanish/es-rules.xdy")
(define-sort-rule-orientations (forward forward forward))
;; Primera etapa
(use-rule-set :run 0
              :rule-set ("es-basic"
                         "es-extras"
                         "isolatin1-tolower"
                         ))
;; Segunda etapa
(use-rule-set :run 1
              :rule-set ("resuelve-acentos"
                         "resuelve-acentos-extras"
                         "isolatin1-tolower"
                         ))
;; Tercera etapa
(require "rules/latin-exchange.xdy")
(require "rules/isolatin1-exchange.xdy")
(use-rule-set :run 2
              :rule-set ("latin-exchange"
                         "isolatin1-exchange")
```

)

Aplicándolo a un documento LATEX

Hasta el momento sólo se han definido una seie de reglas genéricas de ordenación, pero no se ha dicho a xindy que su salida debe ser un archivo .ind compatible con LATEX y qué debe incluir. Para tener un mayor control de todo se puede mirar el archivo tex/makeidx.xdy, que daría un comportamiento similar al de makeindex y a partir de él crear un nuevo estilo con las características deseadas. Sin embargo, si se está a grandes rasgos conforme con el formato de salida y sólo se desea incluir las reglas del castellano, es más sencillo utilizar el carácter modular de los estilos de xindy y escribir algo como

```
(require "tex/makeidx.xdy")
(require "lang/spanish/fix-tex.xdy")
(require "tex/isolatin1m.xdy")
(require "tex/isolatin1s.xdy")
(require "lang/spanish/es-moder.xdy")
```

donde se incluye el estilo makeidx.xdy, las reglas de equivalencia de TEX a *latin1* y es-moder.xdy que incluiría las reglas que acabamos de definir. Añadiendo una línea

```
(markup-letter-group :open-head
    "~n {\bf " :close-head "}" :capitalize)
```

se conseguiría de forma sencilla que los distintos grupos de palabras estén separados por una letra.

5. Más información

```
http://www.iti.informatik.tu-darmstadt.de/xindy/CTAN\_MIRROR://indexing/xindy
Mis archivos para LinuX (libc5 y glibc2.1, no hay para glibc2.0):
http://www.aq.upm.es/~agmartin/linux/store/xindy-2.0.tgz (libc5)
http://www.aq.upm.es/~agmartin/linux/store/xindyg21-2.0.tgz
(glibc2.1)
```

Referencias

- [1] Roger Kehr, xindy A Flexible Indexing System, Proceedings of the EuroTeX'98, St. Malo, France, March 1998
- [2] Roger Kehr, xindy A Flexible Indexing System, Manual, tutorial y documento xindy by topic en la documentación de xindy.
- [3] Joachim Schrod, An International Version of MakeIndex, Cahiers GUTenberg, 10, p81-90, 1991
- [4] Joachim Schrod and Gabor Herr, *MakeIndex Version 3.0*, Technical University Darmstadt, 1991
- [5] José Martínez de Sousa, *Diccionario de lexicografía práctica*, Ed. Bibliograf S.A., Barcelona, 1995

T_EXemplares

Año 3.º, número 3, segundo trimestre 2002

Índice

- 3 Editorial
- 4 Tipografía española con T_EX Javier Bezos
- 17 Typesetting Books with Special Needs Yannis Haralambous
- 39 *Utilidades para la documentación de programas* Manuel Gutiérrez Algaba, Manuel Perera Domínguez
- 48 xindy: un procesador de índices alternativo Agustín Martín Domingo