

Cenni sulla produzione di ebook con L^AT_EX e Calibre

Gianluca Pignalberi

Sommario

La produzione di libri elettronici (ebook) parallela alla produzione di libri cartacei sembra essere ormai la norma. Vediamo una possibile via di trasformazione dei documenti L^AT_EX in file EPUB standard.

Abstract

Making ebooks along with paper books seems to be normal now. We will see a possible way to transform L^AT_EX documents into standard EPUB files.

1 Introduzione

Nel mio vecchio articolo PIGNALBERI (2015) accennavo al fatto che una parte del mio lavoro di compositore L^AT_EX è assorbita dalla produzione di ebook in formato EPUB dei libri di cui realizzo i PDF, siano essi destinati alla sola distribuzione in formato elettronico o anche alla stampa. Per il mio lavoro ricorro esclusivamente al software libero: L^AT_EX per i libri e, se possibile, le copertine; GIMP per il fotoritocco (col grosso handicap della gestione CMYK); Calibre per la conversione da HTML a EPUB; Gnuplot per i grafici; QTikZ per i disegni in TikZ; una serie di altri programmi più o meno noti per altre necessità di lavorazione. A questi strumenti devo aggiungere due: Sigil (SCHEMBER, 2018), un potente editor di ebook che integrava il vecchio programma standard per la validazione dei documenti EPUB (FlightCrew), e epubcheck (EPUB-CHECK TEAM, 2018), attuale validatore standard, che non è rilasciato sotto licenza GPL.

Nell'articolo spiegherò le varie fasi per ottenere un EPUB conforme allo standard e, dunque, accettato dalle più note e comuni piattaforme di vendita di ebook, a partire da un documento L^AT_EX.¹

Dietro suggerimento del direttore ho usato un documento scientifico invece che uno dei saggi letterari che produco di solito. Il documento è costituito da un capitolo del libro D'ANTONA (1998) e ci permetterà di vedere formule e grafici, ma non testo in greco o in ebraico che genera problemi diversi. Riprodussi quasi esattamente² tale capitolo nel 2013

1. Tutte le prove descritte nell'articolo sono state ottenute con i programmi inclusi in T_EX Live 2017.

2. Poiché il progetto con gli allievi non prevedeva la riproduzione *esatta* dell'aspetto del libro, alcuni elementi nelle pagine degli esercizi non sono posizionati al millimetro, sebbene la cosa potrebbe non balzare immediatamente a occhi meno esperti.

insieme ai miei allievi del terzo (e ultimo) anno del corso di Operatore Grafico Multimediale (la mia materia era denominata Open Source) presso Latina Formazione Lavoro. Lo scopo era addestrarli all'uso di LyX e di TikZ (il primo era un tentativo di far digerire loro gli accenni a L^AT_EX mascherando quest'ultimo da editor WYSIWYG, il secondo era parte del programma sul disegno vettoriale) e per mostrare loro con quanta facilità ottenevamo testi matematici di aspetto migliore di quelli ottenibili con i programmi di composizione tipografica studiati in altre materie. Possiamo vedere alcune delle pagine prodotte nella figura 1. Di tale documento producemmo anche la versione EPUB, non validata. La validazione verrà fatta appositamente per questo articolo in virtù della sua finalità di mostrare come creare un prodotto vendibile.

Naturalmente quello mostrato è un progetto didattico, ottimo per studiare alcuni aspetti della lavorazione. Nei limiti della mia esperienza parlerò anche di altri problemi legati alla produzione di ebook. Il progetto didattico è troppo limitato per mostrare tutti i problemi affrontati per lavoro.

2 Primo tempo: da L^AT_EX a HTML

Lo strumento che uso per compilare un documento L^AT_EX e ottenere un file HTML è `htlatex` (GURARI, 2004). Pur essendo per costruzione meno flessibile di L^AT_EX (non è progettato per usare una classe il cui nome non sia uno di quelli standard di L^AT_EX — `book`, `report`, `article`, `letter` e pochissimi altri — né pacchetti di cui non sia stato scritto il “porting”), mi permette con (relativamente) poco lavoro di ottenere un HTML a cui serviranno solo alcune modifiche. Naturalmente `htlatex` non è l'unico strumento utilizzabile. Ritengo che `pandoc` potrebbe essere addirittura migliore se veramente mi permettesse di 1) saltare il passaggio intermedio della mia catena di lavoro (L^AT_EX → HTML → EPUB) e 2) lasciare intatti i miei documenti senza le modifiche necessarie, anche solo al preambolo, per ovviare alle limitazioni e ai vincoli di `htlatex`.

Esiste anche un pacchetto per X_YL^AT_EX e LuaL^AT_EX (HOFTICH, 2016, il refuso nel titolo è quello originale) che simula il risultato di `htlatex` ma io l'ho sperimentato pochissimo e senza alcun successo. Tra l'altro il progetto, iniziato almeno nel 2013, ha avuto un ultimo commit nel 2016 e non sembra essere più attivo.

Come detto in precedenza, `htlatex` non è un prodotto flessibilissimo, anche se alcune libertà ce le permette. Per comporre i libri per lavoro sono



"progetto_gerzo" — 2013/6/13 — 9:20 — page 163 — #1



"progetto_gerzo" — 2013/6/13 — 9:20 — page 167 — #5



4 I grafi cordali e la Ricorrenza Master

4.1 Il polinomio cromatico come combinazione lineare di fattoriali decrescenti

Come abbiamo avuto occasione di dire (ben più di una volta) nelle pagine precedenti, il polinomio cromatico di un grafo indica il numero di modi in cui lo si può colorare usando al più n colori. Più precisamente sappiamo che, per un valore di n arbitrariamente fissato, $p(G, n)$ sarà il numero di modi di colorare G con uno, due o n colori. Ad esempio le due funzioni $chrom_1, chrom_2 : \{v_1, v_2\} \rightarrow \{BLU, GIALLO, ROSSO, VERDE\}$ utilize

$$chrom_1(v_1) = ROSSO, chrom_1(v_2) = BLU$$

$$chrom_2(v_1) = VERDE, chrom_2(v_2) = BLU$$

sono due delle 16 colorazioni di G_2 che si possono realizzare avendo a disposizione quattro colori (infatti $p(G_2, 4) = 4^2 = 16$) anche se abbiamo effettivamente utilizzato prima un solo colore e poi soltanto due. In realtà la cosa non ci sorprende più di tanto per il buon motivo che alle funzioni che rappresentano le colorazioni non c'è rischio di essere surriscritte. Quindi, se fossimo interessati ad una classificazione delle colorazioni di un grafo in termini di quanti colori vengono effettivamente usati, e non tanto di quali, potremmo sospettare che il polinomio cromatico (almeno così come lo abbiamo presentato in [10]) non sia lo strumento più adatto. Definiamo allora un nuovo coefficiente relativo alle colorazioni di un grafo con la notazione $E_{n,k}(G)$ indicheremo il numero di colorazioni di un grafo G di ordine n che utilizzano effettivamente k colori [1], le tal modo abbiamo imposto una sorta di simmetria a queste colorazioni che potremmo quindi chiamare *k-narrative*, o anche *colorazioni esatte*.

Per capire bene il concetto, commentiamo alcune proprietà generali. In primo luogo risulta $E_{n,n}(G) = 0$ per ogni $n > n$. E fin qui tutto bene. Poi si ha sempre $E_{n,n}(G) = 1$.

¹Occasionalmente, orec non ci sia rischio di confusione ci limiteremo a scrivere $E_{n,k}$ in luogo di $E_{n,k}(G)$



"progetto_gerzo" — 2013/6/13 — 9:20 — page 174 — #12



"progetto_gerzo" — 2013/6/13 — 9:20 — page 175 — #13



174 I grafi cordali e la Ricorrenza Master

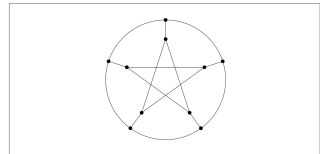


Figura 4.6. Il grafo di Petersen.

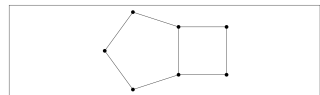


Figura 4.7. Questo grafo non è perfetto.

Tra l'altro, questo ragionamento si estende facilmente a tutti i grafi che contengono, come sottografo indotto, dei ciccoli di lunghezza dispari (maggiore di 3) come le ruote W_{2n+1} con $n \geq 1$. Inoltre, l'elenco di famiglie di grafi perfetti può, per così dire, essere duplicato poiché si sa che il complemento di un grafo perfetto è perfetto. Su questo tema il famoso grafista francese C. Berge ha congetturato che un grafo G sia perfetto se e soltanto se né G né il suo complemento contengono come sottografi indotti ciccoli dispari di lunghezza maggiore di 4.

Passiamo ora al numero di copertura che, nel folklore della teoria, viene interpretato come il minimo numero di guardiani necessari a tenere sotto 'occhio tutti i corridoi di un museo. Un insieme C di vertici di un grafo G è detto *copertura* del lato di G se ogni suo lato è incidente in



FIGURA 1: Pagine riprodotte da D'ANTONA (1998).

Capitolo 4 167

Ogni grafo cordale è ottenuto con una serie di incastri piramidali a partire da I_n . Ma si badi bene: dopo aver effettuato un'operazione di piramide, il grafo risultante conterrà una cricca di ordine 1 o maggiore. Inoltre è ovvio che la più grande cricca che un grafo di ordine n può contenere è proprio K_n . Anzi, a proposito dei grafi completi, possiamo dire che questi sono ottenuti con una serie di incastri piramidali massimali poiché, ad ogni passo, il grado del nuovo vertice è il più grande possibile. In altre parole, A_n è ottenuto aggiungendo ad I_n delle piramidi di grado 1, 2, ..., $n-1$. All'altro estremo si trovano proprio i grafi I_n , che sono invece ottenuti a partire da A_1 con una serie di incastri piramidali minimali (anzi minimi) cioè di grado 0. Dunque la sequenza dei grafi delle piramidi con cui abbiamo costruito I_n è $0, 0, \dots, 0$. Questa caratterizzazione ha il gradito effetto collaterale di farci conoscere la struttura del polinomio cromatico di tutti i grafi cordali. Infatti basta osservare che i polinomi cromatici di K_n e di I_n sono rispettivamente, $(x)_n$ e x^n per iniziare a sospettare che detti $r_1, r_2, r_3, \dots, r_n$ i gradi delle piramidi con cui è stato costruito il generico grafo cordale G di ordine n , il suo polinomio cromatico sarà

$$p(G, x) = (x - r_1)(x - r_2) \dots (x - r_n),$$

dove $0 \leq r_1 \leq r_2 \leq \dots \leq r_n \leq n-1$. E in effetti è così. Abbiamo dunque conseguito il semplice, ma importante risultato che recita: *tutte le radici del polinomio cromatico di un grafo cordale sono intere*. Ma attenzione: è della massima importanza tener presente che

IL CONTRARIO NON È VERO !!!

Nel prossimo capitolo analizzeremo diffusamente questa affermazione, ma ora limitiamoci ad osservare che, se spogliamo la scrittura del polinomio cromatico di un grafo in termini di fattoriali decrescenti dalla sua veste pitonica, ciò che resta è una semplice relazione algebrica tra polinomi. In altre parole, l'espressione

$$p(G, x) = \sum_{k=0}^n E_{n,k}(G) x^k$$

può essere pensata come l'espressione del polinomio $p_n(x) = p(G, x)$ come combinazione lineare di fattoriali decrescenti in cui le costanti di connessione sono indicate da $E_{n,k}(G)$ in luogo dell'usuale (dittono per noi) simbolo $E_{n,k}$. A questo punto si può osservare che i polinomi cromatici dei grafi che costituiscono una sequenza di eliminazioni di un grafo cordale sono una famiglia persistente di polinomi. Dunque la ricorrenza master diventa una relazione tra le colorazioni esatte di un grafo cordale e quelle del grafo che lo precede in una sequenza di eliminazione [3]. Per interpretare la ricorrenza in termini di grafi, suddividiamo in due classi tutte le partizioni indipendenti di G_n in k blocchi. Nella prima metteremo tutte quelle partizioni in cui il vertice v_n costituisce un blocco a sé stante, e nella seconda tutte le altre. Ovviamente la prima classe conterrà tutti elementi quelle partizioni indipendenti di G_{n-1} in $k-1$ blocchi. Infatti aggiungendo a ciascuna di esse un blocco contenente il solo v_n ,

²Ovvero tra le colorazioni esatte di un generico grafo H e quelle ottenute costruendo una piramide su di esso.

Capitolo 4 175

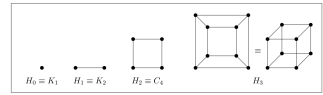


Figura 4.8. Quattro ipercubi.

almeno uno dei vertici di C . Ad esempio, tre vertici qualunque di C_3 sono una sua copertura. Il numero di vertici di una copertura di ordine minimo è detto il *numero di copertura* di G e viene indicato con $\alpha(G)$. Ad esempio, due vertici non consecutivi di C_4 sono una sua copertura, e in generale è facile vedere che $\alpha(C_n) = \lfloor \frac{n}{2} \rfloor$, e lo stesso vale per i cammini con n punti, se invece T_n è un albero che non sia un cammino, allora $\alpha(T_n) < \lfloor \frac{n}{2} \rfloor$. Per le ruote invece si ha $\alpha(W_n) = 1 + \lfloor \frac{n}{2} \rfloor$.

Il numero di copertura in circolo di un grafo è il minimo numero di sottografi completi necessari a ricoprire tutti i suoi vertici, e viene indicato con $\alpha(G)$. Ad esempio, $\alpha(W_n) = 1, \alpha(W_4) = 2$, e per $n > 4, \alpha(W_n) = 1 + \lfloor \frac{n}{2} \rfloor$. Ricordiamo che si può dimostrare la seguente caratterizzazione: un grafo G è perfetto se e soltanto se per ciascuno dei suoi sottografi indotti, S - ivi incluso G stesso - risulta $\alpha(S) = \alpha(S)$. Questo risultato ci dice che, dal punto di vista dei grafi perfetti, il numero di copertura in circolo e il numero cromatico giocano lo stesso ruolo.

Infine ricordiamo che un insieme *dominante* (o *ricoprimento*) di un grafo è un sottoinsieme D dei suoi vertici con la proprietà che ogni altro vertice del grafo sia adiacente ad almeno un vertice di D . Il numero di dominanza di un grafo è l'ordine di un insieme dominante minimo. Questo concetto di grande rilevanza nella teoria dei codici a correzione di errore in cui però i grafi di cui ci si occupa sono quasi esclusivamente gli ipercubi, H_n vedi Figura 4.8. È facile vedere che il numero di dominanza di H_1, H_2, H_3 e H_4 è, rispettivamente, 1, 2, 3 e 4. Ma è ben stato che per scoprire i 32 vertici di H_5 occorrono almeno 7 vertici. Sono noti anche i numeri di dominanza H_6, H_7 e H_8 che sono 12, 16 e 32, ma non quelli di ipercubi più grandi. La tabella qui sotto riporta le migliori limitazioni conosciute per numeri di dominanza di $H_9, H_{10}, H_{11}, H_{12}$ e H_{13} .

Numero di vertici	Limite inferiore	Limite superiore
9	55	62
10	105	120
11	178	192
12	342	380
13	598	736

solito usare una versione pesantemente modificata di `book` (PIGNALBERI, 2015), corredata di plugin e rinominata in ossequio alla LPPL. Per le tabelle, invece, uso `tabu` per la sua flessibilità (sebbene debba poi fare dei piccoli aggiustamenti). Entrambi i software non sono compatibili con l'uso di `htlatex`. Per ovviare al primo problema mi limito a prendere una nuova classe `book`, modificarla molto meno pesantemente (solo per aggiungere il frontespizio, il retrofrontespizio e alcuni comandi usati frequentemente) e, senza rinominarla, compilare senza problemi per ottenere un HTML accettabile. Per far funzionare le tabelle, invece, devo tornare al “primitivo” `tabular`, dovendo quindi modificare la sintassi della dichiarazione delle colonne (e sovente ricorrere a `p{}` con frazioni di `\textwidth` per non uscire “fuori margine”). La nota positiva è che non devo preoccuparmi delle tabelle lunghe perché nelle pagine HTML non manca lo spazio verticale (e l'EPUB gestisce la lunghezza e la suddivisione delle tabelle come lo farebbe `longtable`).

Per il progetto didattico non sussiste alcuno dei suindicati problemi perché il documento è basato sulla classe `book` (con le modifiche fatte direttamente nel preambolo del documento) e l'unica tabella è stata composta con `tabular`. Nel listato 1³ vediamo il preambolo del documento che ci permette di ottenere un documento dall'aspetto uguale a quello della figura 1.

LISTATO 1: Preambolo del documento riprodotto nel libro D'ANTONA (1998).

```

1 \documentclass{book}
2
3 \usepackage[T1]{fontenc}
4 %\usepackage[utf8x]{inputenc}
5 \usepackage[latin1]{inputenc}
6 \usepackage[italian]{babel}
7 \usepackage{color}
8 \usepackage{times}
9 %\usepackage{mathptmx}
10 \usepackage{tikz}
11 \usepackage{amsmath}
12 \usepackage[paperwidth=17cm,
    paperheight=24cm,outer=2cm,
    top=3.3cm,bottom=1.5cm]{
    geometry}
13 \usepackage[cam,a4,center]{crop}
14 \usepackage[format=plain,
    labelfont=it,textfont=it,
    justification=justified,
    singlelinecheck=false,skip=5
    mm]{caption}
15 \usepackage{fancyhdr}

```

3. Alcune righe dei listati numerati, troppo lunghe per entrare interamente nella colonna di testo, sono state interrotte automaticamente e portate a capo. Potete notare queste interruzioni dalla mancanza del numero di riga. Questo significa che non c'è alcuno spazio o invio tra un'ultimo carattere di una riga di codice e il primo della successiva riga non numerata.

```

16 \pagestyle{fancy}{%
17 \fancyhf{}
18 \fancyhead[RE]{\itshape\leftmark
    }
19 \fancyhead[L0]{\itshape
    \chaptername~\thechapter}
20 \fancyhead[LE,R0]{\itshape
    \thepage}
21
22 \makeatletter
23 \def\@makechapterhead#1{%
24 \begin{minipage}[t]{.15\
    textwidth}
25 \vskip-2pt
26 {\fontsize{84}{0}\selectfont
    \textcolor{gray}{
    \thechapter}}
27 \end{minipage}
28 \begin{minipage}{.80\textwidth
    }
29 \rule{0pt}{100pt}{\LARGE{
    \textbf{\ \ #1}
30
31 \hrulefill}}\end{minipage}
32 \vskip66pt
33 }
34 \renewcommand\section{
    \@startsection {section}{1}{
    \z@}%
35 {-3.5ex \@plus -1ex \@minus
    -.2ex}%
36 {2.3ex \@plus .2ex}%
37 {\normalfont\large\itshape}}
38 \makeatother
39 \parindent=14pt
40 \parskip=0pt
41 \frenchspacing

```

Più articolato è il discorso sulle figure. Se dobbiamo includere immagini raster non ci sono problemi: basta limitarci ai formati JPG e PNG. Già con i formati vettoriali EPS e PDF siamo pressoché costretti a una conversione e all'eventuale modifica del sorgente LATEX. Sebbene `htlatex` tenti una conversione delle immagini in questi formati, non sempre il risultato è almeno accettabile. Nel caso poi di figure contenenti il sorgente del disegno (`pict2e`, `TikZ`...), siamo costretti a generarle separatamente e a convertirne il formato, quindi a modificare il sorgente LATEX nel punto di inclusione dell'immagine. Siccome JPG e PNG non sono il massimo per riprodurre immagini vettoriali (meglio comunque il PNG), possiamo convertire questa tipologia di immagini in `.svg`, conformemente allo standard. Ho fatto una prova e il risultato è il seguente: Firefox 55.0.2 mostra le immagini SVG dell'HTML senza problemi. La conversione a ebook avviene nel modo descritto nella sezione 4 e, mentre Sigil mostra tutte le figure nel loro splendore, un lettore pur versatile come Calibre no. Dunque è

rischioso avventurarsi fuori dai due formati raster citati all'inizio del capoverso perché rischiamo che alcuni lettori commerciali non mostrino alcunché. Nel progetto didattico non ce ne discosteremo, pur usando immagini JPG a 72dpi invece che PNG a una risoluzione leggermente maggiore:⁴

```

137 \begin{figure}
138 \includegraphics[width=
      \textwidth]{figura1.jpg}
139 \emph{\caption{\$V=\{a,b,c\},\, E
      =\{(b,c)\}\$ \hspace{26em}}}{\hspace{26em}}
140 \end{figure}

```

del sorgente da dare in input a `htlatex` sostituisce

```

138 \begin{figure}
139 \framebox{
140 \begin{minipage}{.975\textwidth}
141 \vskip2mm
142 \centering
143 \begin{tikzpicture} \coordinate
      (a) at (0,0); \coordinate (c)
      at (2,0); \coordinate (b) at
      (1,1.73); \filldraw (a)
      circle (2pt); \filldraw (b)
      circle (2pt); \filldraw (c)
      circle (2pt); \draw (b) -- (c)
      ; \node [below left] at (a)
      {\$a\$}; \node [above] at (b)
      {\$b\$}; \node [below right] at
      (c) {\$c\$}; \end{tikzpicture}
      \vspace{2mm} \end{minipage}}
      \caption{\$V=\{a,b,c\},\, E
      =\{(b,c)\}\$ \hspace{26em}}
144 \end{figure}

```

del sorgente originale da cui ottenere il PDF. Ciò implica che avremo compilato un documento `standalone` contenente il codice compreso tra `\begin{tikzpicture}` e `\end{tikzpicture}` (riga 143) e convertito il relativo PDF nel file `figura1.jpg`, e questo per tutte le figure `TikZ` contenute nel documento.

Rimane l'ultimo elemento del progetto didattico: le formule. Con `htlatex` possiamo scegliere se renderle immagini (scelta più sicura per la compatibilità con la maggior parte dei lettori di ebook; peraltro non tutte le formule dovranno subire questo trattamento e il compilatore deciderà quali poter rendere in HTML) o formule MathML (scelta preferibile dal punto di vista della coerenza — la formula è conforme al testo indipendentemente dal ridimensionamento, cosa che non può avvenire con un'immagine — ma insicura dal punto di vista della compatibilità con i lettori di ebook, hardware o software⁵).

4. Le immagini sono rimaste quelle convertite originariamente, rese a una risoluzione uguale a quella dei monitor commerciali.

5. È ovvio che un lettore di ebook hardware integri al proprio interno un software per leggere gli ebook, ma la

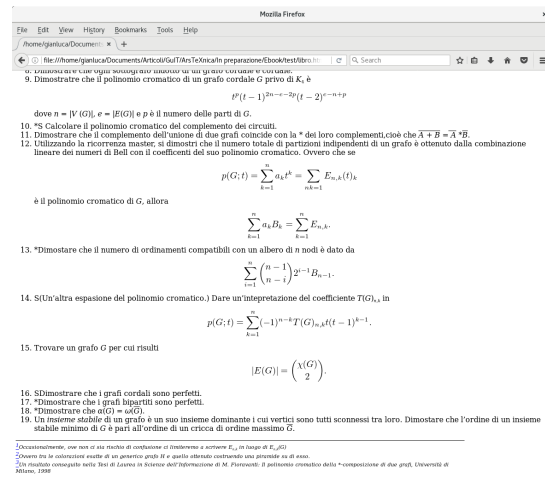


FIGURA 2: Parte finale del documento didattico trasformato in HTML. Le formule sono rese come figure.

Siamo pronti a compilare il sorgente `LATEX` per ottenere un HTML. Abbiamo l'accortezza di eliminare l'inclusione di `crop` commentando la riga 13 del listato 1 e, come visto poc'anzi, di sostituire le figure in `TikZ` con le corrispondenti raster. Sia questo file `libro.tex`:

```
htlatex libro "html,fn-in"
```

ci fa ottenere, dopo aver risposto s alle eventuali interruzioni per segnalare un errore, un file HTML con tutte le note raggruppate alla fine (senza l'opzione `"html,fn-in"` otterremmo una nota per pagina HTML — in pratica, un file HTML per nota — e, nel caso di testi con molte note, questa non è la soluzione più agevole); parte di questo file è mostrata nella figura 2.⁶

Ci sono comandi che purtroppo non si riescono a tradurre in HTML, forse solo per incompletezza del compilatore. Per esempio, sarebbe facile tradurre `\`, col *thin space* di Unicode (codice U+2009) o anche solo con uno spazio, ma ciò non avviene; per evitare di avere parole attaccate dobbiamo eliminare le occorrenze di `\`, dal sorgente. Anche `\raggedleft` non viene convertito; al suo posto ci conviene usare l'ambiente `flushright`.

Sono solito aggiustare i titoli, specie nel sommario, usando `\protect\`. Questo funziona, ma non sono soddisfatto della resa nel sommario dell'HTML, quindi preferisco eliminare le occorrenze di questa sequenza di comandi e lasciare che i titoli vadano a capo quando non hanno più spazio.

scelta infelice di linguaggio è volta a distinguere i programmi e le app, installabili su diversi sistemi operativi, dai lettori dedicati.

6. Avremmo potuto dare il comando `htlatex libro "html,mathml"` col risultato di ottenere un file che il mio Firefox mostra solo fino a un certo punto, pur essendo il sorgente completo, e che, una volta convertito in EPUB, mostra le formule extratestuali in maniera a dir poco fantasiosa.

3 Intervallo: L'HTML è da aggiustare

Quando abbiamo un HTML soddisfacente, cioè che non necessita di essere più rigenerato modificando il sorgente `.tex`, è molto probabile che ci siano piccoli aggiustamenti da fare. I due che faccio riguardano due aspetti. Innanzitutto le spaziature verticali del frontespizio, con l'aggiunta di opportuni tag `
`, e la separazione del frontespizio dal retrofrontespizio (inserisco un titolo fantasma: `<h2></h2>`) che altrimenti starebbero sulla stessa pagina. Poi il riposizionamento delle note a piè di pagina. Come abbiamo visto, `htlatex` può essere regolato per mettere le note in fondo al documento HTML, ma il mio datore di lavoro principale le vuole alla fine di ogni capitolo. Per ottenere questo risultato devo tagliare tutte le righe delle note di un capitolo (con l'aiuto di `bc` e di opportune etichette per sapere quante righe tagliare e ritrovare agevolmente il punto in cui tagliare) e copiarle alla fine del relativo capitolo; all'inizio della prima nota di ogni capitolo devo scrivere il tag `<div class="footnotes">` e, alla fine dell'ultima, `</div>`. Fare questo lavoro con le note poste ognuna in un file separato sarebbe stato molto peggio.

Una cosa a cui prestare attenzione è il fatto che i capitoli non numerati non azzerano la numerazione delle note ma la proseguono. Lo stesso avviene con `htlatex`. Naturalmente tutti sanno che basta riazzere il contatore `footnote` per evitare la numerazione incoerente. Vediamo cosa succede nell'HTML con un MWE (Minimal Working Example):

```

1 \documentclass[a4paper,11pt]{
  book}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage[italian]{babel}
5 \usepackage{lipsum}
6 \begin{document}
7 \chapter*{Prefazione}
8 \addcontentsline{toc}{chapter}{
  Prefazione}
9 \lipsum[1]\footnote{Prima nota.}
10
11 \lipsum[2]\footnote{Seconda nota
  .}
12 \chapter*{Introduzione}
13 \addcontentsline{toc}{chapter}{
  Introduzione}
14 \setcounter{footnote}{0}
15 \lipsum[3]\footnote{Prima nota.}
16
17 \lipsum[4]\footnote{Seconda nota
  .}
18 \chapter{Uno}
19 \lipsum[5]\footnote{Prima nota.}
20
21 \lipsum[6]\footnote{Seconda nota

```

```

  .}
22 \chapter*{Conclusioni}
23 \addcontentsline{toc}{chapter}{
  Conclusioni}
24 \setcounter{footnote}{0}
25 \lipsum[7]\footnote{Prima nota.}
26
27 \lipsum[8]\footnote{Seconda nota
  .}
28 \end{document}

```

Il file HTML risultato della compilazione contiene una serie di link dal testo alle note e viceversa. Vediamo uno per uno i link alle note analizzandone i nomi:

CAPITOLO	PRIMA NOTA	SECONDA NOTA
Prefazione	fn1x0	fn2x0
Introduzione	fn1x0	fn2x0
Uno	fn1x1	fn2x1
Conclusioni	fn1x1	fn2x1

Da questo capiamo che i richiami a nota di Prefazione e Introduzione puntano entrambi alle stesse note (di Prefazione) e quelli di Uno e Conclusioni puntano alle stesse note (di Uno). Questo succede perché il capitolo numerato incrementa un contatore delle note (la parte `x(numero)`), cosa che non fa un capitolo non numerato. Se non avessimo riazzato il contatore delle note, non avremmo avuto di questi problemi e avremmo potuto semplicemente correggere a mano i soli numeri dei riferimenti alle note nell'HTML.

A seconda delle lingue impiegate per produrre l'HTML, capita che la codifica non rispecchi il contenuto del file e il browser presenti dei caratteri "fantasiosi". In questo caso è sufficiente cambiare la codifica iniziale dell'HTML (cosa da fare solo in questa fase e mai quando l'HTML è da rigenerare continuamente):

```

<meta http-equiv="Content-Type"
  content="text/html;
  charset=iso-8859-1">

```

presente subito dopo la riga contenente `<head><title>` può presentare un charset inadatto che va verificato ed eventualmente corretto in accordo con la codifica del testo. In alcuni casi ho dovuto usare anche `iconv` per adattare la codifica dell'intero testo.

Codifica giusta o meno, mi sono capitati casi in cui, soprattutto negli indici analitici, le lettere accentate fossero totalmente sbagliate. Uno script si occupa di trattare i casi specifici, per esempio:

```

sed -i 's/\^c3\^ba/\&uacute;/g'\
  $file.html

```

corregge tutte le occorrenze di `^c3^ba` in ú.

Un'altra cosa di cui tener conto, specie quando il documento contiene del testo in greco politonico, è la mancata “traduzione” di alcuni caratteri e la loro conversione a immagine. Controllate i file generati dalla compilazione con `htlatex` e guardate le immagini PNG. Se ne trovate qualcuna contenente un solo carattere, potete cercare nell'HTML il punto in cui ognuna di esse è caricata e sostituire il relativo codice col codice del carattere in questione (UNICODE CONSORTIUM, 2018, è fondamentale avere a portata di mano le tabelle dei caratteri Unicode).

Nel caso in esame, cioè il progetto didattico, l'HTML non ha bisogno di ulteriori aggiustamenti.

4 Secondo tempo: da HTML a EPUB (o MOBI)

Una volta che siamo soddisfatti dell'HTML, possiamo generare l'ebook. Di questo si occuperanno Calibre e i suoi comodissimi filtri da riga di comando (GOYAL, 2018):

```
ebook-convert <file.html> <file.epub> --cover
<jpeg della copertina> --no-default-epub-cover
--title <titolo> --authors <autore> --publisher
<editore>
```

se vogliamo un file `.epub`,

```
ebook-convert <file.html> <file.mobi> --cover
<jpeg della copertina> --no-inline-toc --title
<titolo> --authors <autore> --publisher <editore>
```

se vogliamo un file `.mobi`.⁷

Per generare l'EPUB diamo il seguente comando (tutto di seguito, anche se qui va a capo per motivi di spazio) dal terminale:

```
ebook-convert libro.html libro.epub
--cover copertina_dantona.jpg
--no-default-epub-cover
--title "Introduzione alla matematica
discreta"
--authors "Ottavio D'Antona"
--publisher Apogeo
```

Ecco come Calibre mostra il libro generato nella biblioteca interna (figura 3) e come leggiamo l'ebook (figura 4).

Durante il processo di conversione da HTML a EPUB Calibre ci informa che, dopo “Splitting markup on page breaks and flow limits, if any...” e “Looking for large trees in libro.html...”, “No large trees found”, cioè in questo documento non ci sono capitoli talmente lunghi da dover essere ulteriormente suddivisi.

7. Il servizio Kindle Direct Publishing di Amazon accetta senza remore un EPUB (senza copertina) e un file di copertina per generare da sé un file `.mobi`. Non ho mai approfondito la questione della validazione del formato MOBI, che sembra essere una faccenda del tutto proprietaria. In ogni caso, tenete presente che tale formato non supporta MathML per definizione.

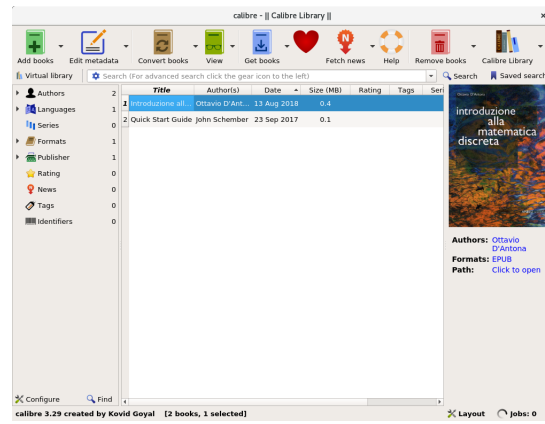


FIGURA 3: L'ebook è stato importato in Calibre, pronto per essere letto.

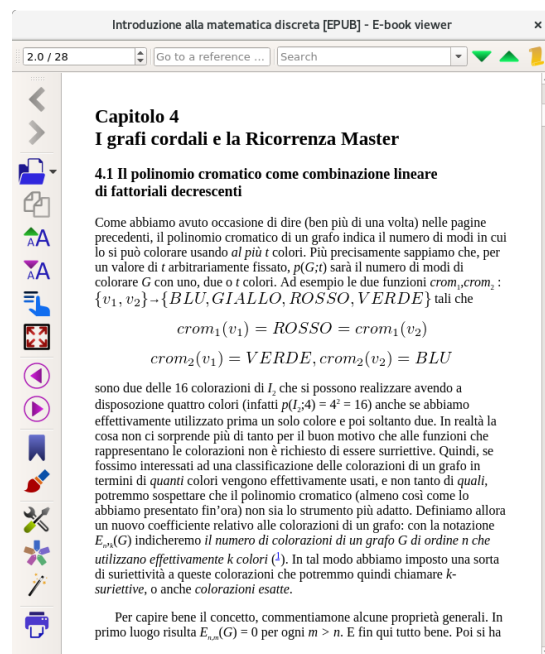


FIGURA 4: Il lettore di Calibre mostra l'ebook al suo meglio.

Può capitare che gli *split*⁸ in cui viene suddiviso il file HTML per ottenere l'EPUB non possano contenere un intero capitolo, e quindi ci sia più di uno split per capitolo. Questo provoca il fastidioso effetto di vedere un titolo di livello inferiore al capitolo cominciare in una pagina nuova, magari dopo un bel bianco. Qui viene in aiuto Sigil, che permette di fare il merge degli split, ma con potenziali problemi di cui parleremo nella sezione 5.

Recentemente mi è capitato un caso che a tutta prima mi sembrava un bug di Calibre: la generazione dell'EPUB si interrompeva con errori a causa di un *large tree* (un capitolo lungo) che Calibre non riusciva a suddividere. Dopo aver inutilmente

8. Ogni *split* contiene un elemento che inizia con un titolo, reale o fantasma, di qualunque livello. La suddivisione di default è uno split per capitolo.

aggiornato Calibre mi sono reso conto del motivo vero: il capitolo conteneva una lunghissima bibliografia, in sostanza una lunga lista, che non poteva essere suddivisa per mancanza di titoli. L'elenco infatti non prevedeva sezioni tematiche ma solo una lista alfabetica, lunga ed estenuante.

La soluzione è consistita nell'interrompere manualmente (editando l'HTML) la lista raggruppando tutti gli autori con la stessa iniziale in una lista a sé e, tra la fine di una lista e l'inizio della successiva, mettere un titolo fantasma (nel caso specifico, `<h3></h3>`). Questo ha permesso di ottenere diversi split che poi è bastato riunire col merge di Sigil.

5 Attualità: validazione dell'EPUB

L'ultima fase della produzione di un ebook consiste nella sua validazione, cioè nel controllo che tutti gli elementi del markup siano conformi allo standard. In tale fase è fondamentale `epubcheck`. Questo è un file Java che, ricevuto in input il file EPUB da validare, restituisce in output una lista di errori (con relativi numeri di riga e colonna) o la scritta "Nessun errore riscontrato". Nel caso in cui ci siano errori da correggere (evento tutt'altro che raro), trovo utilissimo usare Sigil come editor in luogo del mio prediletto `vi`.

Torniamo al progetto didattico: Calibre non ha fatto una piega nell'aprire e mostrarci l'EPUB ottenuto, ma siamo sicuri che potremmo metterlo in vendita senza problemi? Per saperlo dobbiamo validarlo ricorrendo a `epubcheck`:

```
java -jar /opt/epubcheck-4.0.2/
epubcheck.jar libro.epub
```

(sul mio sistema il `.jar` si trova sotto `/opt/epubcheck-4.0.2/`). Il responso è un insieme di sei errori, tutti del tipo "element `"table"` not allowed here; expected the element end-tag, text or element `"a"`, `"abbr"`, `"acronym"`, `"applet"`, `"b"`, `"bdo"`, `"big"`, `"br"`, `"cite"`, `"code"`, `"del"`, `"dfn"`, `"em"`, `"i"`, `"iframe"`, `"img"`, `"ins"`, `"kbd"`, `"map"`, `"noscript"`, `"ns:svg"`, `"object"`, `"q"`, `"samp"`, `"script"`, `"small"`, `"span"`, `"strong"`, `"sub"`, `"sup"`, `"tt"` or `"var"` (with `xmlns:ns="http://www.w3.org/2000/svg"`)" in diversi punti di `libro.epub`.

Per tentare di correggerli apriamo l'ebook con Sigil. La prima cosa che Sigil segnala tramite una finestra di dialogo è il fatto che l'EPUB ha il relativo HTML mal formato e può provare a correggere i file automaticamente, sebbene ciò potrebbe causare una perdita di dati. Proviamo a rispondere affermativamente alla richiesta di correzione automatica: se va bene, siamo fortunati; se va male, possiamo sempre ripartire dalla conversione HTML → EPUB e fare il resto manualmente.

In effetti, Sigil si apre mostrando i due split (copertina e unico capitolo) il cui contenuto sembra tutto a posto. (Per scrupolo ho provato ad aprire l'ebook senza farlo correggere e tutte le formule incluse come immagini e le immagini dei grafi non venivano visualizzate da Sigil; Calibre non aveva problemi di sorta. La tabella aveva un aspetto diverso da quello atteso.) Se salviamo e riproviamo la validazione non otteniamo più errori. In questo caso siamo stati molto fortunati. Nel 100% delle situazioni lavorative non sono stato altrettanto fortunato. Infatti, la procedura a cui di solito sottopongo gli ebook in lavorazione è: 1) apertura dell'EPUB con Sigil (che in genere non riscontra errori); 2) controllare ogni messaggio di errore di `epubcheck` e i relativi numeri di split, di riga e di colonna;⁹ 3) aprire lo split e visualizzarne il codice HTML; 4) tentare di correggere l'errore modificando il codice, magari facendo ricorso ai manuali dello standard per capire la natura dell'errore o la sua soluzione¹⁰. Se la successiva validazione (ricordiamoci di salvare il file modificato) non dà più l'errore, la correzione è riuscita.

L'errore che in assoluto mi capita più spesso (e che mi fa pensare al perché negli ambienti commerciali Calibre è visto in malo modo come strumento inaffidabile) è il posizionamento di un riferimento ipertestuale in un punto non consentito. Nella figura 5 ho evidenziato la linea 17 del sorgente di un ebook di recente realizzazione. Quella linea è corretta ma il file originario conteneva

```
<h2 class="calibre2" id="calibre_pb_3">
Indice</h2>
<a id="x1-1000" class="calibre3"></a>
```

e il validatore diceva che un tag `<a>` in quel punto non è ammesso. Da lì lo spostamento nel punto mostrato nella figura 5.

Sempre nella figura 5, sulla sinistra, vediamo l'elenco degli split in cui è suddiviso il libro. Dopo lo split 7 c'è il 12. Ciò vuol dire che abbiamo attaccato insieme (merge) gli split 7–11. Questo causa un errore in `toc.ncx` perché i riferimenti agli split ivi contenuti non vengono corretti. Dunque `toc.ncx` continua a contenere i riferimenti agli split 8–11 che ora non esistono più. Dovremo aprire il codice del sommario e correggere a mano i riferimenti non più esistenti, scrivendo `split_007` dove compare `split_008-split_011`. Naturalmente il validatore ci assiste anche per questo tipo di errore.

9. A volte succede che le coordinate di riga e colonna del primo errore siano sbagliate — lo split non lo è mai — e che, di conseguenza, lo siano tutte le altre. Corretto il primo errore, la nuova validazione darà tutte le coordinate corrette.

10. Ci può venire in soccorso il sito INTERNATIONAL DIGITAL PUBLISHING FORUM (2018), unitamente a W3C (2018).

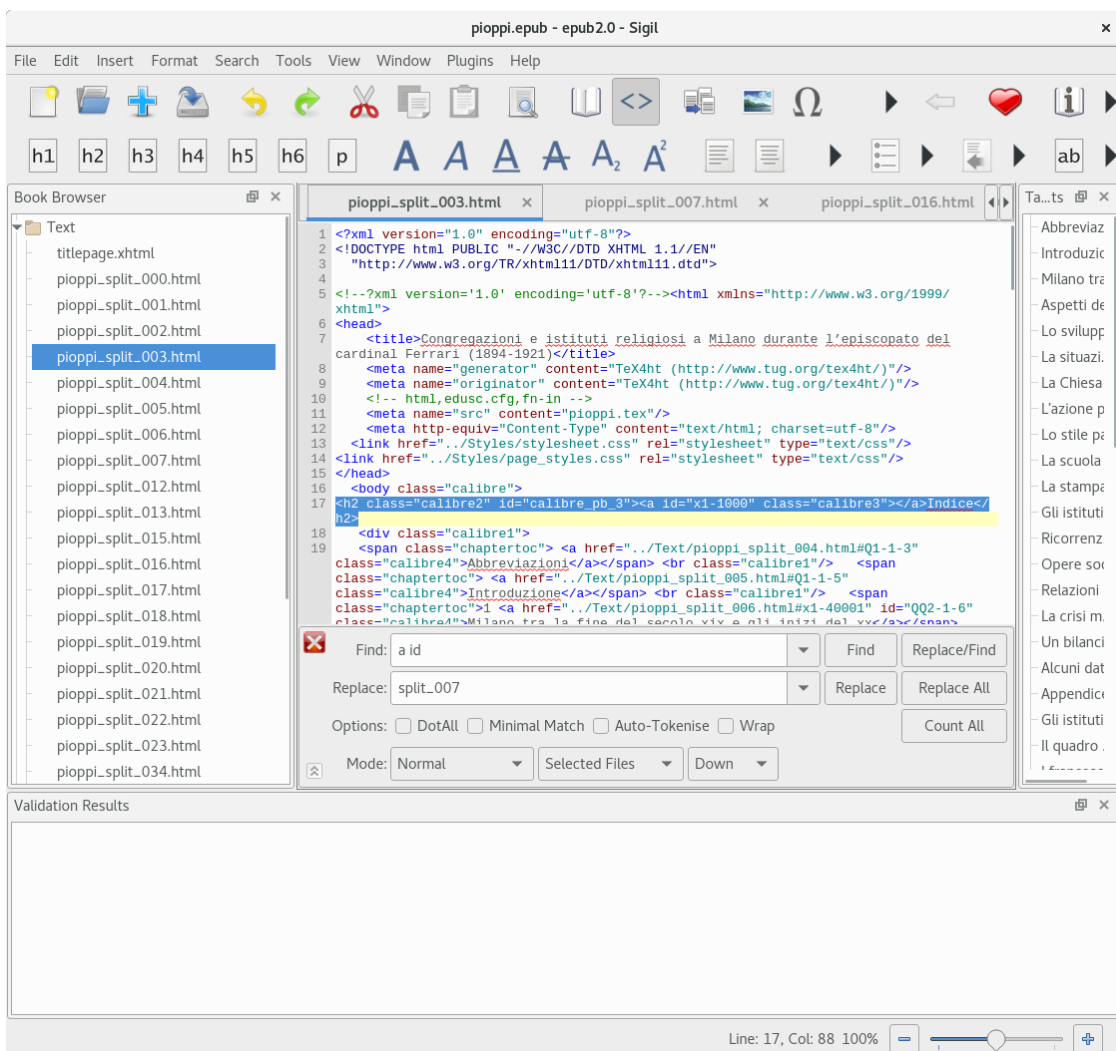


FIGURA 5: Elenco degli split e codice di uno di essi così come mostrati da Sigil.

6 Un editor di ebook alternativo

Finora ho parlato di Sigil perché è veramente lo strumento che uso. Da qualche giorno sto valutando la sua sostituzione con l'editor di Calibre (`ebook-edit`). Mi pare opportuno elencare brevemente alcune ragioni per usare l'uno o l'altro.

Sigil permette di editare l'ebook sia modificando l'HTML che modificando direttamente la versione visuale. Però consente di visualizzare alternativamente nell'editor l'ebook o il suo codice sorgente, visualizzazione che cambiamo agendo sullo switch dedicato. Quando apriamo un file EPUB viene comunque mostrato il contenuto del primo split e le barre dei pulsanti – quella delle operazioni sul file e quella di editing – sono sempre presenti. Non è possibile chiudere tutti gli split: uno di essi rimane comunque aperto nell'editor. Il navigatore, a sinistra, mostra tutti gli split, gli stili, le immagini e tutto quanto componga l'ebook. Come già detto, il validatore adottato è il sorpassato FlightCrew,

in genere molto generoso nel perdonare elementi attualmente fuori standard.

L'editor di Calibre, `ebook-edit`, permette la sola modifica dell'HTML ma, siccome visualizza il codice sorgente nell'editor parallelamente a un'anteprima dell'ebook, le modifiche si propagano dall'editor all'anteprima in tempo quasi reale.¹¹ L'unica interazione che possiamo avere nella finestra di anteprima è evidenziare un brano di testo per fare in modo che il cursore si posizioni all'inizio dei relativi tag nell'editor del codice. Quando apriamo per la prima volta un file EPUB l'editor è vuoto e solo la finestra del navigatore ci mostra il contenuto dell'ebook: split, stili, immagini ecc. La barra dei pulsanti mostra solo quelli relativi alle operazioni generali sul file. La barra dei pulsanti di editing

11. Ricordiamo che il concetto di *tempo reale* è un concetto mutuato dai sistemi operativi che indica la certezza di avere una determinata operazione completata entro un tempo limite non dilazionabile per alcuna ragione. Non ha niente a che vedere con la volgarizzazione dell'espressione che vorrebbe significare la contemporaneità.

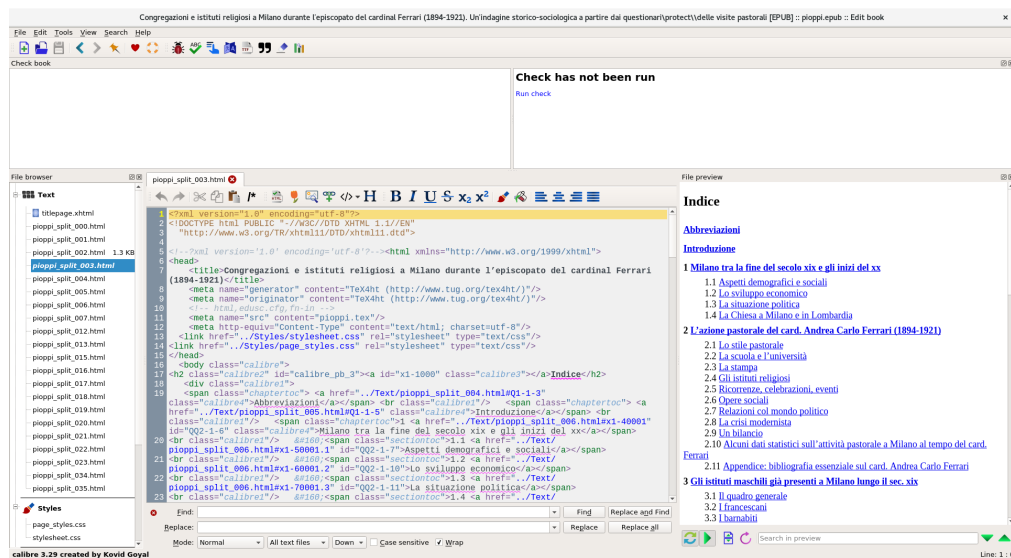


FIGURA 6: L'editor di ebook di Calibre.

è visibile solo quando apriamo uno split per l'editing ed è posta nella stessa scheda del codice dello split. Anche `ebook-edit` ha un validatore non specificato nei manuali ma che, genericamente, è dichiarato conforme all'HTML 5. Nella figura 6 vediamo l'interfaccia di `ebook-edit` all'opera sullo stesso documento della figura 5.

Personalmente, non usando l'editor visuale di Sigil (che trovo alquanto incomprensibile), sto valutandone l'abbandono in favore di `ebook-edit`. Altri utenti potrebbero addurre ragioni contrarie e fare la scelta opposta alla mia.

7 Conclusioni

Produrre un ebook in formato EPUB a partire da un file L^AT_EX non è un'operazione immediata né economica: serve tempo per ognuna delle fasi di lavorazione, serve aver accumulato esperienza per risolvere agevolmente gli intoppi sempre possibili, e serve avere almeno una cosiddetta *working knowledge* degli strumenti e dei linguaggi coinvolti per non aver voglia di cambiare mestiere.

A queste oggettive difficoltà fa da contraltare il controllo assoluto sul prodotto finale: possiamo cambiare l'HTML, il CSS e l'EPUB fino a ottenere il prodotto finale che meglio corrisponde alle nostre aspettative e alle esigenze del committente. L'ulteriore vantaggio è che l'EPUB così generato tende a essere più piccolo degli EPUB generati — pur in meno tempo — con i più noti prodotti commerciali.

Riferimenti bibliografici

D'ANTONA, O. (1998). *Introduzione alla matematica discreta*. Apogeo scientifica. Apogeo, Milano.

EPUBCHECK TEAM (2018). «GitHub - IDPF/epubcheck: Validation tool for EPUB». URL <https://github.com/idpf/epubcheck>.

GOYAL, K. (2018). «Command line interface — calibre 3.30.0 documentation». URL <https://manual.calibre-ebook.com/generated/en/cli-index.html>.

GURARI, E. M. (2004). «T_EX4ht: HTML Production». *TUGboat*, **25** (1). Proceedings of the Practical T_EX 2004 Conference.

HOFTICH, M. (2016). «Github - michal-h21/luatex: experimental package for direct conversion from latex to html using tex4ht and luatex». URL <https://github.com/michal-h21/luatex>.

INTERNATIONAL DIGITAL PUBLISHING FORUM (2018). «The Trade and Standards Organization for the Digital Publishing Industry». URL <http://idpf.org/>.

PIGNALBERI, G. (2015). «L^AT_EX in un'editrice universitaria: come e perché book non è adatta e una possibile alternativa funzionante». *ArsTeXnica*, (19), pp. 33–41.

SCHEMBER, J. (2018). «Sigil Ebook». URL <https://sigil-ebook.com/>.

UNICODE CONSORTIUM (2018). «Unicode 11.0 Character Code Charts». URL <https://www.unicode.org/charts/>.

W3C (2018). «HTML5 Reference». URL <https://dev.w3.org/html5/html-author/>.

▷ Gianluca Pignalberi
g dot pignalberi at gmail dot com