

# Introduzione a X<sub>Y</sub>TEX

Massimiliano Dominici

## Sommario

Unicode e *smart font technologies* rappresentano l'attuale standard *de facto* nella tipografia digitale. Questo articolo mostra come è possibile, con X<sub>Y</sub>TEX, integrarli in un sistema di composizione tipografico basato su T<sub>E</sub>X.

## Abstract

Unicode and *smart font technologies* are the current *de facto* standard in digital typography. This article should explain how they can be incorporated, with X<sub>Y</sub>TEX, in a T<sub>E</sub>X-based typesetting system.

## 1 Introduzione

Per un sistema di composizione tipografica di alta qualità, quale è T<sub>E</sub>X, è necessario poter sfruttare appieno quelle tecnologie che si sono affermate come standard di fatto nel mondo della tipografia digitale. Nel corso della sua esistenza T<sub>E</sub>X ha dimostrato di sapersi adattare alle innovazioni che di volta in volta si sono imposte nell'uso comune. Nato in un contesto monolinguistico anglosassone e pensato, quindi, per risolvere esigenze legate a quel contesto; inizialmente in grado di ricevere solo testo codificato a 7-bit (ASCII); munito di un proprio formato particolare per il tracciamento dei caratteri (METAFONT); con il passare degli anni si è evoluto in modo da adattarsi ad un contesto multilinguistico, accettare testo codificato a 8-bit e usare quei formati di font (TYPE1 e TRUETYPE) divenuti di uso comune.

Questo adattamento è avvenuto senza stravolgere l'implementazione originaria; così, ad esempio, l'uso di font TYPE1 e TRUETYPE è possibile a patto che le metriche di tali font siano state preventivamente trasformate nel formato comprensibile da T<sub>E</sub>X (TFM, *T<sub>E</sub>X Font Metrics*), mentre la scelta della codifica del documento finale, o l'implementazione di funzioni tipografiche avanzate (quali le legature o i numeri "minuscoli") è effettuata per mezzo di "font virtuali" (VF, *Virtual Fonts*). A livello di immissione del testo, i vari tipi di codifica sono gestiti traducendo l'input nel formato interno di T<sub>E</sub>X.

Questa filosofia, se ha garantito un adattamento indolore e un alto grado di compatibilità all'indietro, ha però i suoi limiti e le sue controindicazioni. Limiti e controindicazioni che sono più evidenti oggi che gli standard di fatto nel mondo della tipografia

digitale si chiamano Unicode e *smart font technologies*.

Unicode<sup>1</sup> è l'implementazione ufficiale dello standard ISO/IEC 10646 per la codifica dei caratteri appartenenti ai sistemi di scrittura di tutto il mondo. Ad ogni carattere "astratto"<sup>2</sup> Unicode attribuisce un codice numerico e un nome standard, evitando così il ricorso ad una pletora di codifiche parziali, spesso in conflitto l'una con l'altra (ad esempio "Windows cp1252" and "ISO/IEC 8859-1" che pretendono entrambe di rappresentare una codifica per "Latin 1"). Il suo uso facilita l'interscambio di dati fra diverse applicazioni fornendo uno standard condiviso per la codifica di ogni carattere.

Gli *smart font* ("font intelligenti"), sono quei formati di font in grado di istruire l'applicazione che vi accede (e che è in grado di interpretarne le informazioni) su come applicare al testo in entrata, in maniera automatica, una serie di funzioni tipografiche avanzate. Queste funzioni avanzate possono andare dalla semplice sostituzione di uno o più "glifi" con legature o forme storiche, in maniera sensibile al contesto oppure no, fino a complesse operazioni di posizionamento, tracciamento e riordino dei glifi rispetto a quelli che li precedono o seguono, operazioni necessarie per visualizzare correttamente alcuni sistemi di scrittura non occidentali.

I più comuni esempi di *smart font technologies* sono Apple Advanced Typograhly (AAT), sviluppato da Apple, OPENTYPE, sviluppato congiuntamente da Adobe e Microsoft, e Graphite, sviluppato da SIL International.

Nonostante la tecnologia realizzata da Apple sia stata l'antesignana delle *smart font technologies* e rimanga tuttora superiore in alcuni settori, come quello dei *complex scripts*, il supporto di Adobe e Microsoft, e alcuni innegabili fattori di superiorità come quello di poter immagazzinare indifferentemente i contorni dei glifi sia nel formato tipico dei TRUETYPE, che in quello, preferito dai tipografi professionali, dei TYPE1<sup>3</sup> hanno decretato il successo del formato OPENTYPE. A partire dal 1999, con la decisione della stessa Adobe di convertire l'intera propria collezione di font nel nuovo formato OPENTYPE, questo ha progressivamente soppiantato

1. <http://www.unicode.org>

2. E non alla sua "presentazione": "A" e "A" sono due caratteri distinti per Unicode; non così "A" e "A": entrambi rappresentano il carattere "U+0041 LATIN CAPITAL LETTER A".

3. In realtà, i dati relativi al contorno dei glifi, nel caso di OPENTYPE di tipo POSTSCRIPT, sono contenuti nel formato CFF (*Compact Font Format*), che è, in sostanza, una forma compressa del TYPE1.

tato sia il TYPE1 che il TRUETYPE, affermandosi come nuovo standard di fatto.<sup>4</sup>

Sia l'uso di Unicode come codifica del testo in ingresso, sia l'uso e l'inclusione di font OPENTYPE o AAT è possibile con T<sub>E</sub>X, ma non in maniera nativa. La codifica Unicode in ingresso deve essere "tradotta" da appositi moduli (ne esistono per i principali sistemi di composizione basati su T<sub>E</sub>X: L<sup>A</sup>T<sub>E</sub>X e ConT<sub>E</sub>Xt), mentre l'accesso ai font deve essere effettuato tramite TFM e VF, rendendo molto complicata la loro installazione e perdendo per strada alcune caratteristiche (ad esempio l'unitarietà del font: T<sub>E</sub>X è costretto, dalla limitazione interna dei TFM ad un massimo di 256 caratteri, a considerare ogni font virtuale come un font separato, anche se i singoli glifi sono immagazzinati nello stesso contenitore).

A questi inconvenienti si propone di porre rimedio X<sub>Y</sub>T<sub>E</sub>X, una reimplementazione di T<sub>E</sub>X sviluppata da Johnatan Kew, della SIL International, con lo scopo di facilitare la composizione di documenti in sistemi di scrittura non occidentali, e originariamente disponibile solo per sistemi Macintosh.<sup>5</sup> Le innovazioni che X<sub>Y</sub>T<sub>E</sub>X introduce si concentrano essenzialmente su tre punti: a) supporto nativo per Unicode; b) accesso diretto ai font installati nel sistema operativo, senza l'intermediazione dei TFM; c) capacità di interpretare direttamente le istruzioni contenute nelle tabelle interne dei font AAT, OPENTYPE o Graphite. Un effetto collaterale, dovuto al modo in cui vengono passate al driver di stampa le informazioni relative ai font, è che non è più possibile produrre un output in DVI, ma solo in PDF (vedi sezione 2.3).

In questo modo l'uso, da parte dell'utente, di nuovi font residenti nel sistema operativo viene reso estremamente semplice, così come estremamente semplice diventa l'accesso ad eventuali funzioni tipografiche avanzate presenti nel font stesso.

## 2 Nozioni di base

In questa sezione verrà esposto, in linea di principio, il funzionamento di comandi di basso livello che l'utente di solito non ha bisogno di usare nei propri documenti, ma il cui esame può essere utile per comprendere il meccanismo con cui X<sub>Y</sub>T<sub>E</sub>X opera.

L'utente, viceversa, ha a disposizione, per attivare le funzionalità avanzate di X<sub>Y</sub>T<sub>E</sub>X, macro di alto livello, definite in appositi moduli, per quanto riguarda L<sup>A</sup>T<sub>E</sub>X, o all'interno del sistema stesso, per quanto riguarda ConT<sub>E</sub>Xt. Questi costrutti di alto livello saranno passati in rassegna nella sezione 3.

4. Non tutti, però, condividono il giudizio entusiastico largamente diffuso sui nuovi *smart font*. Per un autorevole parere contrario, si veda lo "sfogo" di Luc Devroye in <http://cg.scs.carleton.ca/~luc/opentyperant.html>.

5. Attualmente esistono versioni sia per Linux che per Windows, incluse nelle maggiori distribuzioni come T<sub>E</sub>X Live e MiK<sub>T</sub>E<sub>X</sub>.

Per maggiori dettagli su quanto verrà esaminato in questa sezione si può consultare KEW (2007).

### 2.1 Codifica del testo in ingresso

X<sub>Y</sub>T<sub>E</sub>X è stato progettato in modo tale da poter interpretare un flusso di dati codificato indifferentemente in UTF-8 o UTF-16 (i due sistemi di codifica Unicode comunemente usati). Non è necessario, quindi ricorrere a pacchetti, moduli aggiuntivi o comandi particolari per compilare un testo espresso in questa codifica, a differenza di quanto avviene per L<sup>A</sup>T<sub>E</sub>X, dove è necessario caricare un modulo di `inputenc` o per ConT<sub>E</sub>Xt, dove si deve passare al programma l'istruzione `\enableregime[utf-8]`.

Al contrario, bisogna segnalare a X<sub>Y</sub>T<sub>E</sub>X se è necessario tradurre in Unicode da una codifica diversa. Le primitive `\XeTeXinputencoding` e `\XeTeXdefaultencoding` indicano a X<sub>Y</sub>T<sub>E</sub>X di operare questa traduzione, rispettivamente, per un singolo file e per tutti i file che verranno letti durante la compilazione, e garantiscono così compatibilità con file preparati con sistemi tradizionali.

È il caso di anticipare qui una caratteristica che logicamente dovrebbe apparire nella sezione 2.2, in quanto si tratta di un'opzione che può essere specificata al momento della selezione di un font. Tuttavia, trattandosi di un'opzione che ha effetto sul metodo di input, è bene esaminarla in questa sezione. L'opzione in questione specifica come tradurre in codifica Unicode alcuni costrutti particolari. In pratica viene usata generalmente (`map=tex-text`) per tradurre legature caratteristiche dell'input T<sub>E</sub>X: "--" → "–", "!' " → "¡", ecc. La traduzione viene fatta tramite un file `.tec` che è il risultato di una compilazione, a partire da un file ASCII `.map`, con TECKit, un programma sviluppato da SIL International<sup>6</sup> per creare, appunto, tabelle di conversione da codifiche obsolete verso Unicode. Questo meccanismo può essere usato anche per "effetti speciali", come la traslitterazione in alfabeto latino di un testo in cirillico.

### 2.2 Accesso ai font

Per poter accedere ai font residenti nel sistema e poter sfruttare le tabelle interne degli *smart font*, la primitiva `\font` ha dovuto essere reimplementata e la sua sintassi estesa in modo da accettare argomenti opzionali.

Mentre la sintassi originale di `\font` è essenzialmente la seguente (si veda KNUTH, 1984, p. 276):

```
\font⟨comando⟩=⟨nome font⟩⟨dimensioni⟩
```

dove `⟨dimensioni⟩` può indicare sia un corpo (come in `at 12pt`) o un fattore di scala (come in `scaled 500`), nella nuova sintassi estesa è possibile

6. Il programma è disponibile, sotto forma di sorgenti compilabili, dall'archivio delle versioni di X<sub>Y</sub>T<sub>E</sub>X.

specificare una serie di opzioni relative alle caratteristiche generali di un font e a quali delle sue funzioni tipografiche avanzate si vuole eventualmente far ricorso:

```
\font<comando>=<nome font>/<opzioni>:
  <caratteristiche avanzate><dimensioni>
```

Innanzitutto  $\langle$ nome font $\rangle$  accetta una sintassi leggermente diversa dall'omonimo argomento del comando tradizionale. Nelle sue prime implementazioni su sistemi Macintosh,  $\text{\X}\text{\TeX}$  era in grado di accedere *solo* ai font installati nel sistema operativo.  $\langle$ nome font $\rangle$  doveva quindi specificare il nome del font, così come risulta al sistema operativo (ad esempio: "Times Roman"). Nelle implementazioni successive, e solo nel caso in cui si usi  $\text{\x}\text{\dvipdfmx}$ <sup>7</sup> per generare il PDF finale, la sintassi è stata estesa per dar modo a  $\text{\X}\text{\TeX}$  di accedere ai font ovunque si trovino. Racchiudendo il nome del font tra parentesi quadrate è possibile adesso indicare a  $\text{\X}\text{\TeX}$  di cercarlo all'interno della TDS,<sup>8</sup> oppure, sempre all'interno delle parentesi quadrate, è possibile specificare l'indirizzo preciso del font.

Tramite  $\langle$ opzioni $\rangle$  è possibile specificare alcune caratteristiche generali del font: variante (/I; corsivo, /B: neretto, ecc.), taglia ottica (/S= $x$ , dove  $x$  indica il corpo),<sup>9</sup> specifica *smart font technology* da utilizzare (/ATSUI per AAT, /ICU per OPEN $\text{\T}$ YPE e /GR per Graphite).<sup>10</sup> Ogni  $\langle$ opzione $\rangle$ , identificata da uno specifico 'tag', deve essere preceduta dal simbolo "/".

Per attivare (o disattivare) specifiche funzioni tipografiche avanzate, l'utente può passare a  $\text{\X}\text{\TeX}$  una lista di elementi preceduti dal simbolo ":". La sintassi dei singoli elementi dipende dal tipo di *smart font technology* implementata nel font. Nel caso di font AAT, ad ogni funzione avanzata viene assegnato un nome dallo sviluppatore, cosicché l'utente dovrà specificare una coppia "chiave = valore" (per esempio: "Letter Case = Small Caps"). Nel caso di font OPEN $\text{\T}$ YPE, invece, le funzioni avanzate sono identificate da codici convenzionali di quattro

7. Si veda la sezione 2.3.

8. *TeX Directory Structure*, è l'insieme di cartelle in cui è distribuita un'installazione standard di  $\text{\T}\text{\E}\text{\X}$ .

9. Alcuni font sono distribuiti in versioni differenti per specifiche gamme di corpi, in modo da evitare variazioni nel 'colore' della pagina (l'idea, ripresa dalla tipografia tradizionale, in cui un tipo di carattere ha un disegno leggermente diverso per ogni corpo, è stata sfruttata anche da Knuth nel disegnare il font *Computer Modern*). Ciascuna di queste versioni ha un suffisso proprio (ad esempio *Caption* o *Subhead*) ed è associata internamente ad una gamma di corpi. Le applicazioni in grado di avvalersi delle *smart font technologies*, tra cui  $\text{\X}\text{\TeX}$ , scelgono automaticamente la versione più appropriata per il corpo selezionato, cosicché non è necessario specificare tale opzione se non per sovrascrivere il comportamento di default.

10. Di solito non è necessario specificare quest'ultima opzione, in quanto  $\text{\X}\text{\TeX}$  è in grado di riconoscere da sé il formato del font. Tuttavia, alcuni font possono contenere tabelle in più di un formato e l'utente può volerne attivare uno in particolare, diverso da quello di default.

lettere ('tag'): all'utente basta quindi specificare uno di questi codici preceduto dal simbolo "+", se vuole attivare la funzione, o dal simbolo "-", se vuole disattivarla. Gli OPEN $\text{\T}$ YPE prevedono anche due funzioni speciali: *script* e *language*, tramite le quali è possibile selezionare il supporto per un sistema di scrittura o per una lingua particolari. In questo caso è necessario far seguire al nome della funzione il 'tag' relativo al sistema di scrittura o alla lingua che si desiderano attivare, con il consueto sistema "chiave = valore".

Poiché sia il  $\langle$ nome font $\rangle$  che il nome delle funzioni tipografiche avanzate in font AAT possono contenere spazi bianchi al loro interno, è consigliabile racchiudere l'intera stringa rappresentante l'argomento di  $\backslash$ font (tranne  $\langle$ dimensioni $\rangle$ ) tra apici doppi.

Per concludere, un paio di esempi serviranno a chiarire come è possibile usare questa sintassi estesa del comando  $\backslash$ font.

```
\font\hoefler="Hoefler Text/B:Letter Case%
=Small Caps" at 12pt
```

Al comando  $\backslash$ hoefler viene associata la selezione del font "Hoefler Text", installato nel sistema operativo, nella variante "neretto", con sostituzione del mauscoletto alle lettere minuscole, in corpo 12.<sup>11</sup>

```
\font\juni="[/percorso/Junicode_Regular]%
:script=latn:language=ISL:+hlig:+hist"%
at 11pt
```

Il comando  $\backslash$ juni seleziona il font "Junicode Regular" (non installato nel sistema operativo, né nella TDS) in corpo 11, utilizzando il supporto specifico per l'alfabeto latino e la lingua islandese, e attivando forme e legature storiche.

### 2.3 Formato dell'output

Il formato ultimo dei documenti compilati con  $\text{\X}\text{\TeX}$  è un PDF. Tuttavia  $\text{\X}\text{\TeX}$  usa, nel corso del processo di compilazione, uno speciale formato intermedio (XDV, "extended DVI") allo scopo di trasmettere al driver di stampa<sup>12</sup> le informazioni necessarie ad interpretare tutto ciò che riguarda le caratteristiche dei font da includere nel documento finale. Questo formato, che non può essere interpretato dagli attuali visualizzatori di DVI, né correttamente disassemblato da programmi come *dvitype* o *dv2dt*, viene successivamente "distillato" nel PDF finale. Tutto ciò avviene in maniera del tutto automatica e trasparente per l'utente, che non deve compiere azioni aggiuntive; anzi, se vuole disattivare, nelle compilazioni intermedie, la generazione del PDF (che è la parte più dispendiosa

11. L'esempio è tratto da ROBERTSON (2007a).

12. Il termine "driver di stampa" non è forse del tutto corretto, in quanto l'output finale è un file.

del processo, in termini di tempo), deve passare al compilatore `xetex` l'opzione `--no-pdf`.

X<sub>Y</sub>T<sub>E</sub>X è in grado di usare due diversi driver di stampa, per ottenere il PDF finale. `xdv2pdf` è quello tradizionale, implementato nella versione originaria di X<sub>Y</sub>T<sub>E</sub>X, ed è quello ancora abilitato di default su computer Macintosh. Ormai, però, le versioni per Linux e Windows utilizzano `xdvipdfmx`, una versione appositamente estesa di `dvipdfmx`, che, a sua volta, era la reimplementazione di `dvipdfm`. Tra i due driver ci sono alcune differenze, la più notevole è il supporto per i font non installati nel sistema operativo (vedi sezione 2.2), presente in `xdvipdfmx` e assente in `xdv2pdf`. Lo stesso vale per la possibilità di interpretare gli `\specials` inseriti da `PSTricks` (vedi sezione 4.2.3).

### 3 Uso con L<sup>A</sup>T<sub>E</sub>X e ConT<sub>E</sub>Xt

Per quanto l'uso di T<sub>E</sub>X nella sua versione più "pura" (Plain T<sub>E</sub>X) sia ancora diffuso, tuttavia è certamente maggiore il numero di coloro che usano T<sub>E</sub>X attraverso "macro pacchetti" come L<sup>A</sup>T<sub>E</sub>X e ConT<sub>E</sub>Xt. Questi utenti sono abituati ad usare un linguaggio di *markup* più astratto rispetto alle istruzioni di formattazione di basso livello tipiche di T<sub>E</sub>X. Per essi è poco conveniente dover ricorrere ad espressioni come quelle esaminate nella sezione 2.2, ogni volta che è necessario selezionare un determinato tipo di carattere o una sua specifica caratteristica. Si aspettano che ciò possa essere fatto per mezzo di dichiarazioni nel preambolo, per istruzioni generali, o per mezzo di specifici comandi di *markup* da applicare a singole porzioni di testo. Questa aspettativa è stata soddisfatta sotto forma di specifiche estensioni ("pacchetti") per quanto riguarda L<sup>A</sup>T<sub>E</sub>X, o di nuove funzionalità aggiunte al nocciolo, per quanto riguarda ConT<sub>E</sub>Xt.<sup>13</sup>

#### 3.1 L<sup>A</sup>T<sub>E</sub>X

##### 3.1.1 *Fontspec*

La principale estensione che mette l'utente di L<sup>A</sup>T<sub>E</sub>X in grado di comporre i propri documenti con X<sub>Y</sub>T<sub>E</sub>X è `fontspec`. Questo pacchetto, scritto da Will Robertson (ROBERTSON, 2007b), mette a disposizione dell'utente una serie di istruzioni per una coerente selezione dei font e delle funzioni tipografiche avanzate ad essi associate. I comandi principali definiti nel pacchetto sono:

```
\setmainfont[{opzioni}]{{nome font}}
\setsansfont[{opzioni}]{{nome font}}
\setmonofont[{opzioni}]{{nome font}}
\fontspec[{opzioni}]{{nome font}}
```

Hanno tutti la medesima sintassi, per cui, nel seguito, gli esempi riguarderanno solo `\fontspec`,

13. Questo diverso tipo di approccio, modulare nel caso di L<sup>A</sup>T<sub>E</sub>X e monolitico nel caso di ConT<sub>E</sub>Xt, rappresenta una delle differenze basilari tra i due programmi.

```
\fontspec[ExternalLocation,\
LetterSpace=8,WordSpace=2,\
Color=AA4444]{texgyreschola-regular}
\begin{center}
{\Large I PROMESSI SPOSI}\[1ex]
ALESSANDRO MANZONI
\end{center}
```

## I PROMESSI SPOSI

### ALESSANDRO MANZONI

FIGURA 1: Maiuscolo spaziato

che è l'istruzione da utilizzare in mezzo al documento, mentre gli altri tre comandi riguardano la dichiarazione, nel preambolo, del font principale e del font senza grazie o a spaziatura fissa che eventualmente lo accompagnano.

L'argomento *<nome font>* può essere sia il nome interno del font ("Times Roman", "Linux Libertine O", ecc.), nel caso che questo sia installato nel sistema operativo, oppure il nome del file che contiene il font, se questo non è installato. In questo secondo caso, che come abbiamo visto nella sezione 2.3 funziona solo se si usa `xdvipdfmx` come driver di stampa PDF, è necessario specificare tra le *<opzioni>* `ExternalLocation`, eventualmente seguito dal percorso, se il font non si trova nella TDS.

Le *<opzioni>* si dividono sostanzialmente in due categorie: quelle che possono essere applicate a ciascun font indifferentemente (colore, spaziatura tra le lettere o tra le parole, ecc.) e quelle, invece, che dipendono da ciò che si trova (o non si trova) nelle tabelle interne al font, nel caso questo disponga di funzioni tipografiche avanzate. Naturalmente non verranno qui esaminate tutte (si rimanda il lettore alla documentazione del pacchetto per un'analisi dettagliata), ma verranno mostrati solo alcuni esempi che si ritengono significativi.

Nella figura 1 è possibile osservare un esempio di maiuscolo spaziato. Questa pratica, di uso comune quando si voglia rendere esteticamente più gradevole l'aspetto di un titolo, non aveva, fino a non molto tempo fa<sup>14</sup> soluzioni realmente soddisfacenti. Come si vede dal codice che precede l'esempio, con `fontspec` è sufficiente passare ad uno dei comandi per la selezione dei font l'opzione `LetterSpace`, impostata al valore *x*, dove *x* rappresenta la dimensione dello spazio aggiunto, in frazioni percentuali del corpo del font. Per migliorare la resa visiva, è preferibile usare, contestualmente, l'opzione `WordSpace`, che moltiplica lo spazio normalmente inserito da T<sub>E</sub>X tra due parole per il valore fornito dall'utente (in questo caso '2', raddoppiando quindi uno spazio normale). In realtà `WordSpace` accetta come valore una tripla

14. Finché, cioè *microtype* non ha reso accessibili all'utente medio di L<sup>A</sup>T<sub>E</sub>X le funzioni di microtipografia implementate in `pdftex`. Il pacchetto `soul` forniva una soluzione limitata e poco robusta.

PSALM LXVIII. 20, *in fine*.

*And unto God the Lord belong the issues of death (i.e. from death).*

Buildings stand by the benefit of their foundations that sustain and support them, and of their buttresses that comprehend and embrace them, and of their contignations that knit and unite them. The foundations suffer them not to sink, the buttresses suffer them not to swerve, and the contignation and knitting suffers them not to cleave. The body of our building is in the former part of this verse. It is this: *He that is our God is the God of salvation; ad salutes*, of salvations in the plural, so it is in the original; the God that gives us spiritual and temporal salvation too.

(a) Numeri ‘oldstyle’

PSALM LXVIII. 20, *in fine*.

*And unto God the Lord belong the issues of death (i.e. from death).*

Buildings stand by the benefit of their foundations that sustain and support them, and of their buttresses that comprehend and embrace them, and of their contignations that knit and unite them. The foundations suffer them not to sink, the buttresses suffer them not to swerve, and the contignation and knitting suffers them not to cleave. The body of our building is in the former part of this verse. It is this: *He that is our God is the God of salvation; ad salutes*, of salvations in the plural, so it is in the original; the God that gives us spiritual and temporal salvation too.

(c) Forme storiche

PSALM LXVIII. 20, *in fine*.

*And unto God the Lord belong the issues of death (i.e. from death).*

Buildings stand by the benefit of their foundations that sustain and support them, and of their buttresses that comprehend and embrace them, and of their contignations that knit and unite them. The foundations suffer them not to sink, the buttresses suffer them not to swerve, and the contignation and knitting suffers them not to cleave. The body of our building is in the former part of this verse. It is this: *He that is our God is the God of salvation; ad salutes*, of salvations in the plural, so it is in the original; the God that gives us spiritual and temporal salvation too.

(b) Legature non comuni

PSALM LXVIII. 20, *in fine*.

*And unto God the Lord belong the issues of death (i.e. from death).*

BUILDINGS STAND BY THE BENEFIT OF THEIR FOUNDATIONS THAT SUSTAIN AND SUPPORT THEM, AND OF THEIR BUTTRESSES THAT COMPREHEND AND EMBRACE THEM, AND OF THEIR CONTIGNATIONS THAT KNIT AND UNITE THEM. THE FOUNDATIONS SUFFER THEM NOT TO SINK, THE BUTTRESSES SUFFER THEM NOT TO SWERVE, AND THE CONTIGNATION AND KNITTING SUFFERS THEM NOT TO CLEAVE. THE BODY OF OUR BUILDING IS IN THE FORMER PART OF THIS VERSE. IT IS THIS: *He that is our God is the God of salvation; ad salutes*, OF SALVATIONS IN THE PLURAL, SO IT IS IN THE ORIGINAL; THE GOD THAT GIVES US SPIRITUAL AND TEMPORAL SALVATION TOO.

(d) Maiuscoletto

FIGURA 2: Funzioni tipografiche avanzate per il font *Junicode*

di numeri, separati da virgola: i tre valori verranno applicati, come fattore di scala, rispettivamente, al valore nominale dello spazio, al suo ‘allungamento’ massimo (*stretch*) e al suo ‘accorciamento’ massimo (*shrink*). Specificare un solo valore significa assegnare lo stesso a ciascuno dei tre elementi della tripla. Abbiamo infine aggiunto un tocco di colore (che nella versione stampata apparirà in una diversa tonalità di grigio) con l’opzione `Color`, seguita da una tripla di valori RGB (ogni valore è formato da due cifre espresse nel sistema esadecimale). È possibile aggiungere anche un quarto valore per indicare il grado di trasparenza.

Negli esempi seguenti (riassunti nella figura 2<sup>15</sup>) verrà utilizzato il font *Junicode* per esaminare le opzioni relative alle funzioni tipografiche avanzate contenute nelle tabelle `OPENTYPE`.

Nell’esempio della figura 2a i numeri normali sono stati sostituiti da numeri ‘oldstyle’, tramite l’opzione `Numbers=Oldstyle`. `Numbers` accetta come valori possibili qualunque combinazione tra

15. Dove sono riprodotte le prime righe di *Death’s Duel*, di John Donne, come appaiono nella versione E-Text di Project Gutenberg dell’originale `DONNE` (1959).

`Lining`, `Oldstyle`, `Monospaced` e `Proportional` (‘normali’, ‘oldstyle’, a spaziatura fissa, a spaziatura variabile). Solo per font `OPENTYPE` è possibile specificare anche il valore `SlashedZero`, che sostituisce lo zero normale con uno zero barrato.

Nell’esempio seguente (figura 2b) sono state aggiunte alcune legature opzionali,<sup>16</sup> aggiungendo all’argomento opzionale di `fontspec` la stringa `Ligatures=Discretionary`. Si può notare, infatti, che la forma di `st`, nella parola ‘stand’ di riga 4 è costituita da un’unico carattere. Se si esaminano attentamente i quattro esempi, poi, ci si accorge che il numero romano ‘LXVIII’ ha una forma diversa, più compatta, negli esempi delle figure 2b e 2c: anche in questo caso si tratta, per l’appunto, di una legatura. Non è qui il caso di riportare tutti i possibili valori dell’opzione `Ligatures`; oltre ad essere molti, infatti, variano anche a seconda che si abbia a che fare con un font AAT oppure `OPENTYPE`. Si rimanda, quindi, per ulteriori dettagli, al manuale di `fontspec` già citato (ROBERTSON,

16. Legature comuni come `fi`, ecc. sono attivate automaticamente, ma possono eventualmente essere disattivate tramite l’opzione `Ligatures=NoCommon`.

2007b).

Nella figura 2c è stata aggiunta un'ulteriore funzione tipografica avanzata: alcuni caratteri sono stati sostituiti con forme arcaiche degli stessi (in particolare è stata sostituita **s** con **slong** e tutte le sue legature). L'opzione da aggiungere, in questo caso, è: `Style=Historic`. Anche in questo caso si rimanda al manuale di `fontspec` per informazioni più dettagliate, dal momento che l'opzione accetta un lungo elenco di valori predefiniti, diversi a seconda che il font incorpori la tecnologia AAT o OPENTYPE.

Infine, nell'esempio riportato nella figura 2d, sono state sostituite le lettere dell'alfabeto minuscolo con le rispettive forme del maiuscoletto. È da notare che il testo in corsivo è rimasto immutato: infatti la versione corsiva di Junicodè non possiede un maiuscoletto e di conseguenza l'opzione è stata in questo caso ignorata.<sup>17</sup> `Letters=Smallcaps` è quanto serve per attivare la suddetta funzione. Altri possibili valori per `Letters` sono `Uppercase`, che trasforma tutto il testo in maiuscolo, `Lowercase`, che trasforma tutto in minuscolo, e `UppercaseSmallCaps` che opera nella direzione opposta a `SmallCaps`, trasformando in maiuscoletto le maiuscole. I font AAT possono anche avere una funzione `InitialCaps` per trasformare in maiuscolo tutte le lettere a inizio parola.

È possibile anche aggiungere o togliere determinate caratteristiche al font in uso al momento, usando il comando `\addfontfeature`, il cui unico argomento accetta la stessa sintassi dell'argomento opzionale di `\fontspec`. Inoltre, con `\newfontfamily<comando>[<opzioni>]{<font>}`, o con `\newfontface`, che accetta la stessa sintassi, è possibile creare degli appositi comandi per la selezione rapida e coerente di caratteri per contesti specifici.

Infine `fontspec` fornisce una serie di comandi per definire nuove funzioni tipografiche avanzate che non sono coperte dal pacchetto stesso, pur essendo presenti nel font. `\newAATfeature`, `\newICUfeature` e `\newfontfeature` servono, appunto, a questo scopo.

### 3.1.2 Polyglossia

XƎTEX è stato ideato per lavorare con font OPENTYPE, AAT o Graphite, in codifica Unicode. Questi font dispongono di solito di un insieme di caratteri che copre numerosi linguaggi e sistemi di scrittura. È quindi naturale considerarlo un programma particolarmente adatto ad operare in un contesto multilinguistico. Peccato, quindi, che esistano profondi conflitti tra XƎTEX e il pacchetto `babel`, che è il pilastro portante del supporto multilinguistico per LATEX. Un conflitto profondo a tal punto da

17. Ma di questa mancata applicazione rimane traccia nel file di log, sotto forma di avviso da parte del pacchetto `fontspec`.

rendere inutilizzabile l'accoppiata XƎTEX+`babel` per ambiti linguistici quali il greco e l'ebraico. Ciò avviene perché, nel momento in cui fu ideato e sviluppato, `babel` aveva bisogno di fare supposizioni a proposito della codifica e dei font utilizzati, supposizioni che intralciano sostanzialmente il funzionamento di XƎTEX.

Esistono dei modi per aggirare queste limitazioni, ma tutti portano ad un uso subottimale di `babel`. Ad esempio, se si vuole utilizzare il modulo di `babel` per il greco con XƎTEX, è possibile caricarlo come non predefinito (basta che non appaia come ultimo nella lista dei moduli caricati) e usarne solo le funzionalità relative alla corretta sillabazione, racchiudendo il testo greco tra `\begin{hyphenrules}{greek}` e `\end{hyphenrules}`. In questo modo, però, si perdono, per esempio, tutti i nomi predefiniti per i vari elementi del testo (Capitolo, Figura, Bibliografia, ecc.).

Per ovviare a questo inconveniente è in via di sviluppo il pacchetto `polyglossia`, che dovrebbe costituire l'alternativa futura a `babel` per XƎTEX e, eventualmente, per gli altri sistemi di composizione derivati da TEX e basati su Unicode (ad esempio LUATEX).

Il pacchetto è ancora in una fase sperimentale e non si trova su CTAN, ma è disponibile dall'archivio dei sorgenti di XƎTEX.<sup>18</sup> L'attuale versione sembra comunque funzionare già abbastanza bene. Non esistendo una documentazione del pacchetto, bisogna dedurre la sua interfaccia utente dagli esempi allegati.

Dopo aver caricato il pacchetto, i vari moduli vengono caricati nel preambolo con il comando `\setdefaultlanguage`, nel caso della lingua principale, o `\setotherlanguage`. Entrambi i comandi, oltre all'argomento obbligatorio con il quale va indicata la lingua da attivare, specificandone il modulo, prevedono anche un argomento opzionale, sotto forma di una lista di elementi "chiave = valore". Ad esempio tramite l'opzione `variant=ancient`, passata al modulo `greek`, è possibile attivare il supporto per il greco antico politonico. Ogni volta che si vorrà attivare, nel testo, una lingua differente da quella predefinita, basterà racchiudere il testo in questione tra `\begin{<lingua>}` e `\end{<lingua>}`, oppure inserire l'istruzione `\selectlanguage{<lingua>}`.

Nelle figure 3<sup>19</sup> e 4<sup>20</sup> sono riportati due esempi corredati del relativo codice. Nel preambolo sono state inserite le seguenti dichiarazioni:

```
\usepackage{polyglossia}
\setdefaultlanguage{italian}
```

18. <http://scripts.sil.org/svn-view/xetex/TRUNK/texmf/tex/xelatex/polyglossia/>.

19. Il testo riproduce il primo capoverso delle *Storie* di Erodoto, come appare in ERODOTO (2006).

20. Si tratta della prima strofa del poemetto di Vladimir Majakovskij *La nuvola in calzon* (MAJAKOVSKIJ, 1989).

```
\fontspec{GFS Didot}
\begin{greek}
'Ηροδότου 'Αλικαρνησσεός ιστορίης απόδεξις
ἦδε, ὡς μήτε τὰ γενόμενα ἐξ ἀνθρώπων τῷ χρόνῳ
ἐξίτηλα φένται μήτε ἔργα μεγάλα τε καὶ
θωυμαστά τὰ μὲν Ἕλλησι, τὰ δὲ βαρβάροισι
ἀποδεχθέντα ἀκλεᾶ γένται, τὰ τε ἄλλα καὶ δι'
ἦν αἰτίην ἐπολέμησαν ἀλλήλοισι.
\end{greek}
```

'Ηροδότου 'Αλικαρνησσεός ιστορίης απόδεξις ἦδε, ὡς μήτε τὰ γενόμενα ἐξ ἀνθρώπων τῷ χρόνῳ ἐξίτηλα φένται μήτε ἔργα μεγάλα τε καὶ θωυμαστά τὰ μὲν Ἕλλησι, τὰ δὲ βαρβάροισι ἀποδεχθέντα ἀκλεᾶ γένται, τὰ τε ἄλλα καὶ δι' ἦν αἰτίην ἐπολέμησαν ἀλλήλοισι.

FIGURA 3: Greco antico

```
\setotherlanguage{russian}
\setotherlanguage[variant=ancient]{greek}
```

I font utilizzati sono, nel primo esempio, GFS Didot e, nel secondo, LinuxLibertine. Il codice, invece, è composto in UM Typewriter.

### 3.1.3 Altri pacchetti

Oltre a fontspec e polyglossia esistono altri pacchetti 'minori' dedicati a X<sub>Y</sub>TeX. In genere si tratta o di pacchetti che l'utente non ha bisogno di caricare direttamente ma che svolgono, ad esempio, compiti di supporto per quanto riguarda la codifica (xunicode e euenc), oppure di pacchetti che offrono alcune funzionalità relative ad ambiti linguistici specifici (arabxetex, xgreek, xecyr). Gli unici due pacchetti che meritano qualche riga di descrizione, in questa sede, sono xltextra e philokalia.

Il primo (ROBERTSON, 2007c) comprende una serie di correzioni e di aggiunte al formato base di X<sub>Y</sub>TeX: incorpora, ad esempio, il pacchetto fixltx2e; ridefinisce il logo di TeX e di alcuni programmi derivati (L<sup>A</sup>TeX, X<sub>Y</sub>TeX, X<sub>Y</sub>L<sup>A</sup>TeX); ridefinisce `\textsuperscript` e `\textsubscript` in modo da usare automaticamente le cifre in apice e in pedice fornite dal font; fornisce il comando `\vfrac` per comporre frazioni all'interno del testo usando le funzioni tipografiche avanzate del font; fornisce, tramite `\namedglyph{<nomeglifo>}`, un'interfaccia di alto livello alla primitiva di X<sub>Y</sub>TeX che consente di recuperare un carattere tramite il suo nome; infine ridefinisce il comando `\showhyphens` in modo che funzioni con X<sub>Y</sub>TeX.

philokalia (SYROPOULOS, 2007), invece, accompagna l'omonimo font, riproduzione di un carattere usato nella composizione di una serie di libri di devozione, assemblati nel monastero del Monte Athos in una collezione detta, appunto, 'Philokalia'.<sup>21</sup> Il pacchetto, che si appoggia a fontspec, xunicode e xltextra, fornisce all'utente i comandi necessari ad usare il font Philokalia come carattere principale

21. Si veda, per maggiori dettagli, <http://orthodoxwiki.org/Philokalia>.

```
\fontspec{Linux Libertine O}
\begin{russian}
\begin{verse}
Вашу мыслб\
мечтающую на размягченном мозгу,\
как выжиревший лакей на засалеиной кушетке,\
буду дразнить об окровавленный сердца лоскут;\
досыта изъиздеваюсьб нахальный и едкийю.
\end{verse}
\end{russian}
```

Вашу мыслб мечтающую на размягченном мозгу, как выжиревший лакей на засалеиной кушетке, буду дразнить об окровавленный сердца лоскут; досыта изъиздеваюсьб нахальный и едкийю.

FIGURA 4: Cirillico (russo)

del documento o relativamente a singole porzioni di esso.

## 3.2 ConTeXt

Gli utenti di ConTeXt non hanno bisogno di ricorrere a pacchetti aggiuntivi. Il supporto per le funzionalità di X<sub>Y</sub>TeX è incluso nel nocciolo del programma. L'uso è semplice e ricalca la sintassi già vista per TeX puro, ovviamente adattata alle convenzioni in vigore in ConTeXt.

Per usare le funzioni tipografiche avanzate di uno *smart font* una volta tanto, è sufficiente la seguente istruzione:

```
\definedfont["Linux Libertine O":%
mapping=tex-text;+onum;+smcp"]
```

con la quale si comunica a ConTeXt di usare il font Linux Libertine, installato nel sistema, nella forma "maiuscoletto", corredato di numeri "oldstyle". Si noti che è stata attivata anche l'opzione per tradurre nei rispettivi codici Unicode i costrutti tipici di TeX (per esempio la legatura --, equivalente ad un trattino medio '–') e che le varie opzioni possono essere separate da punto e virgola invece che da due punti. Nel caso che contenga degli spazi, il nome del font deve essere racchiuso tra ulteriori apici. Questo non è necessario, come vedremo qui sotto, nel caso di dichiarazioni globali.

Il comando `\definedfont` è utile nel caso si debba ricorrere una sola volta ad una particolare istanza di un font. Se invece il font in questione deve essere impostato come carattere principale del documento, in tutte le sue forme (normale, corsivo, maiuscoletto, ecc.) è necessario che le relative dichiarazioni vengano organizzate in maniera coerente. ConTeXt adotta un approccio particolare che può essere considerato l'equivalente di un *Font Definition file* per L<sup>A</sup>TeX. Un possibile esempio di tale organizzazione è il seguente:

```
\starttypescript[serif][libertineos][uc]
```

```

\definefontsynonym[LibertineOS]
  ["Linux Libertine 0%
   :mapping=tex-text:onum"]
  [encoding=uc]
\definefontsynonym[LibertineItalicOS]
  ["Linux Libertine 0/I%
   :mapping=tex-text:+onum"]
  [encoding=uc]
\definefontsynonym[LibertineBoldOS]
  ["Linux Libertine 0/B%
   :mapping=tex-text:+onum"]
  [encoding=uc]
\definefontsynonym[LibertineBoldItalicOS]
  ["Linux Libertine 0/BI%
   :mapping=tex-text:+onum"]
  [encoding=uc]
\definefontsynonym[LibertineSCOS]
  ["Linux Libertine 0%
   :mapping=tex-text:+smcp:+onum"]
  [encoding=uc]

\stoptypescript

\starttypescript[serif][libertineos][name]

\definefontsynonym[Serif]
  [LibertineOS]
\definefontsynonym[SerifBold]
  [LibertineBoldOS]
\definefontsynonym[SerifItalic]
  [LibertineItalicOS]
\definefontsynonym[SerifBoldItalic]
  [LibertineBoldItalicOS]
\definefontsynonym[SerifSlanted]
  [LibertineItalicOS]
\definefontsynonym[SerifBoldSlanted]
  [LibertineBoldItalicOS]
\definefontsynonym[SerifCaps]
  [LibertineSCOS]

\stoptypescript

\starttypescript [LinuxLibertine0][uc]

\definetypface[LinuxLibertine][rm][serif]
  [libertineos][default][encoding=uc]

\stoptypescript

\usetypescript [LinuxLibertine0][uc]
\setupbodyfont [LinuxLibertine,10pt]

```

Nella prima parte si trova la mappatura tra il livello “fisico” (il font “reale”) e un primo livello astratto. Le mappature seguenti aumentano il livello di astrazione, fino a giungere al nome simbolico (`LinuxLibertine`) che verrà usato nel documento per identificare la collezione di font usata per il testo principale. Si può notare, dal codice riportato sopra, che la codifica usata è Unicode (`uc`) e

che non c’è bisogno di un secondo livello di apici attorno al nome del font, anche se questo contiene degli spazi.

Questa interfaccia non è del tutto soddisfacente, perché ancora troppo legata ad una sintassi di basso livello, per quanto riguarda la specificazione delle funzioni tipografiche avanzate da attivare. Non a caso, nelle ultime versioni di ConT<sub>E</sub>Xt<sup>22</sup> (successive alla pubblicazione di T<sub>E</sub>X Live 2007), è stata sviluppata una nuova interfaccia, ancora sperimentale, in grado di trattare ad un livello più astratto anche le funzioni tipografiche avanzate. Il seguente codice

```

\definefontfeature[OSC]
  [onum=yes,smcp=yes]
\definefontsynonim[Libertine]
  [Linux Libertine 0]
  [feature=OSC]

```

definisce il font `Libertine`, che potrà essere richiamato in qualsiasi punto del testo tramite il comando `\definedfont`, e gli applica l’insieme di funzioni tipografiche avanzate definite dal nome simbolico `OSC`. A questo nome sono state in precedenza, tramite il comando `\definefontfeature`, associate le funzioni “maiuscoletto” (`smcp`) e “numeri oldstyle” (`onum`). Un discorso simile vale anche per le definizioni globali.

Per concludere facciamo presente che per usare X<sub>Y</sub>T<sub>E</sub>X insieme a ConT<sub>E</sub>Xt il file sorgente deve essere compilato passando all’eseguibile `texexec` l’opzione `--xtx`.

## 4 Formule e grafica

### 4.1 Formule

Dal momento che X<sub>Y</sub>T<sub>E</sub>X è in grado di usare anche i tradizionali font in versione TFM, le formule possono essere composte senza problemi usando pacchetti e costrutti tradizionali. Tuttavia cominciano ad essere disponibili font matematici in formato OPENTYPE: Microsoft distribuisce Cambria Math insieme al proprio sistema operativo, su CTAN si trova Asana Math, e i font STIX<sup>23</sup>, che dovrebbero diventare lo standard per le pubblicazioni scientifiche, sono appena usciti dalla fase beta dello sviluppo e, a breve, ne dovrebbe essere disponibile la prima versione stabile.

Per poter sfruttare questi nuovi font, e per poter usare un input in formato Unicode anche per le parti matematiche del testo, è necessario il pacchetto `unicode-math`, per L<sup>A</sup>T<sub>E</sub>X. Questo pacchetto, ancora sperimentale e perciò reperibile solo dall’archivio delle versioni di X<sub>Y</sub>T<sub>E</sub>X e non su CTAN, è composto da un file di stile, due file che mappano i simboli in codifica Unicode nella rappresentazione

22. Le informazioni che seguono sono tratte da HAGEN (2007).

23. <http://www.stixfonts.org>.



(a) Computer Modern

(b) Garamond Math Design

(c) Asana Math

(d) Asana Math colorato

FIGURA 5: Formule composte con  $\text{\XeTeX}$ .

interna di  $\text{\LaTeX}$ , e una serie di mappe compilate con TECKit.

All'utente viene fornito il comando `\setmathfont` con il quale è possibile scegliere il font per le formule e applicargli una serie di opzioni. Queste opzioni possono essere applicate anche solo ad una determinata serie di caratteri, individuata per mezzo di codici Unicode, nomi  $\text{\LaTeX}$ , o nomi collettivi predefiniti (ad esempio `\mathopen` indica tutti i delimitatori di sinistra).

Nella figura 5 sono riportati quattro esempi riguardanti la composizione di formule con  $\text{\XeTeX}$ , usando il pacchetto `amsmath`. Nei primi due sono stati usati font tradizionali, in particolare per l'esempio 5b è stato usato il pacchetto `mathdesign` con l'opzione `garamond`. Negli ultimi esempi il font usato è Asana Math, ed è stato caricato nel preambolo il pacchetto `unicode-math`. Per ottenere la colorazione dell'esempio 5d (in tonalità di grigio, nella versione a stampa), è stato impiegato il seguente codice:

```
\setmathfont[Range=ALL]{Asana Math}
\setmathfont[Range={\mathopen,\mathclose},%
  Color=FF0000]{Asana Math}
\setmathfont[Range={\mathop},%
  Color=0000FF]{Asana Math}
\setmathfont[Range={\mathbin,\equal},%
  Color=00FF00]{Asana Math}
```

Va notato che la selezione di opzioni per diverse gamme definite di simboli rallenta notevolmente il processo, fino al punto, nel caso si cerchi di applicarle a `\mathalpha`, di dover abortire il processo stesso (quest'ultima notazione, ovviamente, è vera relativamente al computer su cui sono state effettuate le prove, equipaggiato con un processore Athlon XP 1600+ e 512Mb di memoria RAM).

## 4.2 Grafica

### 4.2.1 Inclusione di illustrazioni esterne

$\text{\XeTeX}$  supporta l'inclusione di illustrazioni prodotte con programmi esterni, nei formati riconosciuti dal driver di stampa in uso. Vi sarà, quindi, differenza, a seconda che venga usato `xdv2pdf` o `xdvipdfmx`. Il primo, che funziona, come si è detto, solo su sistemi Macintosh, si appoggia alle librerie di QuickTime ed è, dunque, in grado di operare l'inclusione di una gamma piuttosto estesa di formati, che vanno dal Bitmap al TIFF, dall'EPS al PDF, passando per PICT, TGA, ecc.<sup>24</sup> `xdvipdfmx`, invece, è in grado di includere nel documento gli stessi formati supportati da `dvipdfm`: PNG, JPEG, PDF, EPS.

Per effettuare l'inclusione, l'utente può avvalersi, in maniera del tutto trasparente, dei comandi usuali: quelli definiti da `graphicx` se usa  $\text{\LaTeX}$ , o quelli forniti di base da `ConTeXt`, nel caso usi quest'ultimo programma.

### 4.2.2 pgf

Dalla fine del 2007 la libreria di utilità grafiche `pgf`, e quindi i programmi costruiti a partire da essa, come `Tikz`, includono il file `pgfsys-xetex.def` che fornisce il supporto per il driver `xdvipdfmx`. A patto, quindi, di compilare con questo driver di stampa, è possibile inserire nel proprio documento grafici espressi nel linguaggio proprio di `pgf`. Tuttavia, non tutte le funzionalità avanzate sono al momento disponibili. Non è possibile, ad esempio, riempire un'area dell'illustrazione con motivi

24. È bene ricordare, però, che per l'inclusione di immagini in formato PDF, è preferibile usare la primitiva `\XeTeXpdfimage`, piuttosto che `\XeTeXpictimage`, se si vuole che venga preservato il carattere vettoriale dell'immagine stessa, nel caso, ovviamente, che si stia usando `xdv2pdf` come driver di stampa.

A	B	C	D	E	F	G	A	B	C	D	E	F	G
H	I	K	L	M	N	O	H	I	K	L	M	N	O
P	Q	R	S	T	V	W	P	Q	R	S	T	v	w
X	Y	Z	Æ	J	U	Œ	x	y	z	æ	J	U	ffb
1	2	3	4	5	6	7	â	ê	î	ô	û	¶	§
8	9	o	fk	f	ft	ffi	à	è	ì	ò	ù	†	‡
ä	ë	ï	ö	ü	ft	k	á	é	í	ó	ú	tt	*
ct	[ ]	æ	œ	ç	`	´	s	( )	? !	:	fh	fl	
&	b	c	d	e	i	s	f	g	fb	ff	fft		
ffl	l	m	n	h	o	y	p	q	w	em	en		
j	st	z	v	u	t	space	a	r	.	:	quad		
x									.	-			

FIGURA 6: Una cassetta di caratteri disegnata con Tikz.

ripetuti (tassellatura), né applicare ad un'immagine esterna una maschera di trasparenza selettiva. `pgf`, infatti, pur mirando ad essere perfettamente portabile da un compilatore all'altro, dipende, per l'effettivo grado di supporto delle singole operazioni, dai diversi driver di stampa, ed è ottimizzato per funzionare con `pdftex`. Tutti gli altri driver, in misura maggiore o minore, mancano invece di qualche funzionalità.

Nella figura 6 è visibile il risultato di un grafico prodotto con `tikzpicture` e compilato con XYTEX.

#### 4.2.3 PSTricks

PSTricks è, probabilmente, il sistema grafico più utilizzato in combinazione con LATEX. La possibilità di inserire nel file DVI frammenti di codice POSTSCRIPT che dovranno poi essere interpretati dai vari driver di stampa, lo rende uno dei più potenti e flessibili strumenti per la grafica nel mondo di TEX. Non è sorprendente, perciò, che si stia cercando di rendere PSTricks compatibile con XYTEX. Anche in questo caso il supporto viene fornito per `xdvipdfmx`. Dall'archivio delle versioni di XYTEX è possibile scaricare il file `xetex-pstricks.com` che, una volta rinominato `pstricks.com` e copiato in una cartella dove `pstricks` può trovarlo, configura quest'ultimo in modo da usare `xdvipdfmx` come driver di stampa.<sup>25</sup>

Nel concreto, l'interpretazione del codice POSTSCRIPT inserito nell'XDVI viene fatta per mezzo di Ghostscript, di cui si consiglia di installare una versione recente, se si vuole che vengano supportate operazioni di carattere avanzato, come, ad esempio, effetti di trasparenza.

25. Alternativamente, è possibile usare `xdvipdfmx.com`, facente parte della distribuzione base di PSTricks, che però appare leggermente indietro nello sviluppo, rispetto al suo omologo distribuito da XYTEX.

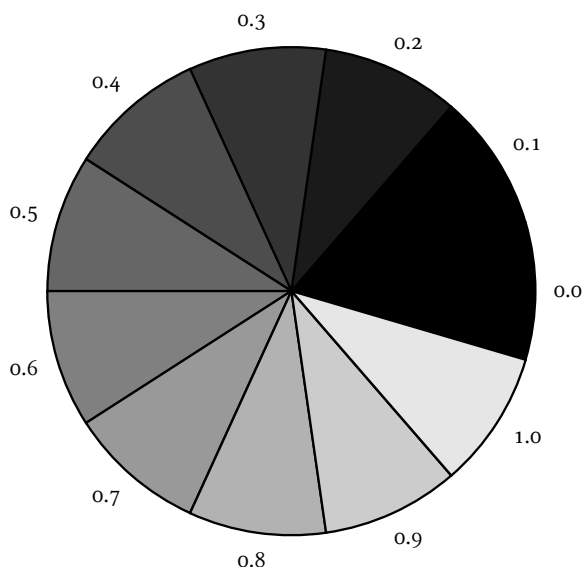


FIGURA 7: Esempio di grafico con pstricks

Per quanto questo progetto sia ancora in fase di sviluppo, i risultati sono già buoni, come si può vedere nella figura 7.

## 5 Conclusioni

Questo articolo mostra come XYTEX sia ormai un prodotto maturo, disponibile su tutte le piattaforme e utilizzabile insieme ai principali "dialetti" di TEX (Plain TEX, LATEX, ConTEXt). Non è ancora esente da imperfezioni: deve migliorare il supporto per alcune funzioni tipografiche avanzate implementate dalle *smart font technologies*, ha bisogno di un sostituto per `babel` (`polyglossia` è il promettente candidato), deve migliorare la gestione dei font matematici in codifica Unicode e il supporto per sistemi di applicazioni grafiche come PSTricks e `pgf`.

Malgrado queste piccole imperfezioni, la facilità d'uso che introduce in un settore, quello della gestione dei caratteri da stampa, tradizionalmente tra i più complicati e ostici, nel mondo TEX, lo rende uno strumento utile e, in alcuni casi, come la composizione di documenti contenenti sistemi di scrittura non occidentali, addirittura quasi indispensabile. Essendo un programma in fase di attivo sviluppo e con una base di utilizzatori numericamente non indifferente, possiamo tranquillamente aspettarci ulteriori miglioramenti nel futuro prossimo.

Per concludere, vorrei aggiungere una nota finale sui font. Per quanto l'utilità di XYTEX si dimostri anche soltanto nel caso in cui si voglia usare un font senza bisogno di doverlo installare nel sistema TEX, il programma dà il meglio di sé quando si trova a gestire degli *smart font*. Questi font, che siano AAT, OPENTYPE o Graphite, si trovano oggi con facilità. Senza voler citare font commerciali, distribuiti da Adobe, Monotype, Linotype, e da

quasi tutte le altre case, grandi, medie o piccole che siano, la cui qualità è certamente garantita, è possibile ormai reperire *smart font* di buona qualità anche nel circuito del software libero. Molti dei font prodotti da *e-foundry* per  $\TeX$  (Latin Modern, la collezione  $\TeX$  Gyre, Antykwa Toruńska, Iwona) sono disponibili anche in formato OPEN-TYPE ed è possibile scaricarli sia da CTAN che dal sito dalla *e-foundry*.<sup>26</sup> Ancora disponibili su CTAN sono UM Typewriter, un font a spaziatura fissa codificato Unicode, e Asana-Math, font matematico anch'esso in codifica Unicode, entrambi di Apostolos Syropoulos.

Per uscire dal mondo  $\TeX$ , Junicode,<sup>27</sup> font per medievalisti e non solo, e Linux Libertine,<sup>28</sup> sono tra i più completi e ben fatti, rilasciati sotto i termini della licenza GNU GPL. Accanto a questi, vanno ricordati i font prodotti da SIL International con licenza OFL, in particolare Gentium, Doulos e Charis, corredati di tabelle sia OPEN-TYPE che Graphite; e quelli della Greek Font Society,<sup>29</sup> anch'essi con licenza OFL, soprattutto GFS Bodoni e GFS Didot, che contengono, oltre ai caratteri dell'alfabeto greco, anche l'alfabeto latino.

Né, infine, sono da dimenticare Old Standard, di Aleksej Kryukov,<sup>30</sup> e i font dedicati a sistemi di scrittura antichi prodotti da George Douros<sup>31</sup> tra cui il bellissimo Alexander, il cui alfabeto greco è ricalcato su quello disegnato da Alexander Wilson nel XVIII secolo, purtroppo disponibile solo in versione corsiva.

## 6 Ringraziamenti

Questo articolo non sarebbe mai stato scritto, se non mi fosse stato, praticamente, "commissionato" da Lapo F. Mori, a cui vanno, quindi, per primo i miei ringraziamenti.

Desidero inoltre ringraziare l'anonimo recensore dell'articolo per avermi indicato i punti in cui poteva essere migliorato, e per avermi aiutato a rettificare alcune inesattezze riguardo alla relazione tra Unicode e lo standard ISO/IEC 10646.

## Riferimenti bibliografici

APPLE COMPUTER (2002). *TrueType Reference Manual*, Apple Computer, capitolo 6: Font files. <http://developer.apple.com/fonts/TTRefMan/RM06/Chap6AATIntro.html>.

BEETON, B., FREYTAG, A. e SARGENT III, M. (2007). «Unicode support for mathematics». <http://www.unicode.org/reports/tr25/>.

26. <http://www.gust.org.pl/projects/e-foundry/>.

27. <http://junicode.sourceforge.net>.

28. <http://linuxlibertine.sourceforge.net>.

29. <http://www.greekfontssociety.gr/>.

30. <http://www.thessalonica.org.ru/en/fonts.html>.

31. <http://users.telar.gr/~g1951d/downloads>.

DONNE, J. (1959). *Devotions Upon Emergent Occasions Together with Death's Duel*. Ann Arbor Paperbacks, University of Michigan Press. La versione utilizzata è quella fornita da *Project Gutenberg*: <http://www.gutenberg.org/2/3/7/7/23772/>.

ERODOTO (2006). *Le storie*. UTET. A cura di Aristide Colonna e Fiorenza Bevilacqua. Con testo a fronte.

HAGEN, H. (2007). «Con $\TeX$ t MKII & MKIV.» Disponibile all'indirizzo <http://www.pragma-ade.com/general/manuals/mk.pdf>.

HOSKEN, M., HALLISSY, B., CLEVELAND, W., CORRELL, S. e WARD, A. (2007). *Graphite Language Description*. SIL Non-Roman Scripts Initiative (NRSI). La versione elettronica in formato PDF è disponibile all'indirizzo [http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&cat\\_id=RenderingGraphite](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&cat_id=RenderingGraphite).

KEW, J. (2002). *TECkit version 2.0. A Text Encoding Conversion toolkit*. SIL Non-Roman Scripts Initiative (NRSI). La versione elettronica in formato PDF è disponibile all'indirizzo [http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&item\\_id=TECkitDownloads](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=TECkitDownloads).

— (2005). « $\XeTeX$  the Multilingual Lion:  $\TeX$  meets Unicode and smart font technologies». *TUGboat*, **26** (2), pp. 115–124.

— (2007). «About  $\XeTeX$ ». Disponibile all'indirizzo <http://scripts.sil.org/xetex/XeTeX-notes.pdf>.

— (2008). « $\XeTeX$  Live». *TUGboat*, **29** (1), pp. 146–150.

KNUTH, D. E. (1984). *The  $\TeX$ book*. Addison-Wesley, Reading, MA, USA.

MAJAKOVSKIJ, V. (1989). *La nuvola in calzoni*. Marsilio. A cura di Remo Faccani. Con testo a fronte.

MICROSOFT CORPORATION (2004). «OpenType specification». <http://www.microsoft.com/typography/otspec/default.htm>.

ROBERTSON, W. (2005). «Advanced font features with  $\XeTeX$  – the fontspec package». *TUGboat*, **26** (3), pp. 215–223.

— (2007a). «The  $\XeTeX$  reference guide». Disponibile su CTAN, all'indirizzo <http://www.ctan.org/tex-archive/info/xetexref/XeTeX-reference.pdf>.

— (2007b). «The fontspec package.» Disponibile su CTAN, all'indirizzo <http://www.ctan.org/tex-archive/macros/xetex/latex/fontspec.pdf>.

- (2007c). «The xltextra package.» Disponibile su CTAN, all'indirizzo <http://www.ctan.org/tex-archive/macros/xetex/latex/xltextra.pdf>.
- (2008). «Experimental unicode mathematical typesetting: The unicode-math package». <http://scripts.sil.org/svn-public/xetex/TRUNK/texmf/source/xelatex/unicode-math/unicode-math.pdf>.
- SYROPOULOS, A. (2007). «The philokalia package.» Disponibile su CTAN, all'indirizzo <http://www.ctan.org/tex-archive/macros/xetex/latex/philokalia.pdf>.
- UNICODE CONSORTIUM (2007). *The Unicode standard, Version 5.0*. Addison-Wesley, Boston, MA.
- ▷ Massimiliano Dominici  
[mlgdominici@interfree.it](mailto:mlgdominici@interfree.it)