

Numero 19
Aprile 2015

ArsT_EXnica

Rivista italiana di T_EX e L^AT_EX

GU_{IT}

<http://www.guitex.org/arstexnica>



G_UI_T – Gruppo Utilizzatori Italiani di T_EX

ArsT_EXnica è la pubblicazione ufficiale del G_UI_T

Comitato di Redazione

Claudio Beccari – *Direttore*

Roberto Giacomelli – *Comitato scientifico*

Enrico Gregorio – *Comitato scientifico*

Ivan Valbusa – *Comitato scientifico*

Lorena Rachele Badile, Renato Battistin,

Riccardo Campana, Massimo Caschili,

Gustavo Cevolani, Massimiliano Dominici,

Tommaso Gordini, Carlo Marmo,

Gianluca Pignalberi, Ottavio Rizzo,

Gianpaolo Ruocco, Enrico Spinielli,

Emiliano Vavassori

ArsT_EXnica è la prima rivista italiana dedicata a T_EX, a L^AT_EX ed alla tipografia digitale. Lo scopo che la rivista si prefigge è quello di diventare uno dei principali canali italiani di diffusione di informazioni e conoscenze sul programma ideato quasi trent'anni fa da Donald Knuth.

Le uscite avranno, almeno inizialmente, cadenza semestrale e verranno pubblicate nei mesi di Aprile e Ottobre. In particolare, la seconda uscita dell'anno conterrà gli Atti del Convegno Annuale del G_UI_T, che si tiene in quel periodo.

La rivista è aperta al contributo di tutti coloro che vogliano partecipare con un proprio articolo. Questo dovrà essere inviato alla redazione di ArsT_EXnica, per essere sottoposto alla valutazione di revisori. È necessario che gli autori utilizzino la classe di documento ufficiale della rivista; l'autore troverà raccomandazioni e istruzioni più dettagliate all'interno del file di esempio (.tex). Tutto il materiale è reperibile all'indirizzo web della rivista.

Gli articoli potranno trattare di qualsiasi argomento inerente al mondo di T_EX e L^AT_EX e non dovranno necessariamente essere indirizzati ad un pubblico esperto. In particolare tutorials, rassegne e analisi comparate di pacchetti di uso comune, studi di applicazioni reali, saranno bene accetti, così come articoli riguardanti l'interazione con altre tecnologie correlate.

Di volta in volta verrà fissato, e reso pubblico sulla pagina web, un termine di scadenza per la presentazione degli articoli da pubblicare nel numero in preparazione della rivista. Tuttavia gli articoli potranno essere inviati in qualsiasi momento e troveranno collocazione, eventualmente, nei numeri seguenti.

Chiunque, poi, volesse collaborare con la rivista a qualsiasi titolo (recensore, revisore di bozze, grafico, etc.) può contattare la redazione all'indirizzo:

arstexnica@guitex.org.

Nota sul Copyright

Il presente documento e il suo contenuto è distribuito con licenza  Creative Commons 2.0 di tipo “Non commerciale, non opere derivate”. È possibile, riprodurre, distribuire, comunicare al pubblico, esporre al pubblico, rappresentare, eseguire o recitare il presente documento alle seguenti condizioni:

- ① **Attribuzione:** devi riconoscere il contributo dell'autore originario.
- ② **Non commerciale:** non puoi usare quest'opera per scopi commerciali.
- ③ **Non opere derivate:** non puoi alterare, trasformare o sviluppare quest'opera.

In occasione di ogni atto di riutilizzazione o distribuzione, devi chiarire agli altri i termini della licenza di quest'opera; se ottieni il permesso dal titolare del diritto d'autore, è possibile rinunciare ad ognuna di queste condizioni.

Per maggiori informazioni:

<http://www.creativecommons.org>

Associarsi a G_UI_T

Fornire il tuo contributo a quest'iniziativa come membro, e non solo come semplice utente, è un presupposto fondamentale per aiutare la diffusione di T_EX e L^AT_EX anche nel nostro paese. L'adesione al Gruppo prevede una quota di iscrizione annuale diversificata: 30,00 € soci ordinari, 20,00 (12,00) € studenti (junior), 75,00 € Enti e Istituzioni.

Indirizzi

Gruppo Utilizzatori Italiani di T_EX

c/o Università degli Studi di Napoli Federico II

Dipartimento di Ingegneria Industriale

Via Claudio 21

80125 Napoli – Italia

<http://www.guitex.org>

guit@sssup.it

Redazione ArsT_EXnica:

<http://www.guitex.org/arstexnica/>

arstexnica@guitex.org

Codice ISSN 1828-2369

Stampata in Italia

Napoli: 15 Aprile 2015

ArsT_EXnica

Rivista italiana di T_EX e L^AT_EX

Numero 19, Aprile 2015

Claudio Beccari	
Editoriale	5
Frank Mittelbach	
Come si può influenzare la posizione degli ambienti galleggianti come figure e table in L ^A T _E X?	6
Barbara Beeton	
Una figura larga sotto un testo a due colonne	14
Claudio Beccari	
\vadjust, questo sconosciuto	15
James R. Hunt	
Comporre liste: un viaggio nella grammatica sconosciuta	19
Claudio Fiandrino	
Personalizzazione e riproducibilità nel disegno programmato: introdu- zione al pacchetto pgfkeys	27
Gianluca Pignalberi	
L ^A T _E X in un'editrice universitaria: come e perché book non è adatta e una possibile alternativa funzionante	33

Gruppo Utilizzatori Italiani di T_EX

Editoriale

Claudio Beccari

*ArsTeX*nica ci offre un'altra novità: in questo numero di aprile 2015 abbiamo solo 6 articoli, ma per la prima volta, con il permesso degli autori e quello di *TUGboat*, abbiamo tre articoli già pubblicati su *TUGboat*; la Redazione li ha considerati di rilievo sia per gli argomenti trattati sia per la preminenza degli autori.

Comincerò da questi tre articoli in ordine più o meno cronologico.

James R. Hunt ha insegnato matematica per anni, poi si è occupato professionalmente di statistica, per dedicarsi quindi alla scrittura tecnica, specialmente nel campo dell'informatica. Nell'articolo qui pubblicato espone l'aspetto stilistico e letterario delle liste, siano esse le enumerazioni (liste numerate) o le elencazioni (liste puntate) o le liste di definizioni: le tre liste principali di $\text{\LaTeX}$. La sua impostazione è basata sulla grammatica e su un certo numero di postulati che ritiene validi nel campo della prosa tecnico-scientifica. Non sono, quindi, postulati validi in generale ma a nostro parere molto ragionevoli. Personalmente ho già rivisto tutte le guide tematiche che ho scritto per il nostro gruppo.

Frank Mittelbach è un matematico e informatico; dagli anni '80 si occupa di $\text{\LaTeX}$. È il coordinatore del $\text{\LaTeX}$ 3-Project e responsabile di $\text{\LaTeX}$ 2 ϵ ; è anche l'autore principale di entrambe le edizioni del $\text{\LaTeX}$ Companion. Tra le altre cose si è occupato della riscrittura della routine di output nel passaggio da $\text{\LaTeX}$ 2.09 a $\text{\LaTeX}$ 2 ϵ . È proprio di questo che parla nel suo articolo focalizzato sulla collocazione degli oggetti "flottanti". Egli ci svela cose che discutiamo spesso nel forum del nostro gruppo. Anche noi abbiamo cercato di esporle come meglio siamo riusciti a fare nelle guide generali e nelle guide tematiche, ma il racconto di chi ha scritto quelle procedure permette di percepire la filosofia seguita nello scegliere le soluzioni realizzate.

Sempre su questo argomento interviene Barbara Beeton, la direttrice storica di *TUGboat* e curatrice del sistema $\text{\TeX}$ nella sua purezza grazie al contatto diretto e continuo con Donald E. Knuth. Cura tutto quanto serve agli utenti del sistema $\text{\TeX}$ e, tra le altre cose, ha fatto parte di un gruppo di

lavoro addetto a definire i code point della codifica UNICODE per le pagine riguardanti la composizione della matematica. In questo breve articolo di una sola pagina, mostra come aggirare certe limitazioni relative alla composizione a due colonne in merito alla collocazione degli oggetti flottanti. La soluzione di Barbara Beeton ha ispirato quella usata in almeno tre articoli di questo numero di *ArsTeX*nica per pareggiare l'ultima pagina.

Gianluca Pignalberi ci offre le sue esperienze per costruire un sistema modulare di classi e pacchetti per gestire alcune collane di pubblicazioni di una casa editrice di Roma. Sono esperienze preziose che condivide con noi tramite il suo articolo. Ci siamo passati in molti, ma direi che Pignalberi è andato più lontano di altri: si è preso un impegno oneroso e strada facendo ha sviluppato una architettura modulare che potrebbe ispirare molti dei creatori di classi. Con piacere modificherò la guida tematica sulla creazione di classi che da qualche anno è disponibile nel sito del $\text{\GJ}$, che alla luce di questo articolo è diventata obsoleta.

Claudio Fiandrino, dottorando in Lussemburgo, ben noto ai frequentatori del Forum $\text{\GJ}$ e autore di alcune guide tematiche, presenta invece un'altra preziosa applicazione di PGF, quella della gestione delle opzioni mediante la sintassi *chiave = valore* in una maniera più semplice di quello che si può fare con altri rinomati pacchetti. La struttura delle chiavi di PGF non serve solo per *TikZ*, ma per qualsiasi classe o pacchetto; meglio ancora, per qualsiasi documento.

Il mio articolo spiega un comando nativo del programma $\text{\TeX}$; può servire per fare molte cose, anche molto complesse, ma sono mostrati tre piccoli esempi alla portata di tutti.

Auguro quindi buona lettura; gli articoli di questo numero di *ArsTeX*nica sono molto istruttivi.

▷ Claudio Beccari
Professore emerito
Politecnico di Torino
claudio dot beccari at gmail
dot com

Come si può influenzare la posizione degli ambienti galleggianti come `figure` e `table` in $\text{\LaTeX}$?*

Frank Mittelbach

Sommario

Nel 2012, su $\text{\TeX}$ - $\text{\LaTeX}$ Stack Exchange venne posta la domanda «How to influence the position of float environments like `figure` and `table` in $\text{\LaTeX}$?» (DANIEL, 2012), e siccome in precedenza erano comparse numerose altre domande su quest'argomento, ho deciso di trattarlo in profondità e di spiegare la maggior parte dei misteri che il meccanismo sottostante presenta a chi cerca di usarlo con successo.

Da quando la mia risposta è apparsa sul web, si è chiesto di vederla trasformata in un articolo e io risposi stupidamente «solo se questa risposta finisce per diventare una ‘grande’ risposta» (cioè se ottiene 100 voti). Al momento di scrivere questo articolo la risposta ha collezionato 222 voti, perciò sarà meglio mantenere la promessa.

Abstract

In 2012, a question “How to influence the float placement in $\text{\LaTeX}$ ” was asked on $\text{\TeX}$.stackexchange and as there had been many earlier questions around this topic I decided to treat the topic in some depth and explain most of the mysteries that the underlying mechanism poses to people trying to use it successfully.

Once my answer appeared on the web, people asked to see this converted into an article and I foolishly replied “only if this answer ends up becoming a ‘great’ answer” (gets 100 votes). At the time of writing this article, the answer stands at 222 votes, so I had better make good on that promise.

Nota dei traduttori

Nella traduzione si è deciso di lasciare il termine *float* in inglese per la mancanza di un corrispettivo traduttore italiano sintetico. In pratica, con *float* si intende uno dei due ambienti mobili standard di

*Questo articolo è una traduzione autorizzata dall'autore della risposta a (DANIEL, 2012) (pubblicata originariamente su $\text{\TeX}$ - $\text{\LaTeX}$ Stack Exchange il 21 dicembre 2011) eseguita da Claudio Beccari e Tommaso Gordini, unici responsabili di eventuali travisamenti del testo originale. Frank Mittelbach vive in Germania e ha studiato matematica e informatica. Si occupa di $\text{\LaTeX}$ dalla metà degli anni '80 del Novecento ed è autore e coautore di numerosi pacchetti, manuali e pubblicazioni su questo programma. Attualmente è direttore tecnico del $\text{\LaTeX}$ 3 Project e supervisore di $\text{\LaTeX}$ 2_ε.

$\text{\LaTeX}$ (`figure` o `table`) o gli ambienti mobili definiti da certe classi di documento o certi pacchetti di estensione. Con *oggetto* intendiamo il contenuto di un float: un'immagine, una tabella, un disegno, un grafico, eccetera. Per *composizione a due colonne* si intende il risultato dell'opzione di classe `twocolumn` o del comando `\twocolumn` e non quello possibile grazie al pacchetto `multicol`. Con l'espressione *colonna di testo* si intende l'area della pagina corrispondente a `\columnwidth` (che nella composizione a una sola colonna coincide con `\textwidth`). Con *area di float* si intende lo spazio della pagina destinato a ospitare i float. Infine, con *float a giustezza piena* si intende un oggetto largo quanto due colonne nella composizione a due colonne.

1 Introduzione

Per rispondere a questa domanda, bisogna prima comprendere le regole fondamentali che governano la collocazione standard dei float sulla pagina in $\text{\LaTeX}$. Capite queste, si possono operare aggiustamenti, per esempio cambiando i parametri dei float o caricando certi pacchetti che modificano o estendono le funzionalità di base del programma.

2 Terminologia

2.1 Categorie di float

In $\text{\LaTeX}$, ogni float appartiene a una categoria. Per impostazione predefinita $\text{\LaTeX}$ ne conosce due, corrispondenti agli ambienti mobili standard `figure` e `table`. Non sono le uniche: altre classi di documento e altri pacchetti possono aggiungerne di nuove. La categoria di appartenenza influenza alcuni aspetti della collocazione del float sulla pagina, come la sua specificazione di collocazione predefinita (a meno che non venga sovrascritta dal float stesso nei modi descritti nella sezione 2.3).

Un'importante proprietà dell'algoritmo di collocazione dei float¹ è che $\text{\LaTeX}$ non viola mai l'ordine della disposizione all'interno di una categoria. Per esempio, se in un documento si ha la serie ‘Figura 1, Tabella 1, Figura 2’, la ‘Figura 1’ verrà posata sempre prima della ‘Figura 2’. Tuttavia, la ‘Tabella 1’ (appartenendo a un'altra categoria) sarà gestita indipendentemente e quindi nel documento finito potrà comparire prima, dopo o tra le due figure.

1. D'ora in poi *algoritmo*. (N.d.T.)

2.2 Aree di float

In una colonna di testo, L^AT_EX conosce due aree in cui può mettere i float: l'area superiore e l'area inferiore. Nella composizione a due colonne, conosce anche un'area superiore che si estende su entrambe le colonne per float larghi l'intera pagina, mentre non esiste una corrispondente area inferiore.

Inoltre, L^AT_EX è in grado di produrre colonne e pagine che contengono solo float.²

Infine, L^AT_EX è capace di mettere i float a metà pagina intercalato nel testo (ma solo se di volta in volta si guida la posa di ogni singolo float in questo senso).

2.3 Codici di posizione

Per indirizzare la posa di un float in una di queste aree, gli si può fornire un *codice di posizione*³ come argomento facoltativo. Se tale indicazione non viene espressa, L^AT_EX userà un codice predefinito (che, come s'è detto sopra, dipende dalla categoria di appartenenza ma di solito permette all'oggetto di andare in tutte le aree a meno di altre restrizioni).

Un codice può consistere dei seguenti caratteri in un ordine *qualsiasi*.

- ! Indica che alcune delle restrizioni normalmente attive dovrebbero essere ignorate (se ne discuterà più avanti).
- h Indica che il float ha il permesso di essere messo intercalato con il testo (*here*, 'qui').
- t Indica che il float ha il permesso di andare nell'area superiore (*top*, 'in alto').
- b Indica che il float ha il permesso di andare nell'area inferiore (*bottom*, 'in basso').
- p Indica che il float ha il permesso di andare in una pagina *p* o nell'area occupata da una colonna (*page*, 'pagina').

L'ordine in cui si scrivono questi caratteri nell'argomento facoltativo *non* influenza il modo in cui l'algoritmo cerca di posare il float! L'esatta sequenza osservata dall'algoritmo verrà discussa nella sezione 3.2. Questo è uno degli equivoci più comuni che si verifica, per esempio, quando gli utenti pensano che [bt] chieda a L^AT_EX di provare per prima l'area inferiore.

Tuttavia, se una delle lettere non è presente, l'area corrispondente non verrà nemmeno provata.

2.4 Parametri dell'algoritmo

La posa di un float sulla pagina è influenzata da circa 20 parametri. Fondamentalmente, essi definiscono quanto segue.

1. Quanti float possono andare in una certa area.
2. D'ora in poi *pagina* o *colonna* *p*. (N.d.T.)
3. D'ora in poi *codice*. Nella sola sezione 5 *codice* è usato nell'accezione di *codice sorgente*. (N.d.T.)

2. Quanto grande un'area di float può divenire.
3. Quanto testo deve esserci in una pagina (cioè, quanta parte della pagina le aree di float superiore e inferiore possono occupare).
4. Quanto spazio verrà inserito:

- tra float consecutivi in una stessa area e
- tra l'area di float e il testo fra cui è intercalato.

2.5 Punto di riferimento di un float

Un luogo del documento che contiene il riferimento a un float (per esempio, «vedi la figura X») è detto *chiamata*. Il corpo del float dovrebbe essere messo vicino alla propria (prima) chiamata: infatti, la posizione dell'ambiente mobile nel documento sorgente influenza quella dell'oggetto nel documento finito, poiché determina il momento in cui L^AT_EX lo vede per la prima volta.

È importante capire che se un float viene messo in mezzo a un capoverso, per l'algoritmo il punto di riferimento diventa la successiva interruzione di riga o di pagina nel capoverso che segue l'effettiva sede del float nel sorgente.

Per ragioni tecniche e pratiche, di solito è meglio mettere tutti i float tra capoversi (per esempio, dopo il capoverso che ne contiene la chiamata), anche se la cosa allontana leggermente chiamata e punto di riferimento.

3 Comportamento fondamentale del meccanismo flottante

Sulla scorta di queste conoscenze, ora si può scavare a fondo nel comportamento dell'algoritmo.

Per prima cosa, bisogna capire che tutti gli algoritmi di composizione tipografica di L^AT_EX sono progettati per evitare qualsiasi tipo di ripensamento. Ciò significa che L^AT_EX legge da cima a fondo il documento sorgente, formatta ciò che vi trova e lo compone (più o meno) immediatamente. I programmatori scelsero questa condotta per limitare la complessità (che rimane ancora piuttosto elevata) e mantenere ragionevole la velocità di composizione (si ricordi che il programma risale ai primi anni '80 del Novecento quando anche i main frame erano lentissimi a confronto con i calcolatori moderni).

Per quanto riguarda i float, in altre parole l'algoritmo è 'avido': nel momento in cui incontra un ambiente mobile, L^AT_EX proverà immediatamente a posarlo e, se ci riesce, non ritornerà mai più su questa decisione. Dunque, questo significa anche che la soluzione adottata potrebbe rivelarsi infelice alla luce dei dati ricevuti successivamente.

Per esempio, se una figura ha il permesso di andare nell'area superiore o inferiore, L^AT_EX può decidere di metterla nella prima. Se a questa figura seguono due tabelle che possono andare solo in

cima alla pagina, queste tabelle potrebbero non starci più. Una soluzione probabilmente efficace in questo caso (ma che l'algoritmo non ha esaminato) sarebbe stata quella di mettere la figura in basso e le due tabelle in alto.

3.1 Sequenza di base

Ecco la sequenza di base eseguita dall'algoritmo.

1. Quando incontra un float, L^AT_EX prova a posarlo immediatamente secondo le proprie regole (descritte nel dettaglio più avanti).
2. Se ci riesce, lo posa e non cambierà più questa decisione.
3. Se non ci riesce, lo mette in una *coda di attesa*⁴ per reconsiderarlo quando comincerà a comporre la pagina successiva (ma non prima).
4. Una volta terminata una pagina, L^AT_EX esamina la coda di attesa e prova a svuotarla meglio che può, innanzitutto cercando di generare il maggior numero possibile di pagine *p* (nella speranza di svuotare la coda). Esaurita anche questa possibilità, tenterà di posare i float rimanenti in aree superiori e inferiori e, se non ci riesce, li differirà in una pagina successiva (cioè aggiungendoli ancora una volta alla coda di attesa).
5. Dopo di che, comincia a elaborare il materiale per questa nuova pagina. Durante il processo, può incontrare ulteriori float.
6. Se ha raggiunto la fine del documento o ha incontrato un comando `\clearpage`, L^AT_EX comincia una nuova pagina, allenta tutte le restrizioni gravanti sui float e svuota la coda sistemandone il contenuto in una o più pagine *p*.

Nella modalità a due colonne, il programma usa lo stesso algoritmo, con la differenza che lavora a livello di colonne: quando ha terminato una colonna, cioè, L^AT_EX esaminerà la coda e genererà colonne *p*, eccetera.

3.2 Regole di collocazione dettagliate

Ogni volta che L^AT_EX incontra un ambiente galleggiante nel sorgente, prima di tutto esaminerà la coda di attesa per controllare se ne contiene già un altro della stessa categoria. Se lo trova, il float appena incontrato non ha il permesso di essere posato e va immediatamente in coda.

In caso contrario, L^AT_EX ne esamina il codice, sia quello esplicito eventualmente dichiarato nell'argomento facoltativo, sia quello predefinito per la categoria di float in questione. Quest'ultimo è impostato nel file di classe (per esempio, `article.cls`) e molto spesso, ma non sempre, si risolve in `tbp`.

4. D'ora in poi anche semplicemente *coda*. (N.d.T.)

1. Se il codice contiene un `!`, l'algoritmo ignorerà qualunque restrizione connessa al numero di float che può andare in un'area o allo spazio massimo occupabile da quest'ultima sulla pagina. In caso contrario, le restrizioni imposte dai parametri restano attive.
2. Il passo successivo è controllare se tra i codici c'è `h`.
3. Se c'è, l'algoritmo tenterà di posare il float esattamente dove l'ha incontrato. Se c'è abbastanza spazio da permetterlo, l'elaborazione di quest'oggetto termina qui.
4. Se non c'è, subito dopo verificherà la presenza di `t`. Se lo trova, l'algoritmo proverà a posare il float nell'area superiore. In assenza di altre restrizioni che lo impediscano, il float viene messo lì e la sua elaborazione termina.
5. Se nemmeno `t` è tra i codici, L^AT_EX va alla ricerca di `b` e, se lo trova, proverà a posare il float nell'area inferiore (sempre obbedendo a tutte le restrizioni attive in assenza di `!`).
6. Se la cosa non funziona o non è permessa perché il codice non è stato indicato, il float viene aggiunto alla coda di attesa.
7. Durante il processo appena descritto, un codice `p` (se presente) non viene considerato. Verrà esaminato solo quando la coda viene svuotata al successivo confine di pagina o di colonna.

Questo termina il procedimento per posare un float quando L^AT_EX ne trova uno nel documento.

3.3 Svuotamento della coda al confine di pagina o di colonna

Dopo aver terminato una pagina o una colonna, L^AT_EX esamina la coda di attesa e cerca di svuotarla nel miglior modo possibile. Per farlo, prima di tutto proverà a creare pagine *p*.⁵

Ogni float che partecipa a una pagina (o colonna) *p* deve avere *p* tra i propri codici. Se non ce l'ha, non solo non può finire in una pagina di questo tipo, ma impedisce di finirci anche a ogni ulteriore float della stessa categoria che lo segue nella coda di attesa!

Se il float ha il permesso di andarci, verrà contrassegnato per esservi incluso, ma se la pagina non verrà riempita 'abbastanza' (secondo le impostazioni dei parametri che governano le pagine *p*) il programma può ancora interrompere il tentativo. Queste restrizioni vengono abolite solo alla fine del documento o quando è stato emesso un `\clearpage`: allora, un float verrà messo in una

5. Nella modalità a due colonne, L^AT_EX creerà colonne *p* (quando ha terminato una colonna) e cercherà anche di generare pagine *p* quando ha terminato una pagina. Nel resto dell'articolo, *pagina p* indicherà l'una o l'altra a seconda del contesto.

pagina **p** anche se non ha **p** tra i codici e risultasse essere l'unico di quella pagina.

La creazione di pagine **p** continua finché l'algoritmo non ha più float da posare o finché non fallisce nel produrle a causa delle impostazioni dei parametri. In questo secondo caso, tutti i float ancora in attesa di sistemazione vengono considerati per l'inclusione nelle aree superiore o inferiore della successiva pagina (o colonna).

Il processo è lo stesso di quello appena descritto, tranne che per i due punti seguenti.

1. Il codice **h** non ha più alcun significato (dato che ora si è molto lontani dall'originario *here*, 'qui') e perciò viene ignorato.
2. Questa volta i float non provengono dal documento sorgente, ma sono prelevati uno dopo l'altro dalla coda di attesa.

Tutti i float che non possono essere sistemati vengono rimessi in coda, in modo che quando L $\text{\TeX}$ è pronto per esaminare l'ulteriore input testuale dal sorgente, la coda può già contenerne. Una conseguenza di questo comportamento è che un float incontrato nel sorgente può essere immediatamente accodato solo perché un precedente float della stessa categoria è ancora in attesa di sistemazione.

3.4 Parametri che influenzano la posa

Quattro contatori controllano quanti float possono andare nelle varie aree di float.

totalnumber (valore predefinito: 3) Specifica il numero massimo di float permesso in una colonna di testo indipendentemente dalla loro posizione rispetto al testo. Non viene usato per le pagine **p**.

topnumber (valore predefinito: 2) Specifica il numero massimo di float permesso nell'area superiore della colonna di testo.

bottomnumber (valore predefinito: 1) Specifica il numero massimo di float permesso nell'area inferiore della colonna di testo.

dbltopnumber (valore predefinito: 2) Specifica il numero massimo di float a piena pagina permesso sopra le colonne di testo nella composizione a due colonne.

Le dimensioni delle aree sono controllate tramite parametri (modificabili con **\renewcommand**) che definiscono la massima (o minima) dimensione dell'area, espressa come una frazione dell'altezza della pagina. Di seguito se ne dà l'elenco.

\topfraction (valore predefinito: 0,7) Specifica l'altezza massima della frazione di pagina destinata ai float in testa a una pagina, se questa contiene anche del testo.

\bottomfraction (valore predefinito: 0,3) Specifica l'altezza massima della frazione di pagina destinata ai float in calce a una pagina, se questa contiene anche del testo.

\dbltopfraction (valore predefinito: 0,7) Specifica l'altezza massima della frazione di pagina a giustezza piena da destinare ai float a giustezza piena in testa alla pagina nella composizione a due colonne.

\textfraction (valore predefinito: 0,2) Specifica l'altezza minima della frazione di pagina destinata al testo in una pagina che contenga anche del testo (cioè, l'altezza minima dell'area che non deve essere occupata da float).

La costruzione di pagine o colonne **p**, invece, è controllata tramite i seguenti due parametri.

\floatpagefraction (valore predefinito: 0,5) Specifica la minima frazione della pagina (o della colonna) che deve essere occupata da float per fare di quella pagina (o di quella colonna) una pagina (o una colonna) **p**.

\dblfloatpagefraction (valore predefinito: 0,5) Specifica la minima frazione di una pagina **p** da occupare con float a giustezza piena nella composizione a due colonne. (*Aggiunta d.T.*)

Lo spazio che separa i float all'interno di un'area, come pure quello che c'è tra aree di float e aree riservate al testo, è definito dai seguenti parametri (che sono tutti lunghezze elastiche, cioè possono contenere alcuni componenti di allungamento o di accorciamento). I loro valori predefiniti dipendono dal corpo del font del documento e variano al variare delle relative opzioni di classe come **11pt** o **12pt**. Di seguito mostriamo solo i valori predefiniti per l'opzione **10pt**.

\floatsep (valore predefinito: **12pt** plus **2pt** minus **2pt**) Specifica la distanza minima tra due float consecutivi nell'area di float superiore o inferiore.

\dblfloatsep (valore predefinito: **12pt** plus **2pt** minus **2pt**) Specifica la distanza minima tra due float larghi l'intera pagina consecutivi in testa a una pagina composta a due colonne.

\textfloatsep (valore predefinito: **20pt** plus **2pt** minus **2pt**) Specifica la distanza minima da interporre tra l'area di float superiore o inferiore e quella riservata al testo.

\intextsep (valore predefinito: **12pt** plus **2pt** minus **2pt**) Specifica la distanza minima fra un float intercalato con il testo e il testo precedente e seguente. (*Aggiunta d.T.*)

\dbltextfloatsep (valore predefinito: **20pt** plus **2pt** minus **2pt**) È l'analogo di **\textfloatsep** per float a giustezza piena.

4 Conseguenze dell'algoritmo

4.1 Nel documento finito, un float può comparire prima di dove si trova nel sorgente

Nonostante che la posizione dell'ambiente mobile nel sorgente determini il punto a partire dal quale l'oggetto compare nel documento finale, il float può ancora indietreggiare di una certa misura, poiché gli è permesso andare nell'area superiore della pagina corrente (nella sezione 6.3 si spiega come cambiare questo comportamento). Tuttavia, non può finire in una pagina precedente il testo circostante, dato che L^AT_EX non riconsidera le pagine precedenti che sono già state composte.

È per questo motivo che di solito nel sorgente si mette un float vicino alla sua prima chiamata (cioè un testo come «si veda la figura 5»), poiché la cosa garantirà di vedere l'oggetto o nella stessa pagina della chiamata o in una pagina successiva. In alcune situazioni, tuttavia, si potrebbe voler mettere un float nella pagina precedente la chiamata (in modo che oggetto e chiamata si trovino in due pagine affiancate nel documento finale). Lo si può fare, ma solo spostando indietro a mano il float nel sorgente.

4.2 I float a giustezza piena vengono sempre differiti in prima istanza

Quando incontra un ambiente mobile largo tutta la pagina (indicato con un `*` alla fine del nome dell'ambiente, come per esempio `figure*`) nella composizione a due colonne, L^AT_EX sposta immediatamente quel float nella coda di attesa. La ragione di questo comportamento risiede ancora una volta nell'«avidità» dell'algoritmo: se L^AT_EX al momento sta assemblando la seconda colonna di quella pagina, significa che la prima colonna è già stata composta ed è conservata lontano. Si ricordi che non c'è modo di far stare il float nella pagina in questione, dato che L^AT_EX non ritorna sui propri passi. Per mantenere l'algoritmo semplice, il programma si comporta nello stesso modo anche se sta lavorando alla prima colonna (dove in teoria potrebbe fare di meglio anche senza ripensamenti).

Perciò, per mettere un float di questo tipo nella pagina corrente, bisogna spostarlo a mano nel sorgente prima dell'inizio della pagina. Così facendo, naturalmente, ogni successivo cambiamento nel testo potrebbe vanificare questo accorgimento. Di conseguenza, tali aggiustamenti funzionano meglio (se non benissimo) solo nell'ultimissima fase della produzione del documento – quando, cioè, tutto il materiale è stato scritto e l'attenzione è concentrata sulla messa a punto fine dell'aspetto del proprio lavoro.

Si noti anche che l'algoritmo di base presenta un bug⁶ in quest'area di float: esso mantiene due

code di stampa indipendenti, una per i float larghi una sola colonna e una per quelli larghi due. Di conseguenza, l'ordine con cui i float si susseguono nel sorgente non viene necessariamente rispettato nel documento finale: li si potrebbe vedere in una sequenza inaspettata. Se ciò dovesse accadere, bisogna spostare a mano l'oggetto a giustezza piena prima (o dopo) la sua sede originaria oppure caricare il pacchetto `fixltx2e`, che contiene una correzione per questo problema.

4.3 Non esiste un'area di float inferiore per i float a giustezza piena

Questa non è tanto una conseguenza dell'algoritmo quanto una caratteristica della sua implementazione. Le uniche posizioni in cui i float a giustezza piena possono essere posati sono l'area superiore o una pagina `p`. Come caso speciale di una certa rilevanza, si noti che il float contenuto in un ambiente del tipo `{figure*}[b]` verrà posato solo dopo un `\clearpage` o alla fine del documento.

4.4 Tutti i parametri (normalmente) restringono le possibilità di posa

Potrebbe essere ovvio, ma vale la pena di ripeterlo: ciascun parametro relativo ai float diminuisce le possibilità di L^AT_EX di posare gli oggetti sulle pagine. L'entità di questa restrizione dipende da come il parametro è stato impostato: c'è sempre un modo per farlo che non influenza affatto la posa dell'oggetto. Sfortunatamente, nell'agire in questo modo si incoraggiano collocazioni piuttosto brutte.

Le impostazioni predefinite di L^AT_EX sono abbastanza permissive. Per esempio, perché una pagina `p` venga accettata, il float (o i float) deve occupare almeno la metà. Il che significa, in altre parole, che una pagina di questo tipo può rimanere mezza vuota (e questa, nella maggior parte dei casi, non è certo la migliore disposizione possibile).

Spesso gli utenti cercano di migliorare queste impostazioni e rimangono sorpresi quando all'improvviso tutti i float si accumulano alla fine del documento. Per restare su quest'esempio: se si porta il parametro `\floatpagefraction` a 0,8, diciamo, un float che occupi circa lo 0,75 della pagina non potrà formare da solo una pagina `p`. Così, se non c'è un altro float da affiancargli e che ci stia effettivamente nello spazio rimanente, verrà differito e con lui tutti gli altri float della stessa categoria. Ma, peggio ancora, questo float in particolare è anche troppo grande per andare nella successiva area di float superiore, che può occupare al massimo lo 0,7 della pagina (valore predefinito di `\topfraction`). Di conseguenza, tutti i float rimangono differiti fino al successivo `\clearpage`.

Per questa ragione, durante la stesura del documento è meglio non toccare i parametri, o almeno non farlo in modo da rendere più difficile all'algoritmo la posa di un float vicino alla sua chiamata. Per la correzione delle bozze è molto più importante

6. Dato che questo è il comportamento documentato nel manuale di L^AT_EX, forse è più corretto chiamarlo *caratteristica indesiderata* anziché *bug*.

avere una figura vicino alla chiamata che evitare pagine mezze vuote. Le possibilità per mettere a punto finemente un documento il cui testo non necessita di modifiche vengono discusse più avanti.

Un'altra conclusione da tirare qui è che alcuni parametri dipendono da altri parametri: è importante tenere in considerazione queste dipendenze quando se ne modificano i valori.

4.5 ‘Qui’ significa solo ‘qui se ci sta’

... e spesso non ci sta. La cosa è piuttosto sorprendente per molte persone, ma per il modo in cui l'algoritmo è stato progettato, il codice `h` (specificato da solo) non è un comando incondizionato. Se servisse un comando di questo tipo, si può ricorrere a pacchetti di estensione come `float`, che offre il codice alternativo `H`, che significa davvero ‘qui’ (e, se necessario, comincia una nuova pagina).

4.6 I codici non definiscono un ordine di preferenza

Come s'è detto sopra, l'algoritmo prova a posare i float nelle aree di float disponibili in un ordine ben preciso nativo dell'algoritmo: ‘qui’, ‘in alto’, ‘in basso’ e – ai confini di pagina – prima ‘in una pagina p’ e, solo se la cosa non è più possibile, ‘in alto’ seguito da ‘in basso’ per la pagina successiva.

Così, specificare `[bt]` non significa ‘prova a posare questo float prima in basso e solo dopo in alto’, ma semplicemente ‘permetti che questo float finisca nell'area superiore o inferiore (ma non in una pagina p)’ esattamente come farebbe `[tb]`.

4.7 Relazione tra float e note al piede

Questa non è esattamente una conseguenza dell'algoritmo ma una delle sue implementazioni: ogni volta che $\LaTeX$ prova a decidere la posizione di un float (o di una nota al margine con `\marginpar`!) deve attivare la routine d'uscita per farlo. E come parte di questa routine, tutte le note al piede presenti sulla pagina vengono rimosse dalla loro posizione corrente nella bozza e riunite nella scatola `\footins`. Si tratta di un passaggio obbligato con cui $\TeX$ si prepara a produrre la pagina.

Ma dopo aver posato il float (o dopo averlo differito), $\LaTeX$ restituisce il materiale della pagina alla bozza, che ora è cambiata: le note, infatti, sono state rimosse dalle loro posizioni iniziali. $\LaTeX$, così, deve spostare le note, ma non può fare altro che metterle in uno stesso posto (non conoscendone più le sedi originali). Ciò che fa è reinserire le note (il loro testo, per la precisione) alla fine della bozza. Esistono alcune buone ragioni per fare così, una delle quali è che $\LaTeX$ si aspetta che tutto il materiale restituitogli riesca a stare ancora nella pagina corrente.

Tuttavia, se per qualche ragione alla fine il programma rileva un'interruzione di pagina in un punto precedente, le note al piede compariranno nella pagina o nella colonna sbagliata. Si tratta di uno

scenario abbastanza improbabile e $\LaTeX$ lavora duramente per renderlo quasi impossibile, ma se succedesse, si controlli se c'è un float vicino all'interruzione di pagina scelta e o lo si sposti o si guidi l'algoritmo usando interruzioni di pagina esplicite. Si può trovare un esempio di questo comportamento in un'altra domanda posta su $\TeX$ - $\LaTeX$ Stack Exchange (HERMANN, 2012). Vale la pena di sottolineare il caso particolare discusso nella domanda appena segnalata: *non* si metta un float immediatamente dopo un titolo, a meno che non sia un titolo che comincia sempre una pagina. La ragione è che di solito i titoli formano oggetti molto grandi (dato che impediscono un'interruzione di pagina immediatamente dopo di sé). Tuttavia, mettere un float in questa posizione comporta l'attivazione della routine di uscita prima che $\LaTeX$ abbia deciso dove interrompere la pagina, con il conseguente spostamento di eventuali note al piede nel posto sbagliato.

5 Documentazione dell'algoritmo

Come richiesto, ecco alcune informazioni sulla documentazione esistente. Algoritmo e relativa implementazione sono documentati nel file `l1output.dtx` come parte del sorgente del kernel di $\LaTeX$. Questo file può essere composto da solo o come parte dell'intero kernel (cioè componendo il file `source2e.tex` – si ignori il checksum error se è ancora lì, mi dispiace⁷).

Questa documentazione è un interessante manufatto storico. Parte di essa mostra pseudocodice semiformattato risalente a $\LaTeX$ 2.09; in altre parole, proviene dalla documentazione originale di Leslie Lamport. Il codice vero e proprio è documentato usando lo stile doc e in parte è più o meno documentato (da zero) e risale al 1994 o giù di lì, quando Chris Rowley e io abbiamo adeguato ed esteso l'algoritmo originale per $\LaTeX$ 2 ϵ (la versione corrente). Inoltre, documenta abbastanza apertamente i vari problemi con l'algoritmo e/o la sua implementazione – in molti casi non abbiamo avuto il coraggio di cambiarla per via delle numerose dipendenze e, naturalmente, per il pericolo di danneggiare troppi documenti esistenti che nel bene e nel male si affidavano implicitamente al comportamento corrente.⁸ Verso la fine del codice si troverà un elenco di commenti sull'algoritmo compilati allora, ma ci sono anche commenti, domande e cose rimaste da fare (?-) disseminati lungo tutta la documentazione del codice.

Un aspetto interessante di questo file (che ho dimenticato del tutto) è che esso contiene tutto il codice necessario per tracciare il comportamen-

7. Ma ciò significa anche che si sta facendo girare una precedente versione di $\LaTeX$.

8. Questa è la ragione, per esempio, per cui la correzione del problema discusso nella sezione 4.2 è stata inserita nel pacchetto `fixltx2e` e non fa parte dell'algoritmo del kernel.

to dell'algoritmo nella vita reale. È un prodotto piuttosto grezzo e dettagliato e probabilmente per questa ragione non l'ho reso pubblicamente disponibile allora. Ma anche nella sua forma attuale permette di gettare qualche sguardo sul comportamento dell'algoritmo e su come avvengono certe decisioni.

Così, mentre scrivevo questo articolo ho avuto dei ripensamenti e ora la più recente distribuzione di L^AT_EX (maggio 2014)⁹ offre il pacchetto `fltrace` che si può caricare per tracciare alcune strane decisioni sulla posa dei float o semplicemente per capire un po' meglio l'algoritmo. Esso offre i comandi `\tracefloats` e `\tracefloatsoff` per avviare o interrompere la tracciatura dell'algoritmo e `\tracefloatvals` per mostrare i valori correnti dei vari parametri discussi in questo articolo.

Poiché il pacchetto è identico al codice del kernel tranne che per l'aggiunta della tracciatura, potrebbe non funzionare correttamente caricando qualunque altro pacchetto che manipola quella stessa parte del codice del kernel. Se ciò dovesse accadere, la soluzione migliore è caricare `fltrace` per primo.

6 Affrontare problemi specifici

In questa sezione finale discuteremo alcune strategie per aggirare o risolvere problemi comuni. Non è in nessun modo completa e ulteriori informazioni si possono trovare in altre pubblicazioni, per esempio (GOOSSENS *et al.*, 2004) che dedica ai float un intero capitolo.

6.1 Assicurarsi che i float compaiano 'qui'

Talvolta bisogna garantire che i float compaiano in linea in determinati punti del documento, anche se ciò si traduce in pagine parzialmente vuote. Come s'è discusso sopra, il codice `h` non fornisce questa funzionalità, presente, però, in certi pacchetti di estensione: `float`, per esempio, definisce a questo scopo un codice `H`.

Una possibile alternativa è costituita dal comando `\captionof` definito dal pacchetto `caption`, che genera una normale didascalia (comprensiva della relativa voce nella lista delle figure o delle tabelle, eccetera) ma senza includere per forza l'oggetto in un float.

6.2 Area di float inferiore per float a giustezza piena

Come s'è visto nelle sezioni precedenti, l'algoritmo standard non supporta float a giustezza piena nel-

9. Se si dispone di una precedente versione di L^AT_EX, si può ancora estrarre il codice da sé, scrivendo un breve file di installazione `fltrace.ins` con il seguente contenuto:

```
\input docstrip
\generateFile{fltrace.sty}{t}{%
  \from{ltoutput.dtx}{fltrace,trace}}
\endbatchfile
```

ed elaborandolo con L^AT_EX.

la parte inferiore di una pagina composta a due colonne. Caricando il pacchetto `stfloats`, questa funzionalità mancante viene aggiunta, tranne che per la prima pagina.¹⁰

6.3 Assicurarsi che i float siano posati sempre dopo la loro chiamata

Per impostazione predefinita, l'algoritmo permette a un float di retrocedere prima della sua chiamata fin tanto che entrambi si trovano sulla stessa pagina. Più precisamente, gli permette di comparire nell'area superiore della colonna in cui è stato incontrato.

Questa prassi offre al float migliori possibilità di essere visibile dal punto in cui si trova la chiamata e di non finire in una pagina successiva. Tuttavia, le redazioni di alcune riviste considerano questo comportamento troppo permissivo e richiedono che gli oggetti siano messi tassativamente dopo la loro chiamata. Dal punto di vista dell'algoritmo, in altre parole, solo l'area inferiore della colonna in cui c'è la chiamata costituisce un'opzione di posa valida. Soddisfa tale requisito il pacchetto `flafter`, che implementa questa strategia.

La cosa può funzionare bene se il documento contiene solo pochi float. Per documenti che ne contengono moltissimi, ovviamente, la questione diventa molto più difficile, tanto che si potrebbero trovare gli oggetti tutti insieme alla fine del documento o del capitolo, o si potrebbe ricevere l'errore `Too many unprocessed floats`.

6.4 Evitare float in certe pagine

Talvolta è utile impedire che i float compaiano in una determinata pagina, come per esempio impedire che un float presente in una nuova sezione finisca nell'area superiore della pagina corrente ma consentendo un piazzamento nell'area superiore di una pagina successiva. Per questo tipo di messa a punto fine, L^AT_EX offre il comando `\suppressfloats[⟨collocazione⟩]`, dove `⟨collocazione⟩` può essere `t` o `b` e impedisce ogni ulteriore posa nelle rispettive aree della pagina corrente. Se non lo si dà, tutti i float rimanenti sulla pagina corrente verranno differiti.

6.5 Implementare barriere per i float

L^AT_EX standard implementa già una barriera per i float chiamata `\clearpage`. Se nel testo sorgente un float si trova prima della barriera, nel documento finito non potrà mai comparire dopo, e viceversa. La cosa funziona così: il programma emette tutti i float differiti estraendoli dalla coda di attesa, genera pagine `p` se necessario e infine comincia una nuova pagina. Mentre questo meccanismo è adatto a mantenere gli oggetti all'interno di un capitolo (dato che i capitoli cominciano di solito su una pagina nuova) esistono situazioni in cui si

10. Si veda (BEETON, 2014) per eliminare a mano anche questa restrizione.

potrebbe desiderare una barriera meno rigida, cioè che funzioni senza forzare una nuova pagina o che sia parzialmente permeabile.

Questa funzionalità è offerta dal pacchetto `placeins`, il cui comando `\FloatBarrier` non introduce un'interruzione di pagina. Tramite le opzioni del pacchetto si può modificare il comportamento della barriera per permettere ai float di migrare prima o dopo di essa, ma fin tanto che compaiono ancora sulla stessa pagina.

6.6 Sovrascrivere restrizioni di posa

Se un determinato float è (leggermente) troppo grande per stare in una certa area di float o se un'area contiene già il massimo numero di float permesso ma si vuole forzare comunque un ulteriore oggetto in questa posizione, allora è una buona scelta aggiungere ! all'argomento facoltativo. Questo carattere ordina all'algoritmo di ignorare tutte le eventuali restrizioni gravanti sul float in questione attuate mediante parametri, in modo che l'oggetto verrà posato sempre, purché non ci siano già altri float differiti della stessa categoria o le aree permesse non diventino più ampie dello spazio disponibile quando si aggiunge l'oggetto.

Siccome l'ordine dei tentativi è sempre lo stesso (prima in alto, poi in basso), si può usare `!b` per forzare un float nell'area inferiore poiché `!tb` riuscirebbe già normalmente a posarlo in quella superiore. Ovviamente, il rovescio della medaglia è che se il float non ci sta, comparirà solo nell'area inferiore di una pagina successiva. Perciò, qualsiasi modifica del testo operata in seguito potrebbe creare il caos nelle decisioni di posa dell'utente.

6.7 Un consiglio finale di messa a punto

Esistono molti modi per regolare il comportamento dell'algoritmo e la maggior parte di essi è stata discussa in questo articolo. Tuttavia, c'è ancora

una 'messa a punto' possibile, la più potente di tutte: le modifiche al testo del documento.

Pertanto, come ultimo consiglio: non cominciate a manipolare i parametri o a cambiare i codici o a spostare i float nel documento sorgente finché il testo non è definitivamente concluso e il documento è prossimo al termine. Queste manovre sono uno spreco di energie e possono produrre facilmente pose scadenti poiché, le restrizioni imposte all'inizio potrebbero non risultare più adeguate dopo una (anche minima) modifica del testo.

Riferimenti bibliografici

BEETON, B. (2014). «Placing a full-width insert at the bottom of two columns». *TUGboat*, **35** (3), p. 255. URL <https://www.tug.org/TUGboat/tb35-3/tb111beet-banner.pdf>.

DANIEL, M. (2012). *How to influence the position of float environments like figure and table in L $\text{\TeX}$?* URL <http://tex.stackexchange.com/questions/39017/>.

GOOSSENS, M., MITTELBACH, F., BRAAMS, J., CARLISLE, D. e ROWLEY, C. (2004). *The L $\text{\TeX}$ Companion. Tools and Techniques for Computer Typesetting*. Addison-Wesley, Reading, MA, USA, 2^a edizione.

HERMANN, M. (2012). *“Thanks” note (footnote) placed below right column even though there is enough space on the left*. URL <http://tex.stackexchange.com/questions/43294/>.

LAMPORT, L. (1994). *L $\text{\TeX}$. A Document Preparation System. User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 2^a edizione. Ristampato con correzioni nel 1996.

▷ Frank Mittelbach
L $\text{\TeX}$ 3 Project
<http://www.latex-project.org/>

Una figura larga sotto un testo a due colonne*

Barbara Beeton

Componendo a due colonne c'è una solo posto nella pagina dove in condizioni normali non si può mettere una figura a giustezza piena con l'istruzione `\begin{figure*}`: il fondo della pagina. Il pacchetto `stfloats` toglie questa limitazione salvo nella prima pagina dove compare il titolo del documento. Lo scopo di questo esercizio è di dimostrare che invece questo può essere fatto usando solo i normali strumenti di $\text{\LaTeX}$.

Perché una persona vorrebbe (o dovrebbe) fare una cosa simile? Si consideri un progetto per il quale un rapporto intermedio sia rappresentato meglio mediante una tabella o un diagramma, con pochissimo testo, ma che sia specificato un formato a due colonne. L'effetto espressivo dei dati viene perso se l'illustrazione deve essere posposta alla pagina successiva lasciando la prima pagina quasi vuota. Di fatto lo scopo del rapporto potrebbe perdersi se il destinatario impaziente non volta la pagina.

Nell'incontro annuale di TUG 2010, Frank Mittelbach presentò un lavoro intitolato “Exhuming coffins from the last century” che trattava i problemi relativi alla collocazione delle scatole sulla pagina. Il lavoro non poté essere stampato su *TUGboat*, ma Kaveh Bazargan era presente con la sua attrezzatura di registrazione e produsse un video che può essere visto sul sito river-valley.zeeba.tv/exhuming-coffins-from-the-last-century/. Le tecniche proposte in quel contributo non possono risolvere questi problemi in tempi brevi, ma sono promettenti per il futuro.

Nell'incontro TUG 2014 di Portland, Boris Veytsman presentò un lavoro¹ sulla composizione

*Questo breve articolo di una sola pagina è stato pubblicato su *TUGboat*, vol. 35 (2014) n. 3. È stato tradotto da Claudio Beccari e Tommaso Gordini col permesso dell'autrice. Eventuali errori o fraintendimenti dell'articolo originale sono da attribuirsi ai traduttori.

1. An output routine for an illustrated book: Making the *FAO Statistical Yearbook*, *TUGboat* **35**:2, pagine 202–204.

di un libro dove le figure erano molto più importanti — e occupavano più spazio — del testo e c'erano pagine con due colonne di testo in alto nella pagina e una singola figura larga in basso. Tuttavia la natura del materiale consentiva di trattare tutte le pagine in quattro quadranti che potevano essere gestiti individualmente o come coppie orizzontali o verticali.

Che cosa si può fare oggi? Nella figura 1² viene mostrato un testo $\text{\LaTeX}$ che produce l'inserimento di un oggetto a giustezza piena sotto ad un testo composto a due colonne. Naturalmente questo metodo funziona anche per un testo più lungo di una pagina, ma per questa dimostrazione una pagina di *TUGboat* è sufficiente.

La figura larga è inserita nella prima colonna con collocazione [b]. La pagina deve contenere abbastanza testo per continuare nella seconda colonna. Il trucco consiste nello specificare un comando `\enlargethispage` con un argomento negativo che lascerà parte della seconda colonna vuota, cosicché la figura larga può uscire dal suo margine nello spazio rimasto vuoto. (Componendo a due colonne `\enlargethispage` equivale al comando (inesistente) `\enlargethiscolumn`.)

Questo metodo è manuale e richiede iterativamente l'intervento dell'utente da eseguirsi quando il testo è completamente a posto. È possibile modificare alcuni parametri di $\text{\LaTeX}$, come per esempio `\bottomfraction` o lavorare sulle didascalie. Tuttavia in certe situazioni consente di ottenere effetti che non si possono ottenere per altra via.

▷ Barbara Beeton
<http://tug.org/TUGboat>
tugboat (at) tug dot org

2. Questo metodo è stato presentato su sito $\text{\TeX}$ – $\text{\LaTeX}$ Stack Exchange in [TeX.stackexchange.com/q/107270](https://tex.stackexchange.com/q/107270).

```
\begin{figure}[b]
\setlength{\hfuzz}{1.1\columnwidth}
\begin{minipage}{\textwidth}
... codice per inserire la figura
\caption{Una figura a giustezza piena in fondo alla prima pagina!}
\end{minipage}
\end{figure}
... Che cosa si può fare oggi?\enlargethispage{-11.46\baselineskip}
```

FIGURA 1: Una figura a giustezza piena in fondo alla prima pagina!

\vadjust, questo sconosciuto

Claudio Beccari

Sommario

Il comando `\vadjust` è virtualmente sconosciuto alla maggior parte degli utenti di $\text{\LaTeX}$. Serve per eseguire cose un po' insolite, ma semplici e con poco sforzo. In questo articolo verranno mostrate alcune applicazioni.

Abstract

Command `\vadjust` is virtually unknown to the majority of $\text{\LaTeX}$ users. This primitive command allows to do some actions that are unusual, but at the same time rather simple and with little effort. This article describes some applications.

1 Introduzione

Il comando `\vadjust` è nativo di $\text{\TeX}$ -the-program; quindi è disponibile in tutti i motori di composizione del sistema $\text{\TeX}$, dato che devono essere compatibili con il capostipite ideato da Donald E. Knuth.

Esso entra nelle definizioni di molti comandi sia di Plain $\text{\TeX}$, sia di $\text{\LaTeX}$; ma gli utenti di questi mark-up difficilmente ne conoscono l'esistenza. $\text{\LaTeX}$ è stato creato con lo scopo di funzionare da interfaccia fra l'utente e l'interprete primitivo del mark-up, pertanto separa l'utente dalla difficoltà di usare un linguaggio di programmazione ricco di comandi (circa 300 nativi di $\text{\TeX}$) non molto semplici da usare. Persino Knuth, nello scrivere i suoi libri, ha sviluppato pacchetti di macro proprio per rendere più sicure molte azioni ripetitive. Gli utenti di $\text{\LaTeX}$ raramente hanno aperto il file `latex.ltx` per vedere che cosa contiene; se lo facessero, rimarrebbero impressionati dal numero di macro che vi vengono definite e dalla moltitudine di comandi nativi che vi vengono usati.

La disponibilità di molti *moduli* di espansione¹ rende ancora più potente l'uso di $\text{\LaTeX}$, ma difficilmente l'utente si rende conto di quello che c'è dietro questa architettura modulare. È giusto che sia così, ma questo non toglie che qualche volta anche un normale utente non possa desiderare di crearsi qualche semplice macro per svolgere compiti che forse sono già svolti da moduli esistenti (quasi 3000), eppure difficili da trovare anche qua-

lora si disponesse di una installazione completa del sistema $\text{\TeX}$.

Ecco allora che sapere qualcosa sui comandi nativi di $\text{\TeX}$ può essere utile a chiunque, specialmente per fare cose semplici anche se insolite.

2 Cosa fa \vadjust

Il comando `\vadjust` richiede un argomento che deve essere costituito da comandi che si possono eseguire in modo verticale. Questi comandi verranno eseguiti appena l'impaginatore, che fa parte del motore di composizione, passa in modo verticale. Il *modo verticale*, in parole povere, è quella modalità di funzionamento del motore di composizione nella quale esso dispone degli oggetti uno sopra l'altro, tipicamente quando il testo di un capoverso viene spezzato in righe che vengono collocate l'una sull'altra a formare una lista verticale. Il motore di composizione si trova in modo verticale anche fra un capoverso e l'altro, fra un capoverso e la matematica in display, eccetera. In questa modalità il motore di composizione può inserire fra un oggetto e l'altro degli spazi (verticali), delle penalità e altri oggetti come il contenuto dell'argomento di `\vadjust`. Però il comando `\vadjust` deve essere usato solo in modo orizzontale, cioè quando il motore di composizione sta raccogliendo il testo prima di scomporlo in una lista di righe giustapposte verticalmente. Per questo genere di dettagli si può andare a fondo consultando sia il testo di [KNUTH \(1984\)](#), sia quello di [ELJKHOUT \(2014\)](#).

La particolarità dell'argomento di `\vadjust` è che esso viene sviluppato ed eseguito solo quando viene trasferito nella lista verticale, non quando il motore di composizione trova il comando `\vadjust` che lo contiene; quindi l'esecuzione di quell'argomento viene differita al momento in cui esso viene trasferito alla lista verticale.

Il discorso potrebbe apparire astratto, ma gli esempi che seguono ne mostrano il funzionamento e i capoversi precedenti diventano più chiari.

3 Spezzare una colonna di testo

Quanto viene descritto qui di seguito vale anche quando si sta componendo in modalità `onecolumn`, ovvero la modalità di composizione predefinita di quasi tutte le classi, o quando si sia specificato il comando `\onecolumn` in un testo altrimenti composto a due colonne. Ma trovo che quanto descritto qui di seguito sia più interessante quando si compone a due colonne senza ricorrere al pacchetto

1. Generalmente chiamati *pacchetti*, perché solitamente non sono composti da una solo file con estensione `.sty`, ma i vari file componenti sono impacchettati in un unico archivio compresso, la cui installazione nelle distribuzioni del sistema $\text{\TeX}$ viene gestita da appositi programmi.

multicol, che certamente è molto utile, ma ha anche alcune sue limitazioni; in particolare non consente di inserire oggetti flottanti in colonna, ma solo fuori delle colonne e a piena pagina. Tuttavia, uno dei suoi pregi è quello di consentire il bilanciamento delle colonne nell'ultima parte composta dentro il suo ambiente *multicols*.

Componendo invece a due colonne, o perché si è specificata l'opzione *twocolumn* alla classe, o perché si è dato il comando `\twocolumn` esplicitamente o implicitamente attraverso altri comandi o ambienti², le colonne non vengono pareggiate nell'ultima pagina. Esiste il modulo *balance*, che però modifica la routine di output di LATEX e per questo può dare problemi non indifferenti, specialmente se le colonne da bilanciare contengono note e oggetti flottanti. Si può ottenere lo stesso risultato in modo meno automatico, ma molto più semplice, usando `\adjust`.

Infatti basta definire il comando `\pareggia` in questo modo:

```
\newcommand*\pareggia{\adjust{\newpage}}
```

e intercalarlo fra due parole della riga dopo la quale si vuole spezzare la prima colonna. Sembra troppo semplice e lo è, ma non è automatico.

Bisogna procedere così: si compone il testo senza usare quel comando; si vede quanto sia più corta la seconda colonna dell'ultima pagina rispetto alla prima colonna; si contano le righe di differenza e si divide il loro numero per due; si cerca la riga che sia quel numero di righe prima del fondo della prima colonna; si inserisce quel comando e probabilmente il risultato che si ottiene compilando di nuovo è proprio quello che si desiderava; può succedere di avere mal contato le righe o che il comando vada inserito nella riga precedente, ma in sostanza il gioco è fatto.

Nel comporre a due colonne, la routine di uscita compone ciascuna colonna come se fosse una pagina, solo che nella pagina fisica dispone le due pagine/colonne una accanto all'altra; una volta emessa la colonna di sinistra, si dimentica di quello che conteneva e procede ad elaborare la colonna di destra.

Ecco allora che mentre predispone la colonna di sinistra, ne incolonna le scatole che contengono ogni riga, e quando arriva alla riga che contiene il comando `\adjust` ne estrae il contenuto per inserirlo subito dopo la riga che lo conteneva; nel nostro caso la definizione di `\newpage` è la seguente:

```
\def \newpage {%
  \if@noskipsec
    \ifx \@nodocument\relax
      \leavevmode
    \global \@noskipsecfalse
```

2. Per esempio, l'indice analitico viene composto a due colonne anche nei documenti composti ad una colonna per impostazione predefinita.

```
\fi
\fi
\if@inlabel
  \leavevmode
\global \@inlabelfalse
\fi
\if@nobreak\@nobreakfalse\everypar{}\fi
\par
\vfil
\penalty -\M}
```

Di questa definizione ci interessano le ultime righe: `\par` termina il capoverso, nel caso si stesse lavorando in modo orizzontale, e questo è il motivo per il quale non si può usare `\newpage` direttamente; ma quando `\newpage` viene estratto siamo già in modo verticale, quindi il comando `\par` è del tutto inerte (non operativo); `\vfil` riempie il fondo della colonna di spazio verticale infinitamente allungabile, così da evitare l'allargamento degli spazi fra ogni capoverso insieme ai messaggi di colonna/pagina (`\vbox`) mal riempita; infine inserisce una penalità negativa che induce la routine di uscita a spezzare la colonna proprio in quel punto. La routine di uscita si dimentica della colonna appena composta e comincia a elaborare la seconda colonna, aggiungendovi subito le righe del capoverso appena interrotto e non ancora emesse, per accodarvi poi le righe dei capoversi successivi. Se la posizione del comando `\adjust` con il suo argomento è stata scelta in modo corretto, le due colonne della pagina appariranno bilanciate, o al massimo la seconda colonna sarà una riga più corta della colonna di sinistra.

Come ho spiegato sopra, le righe importanti sono le ultime due; tanto varrebbe allora che l'utente definisse semplicemente

```
\newcommand*\pareggia
{\adjust{\vfill\penalty -10000}}
```

e si sarebbe ottenuto lo stesso risultato, visto che nel caso specifico `\adjust` manda in esecuzione i comandi del suo contenuto solo quando si è già in modo verticale.

4 Come marcare la riga che contiene un voce riportata nell'indice analitico

È esperienza comune che se si consulta un indice analitico di un testo stampato, si trova facilmente la pagina o l'elenco delle pagine del testo dove la voce indicizzata è usata in modo non casuale. Poi si va alla pagina indicata o, in successione, alle pagine indicate, ma spesso bisogna leggere l'intera pagina prima di trovare la parola che interessa. Se la voce che è stata inviata all'indice non è evidenziata, in qualche modo la lettura dell'intera pagina fa perdere non poco tempo. Ma se la riga che contiene almeno una parola indicizzata è marcata a margine

con un segno qualsiasi, l'operazione diventa più semplice; anche se quella pagina contenesse diverse parole indicizzate, esse sarebbero certamente in numero molto minore delle righe presenti, quindi basta leggere solo le poche righe che hanno il marchio nel margine destro.

Si potrebbe usare una nota marginale; peccato che se le note marginali sono molte e il testo contiene un buon numero di oggetti flottanti, la presenza di molte note marginali può far perdere l'inserimento di alcuni essi.

Mi è capitato di vedere un testo con moltissime figure raccolte in una apposita appendice in fondo al documento e qualche figura mantenuta nel testo; ognuna di queste figure aveva la sua didascalia numerata, ma le figure in appendice venivano segnalate con il loro numero nel margine esterno collocato col meccanismo delle note marginali; le figure nel capitolo erano in numero decisamente limitato e non c'era, quindi, nessun motivo per perderne per la strada³. Ma collocare nel file di uscita degli oggetti flottanti implica fare i conti anche con le note marginali, anch'esse mobili. Di conseguenza la persona che stava componendo quel testo non riusciva ad uscirne; le ho cambiato tutte le note marginali con un comando che faceva uso di `\vadjust` e il problema è sparito.

In entrambi i casi conviene definire una macro. Nel caso delle voci da indicizzare, basta definire un nuovo comando, diciamo `\Index` in questa maniera:

```
\newcommand\Index[1]{#1\index{#1}%
\vadjust{\hbox to \linewidth{\hfill
\rlap{\enspace\smash{\raisebox
{0.3ex}{\rule{1ex}{1ex}}}}}}}
```

e ogni riga che contiene una voce indicizzata appare marcata con un rettangolino nero, come accanto a questa riga. Qualche commento non è fuori luogo. ■

1. `\hbox to\linewidth{\hfill...}` è del tutto equivalente a `\makebox[\linewidth][r]{...}` tranne per il fatto che il comando `\hbox` non modifica il modo di composizione, mentre `\makebox` passa al modo orizzontale.
2. Si è usato `\linewidth` che rappresenta sempre la lunghezza corrente della riga di testo, sia essa ad una colonna o a due colonne, sia essa con margini rientrati o non rientrati rispetto ai margini normali; in talune circostanze potrebbe non essere la scelta giusta, ma va bene quasi sempre.
3. `\rlap{...}` è quasi equivalente al comando $\LaTeX$ `\makebox[0pt][l]{...}`; anche `\rlap` non passa in modo testo, ma qui non è importante perché è contenuto dentro una `\hbox`; lo

3. Le code degli oggetti flottanti arrivano a contenerne al massimo 18; se si usa il pacchetto *morefloats* si può arrivare fino a 36 oggetti in coda.

si è usato semplicemente perché è più corto da scrivere rispetto al corrispondente comando $\LaTeX$.

4. Il rettangolino nero composto con il comando $\LaTeX$ `\rule` ha il suo punto di riferimento alla base della scatola che lo contiene, quindi risulterebbe più in basso del desiderato. Ecco perché si usa `\raisebox`: l'ammontare del rialzo è confrontabile con la profondità dei discendenti della riga dopo la quale si trova il contenuto di `\vadjust`; si potrebbe rendere più precisa questa operazione usando `\prevdepth`, che rappresenta la profondità della riga precedente (KNUTH, 1984; EIJKHOUT, 2014).
5. Non a caso si è usata l'unità di misura `ex`, cosicché tutta la costruzione può funzionare correttamente accanto a ogni riga composta con font di qualunque corpo.
6. Un altro comando usato è `\smash`. Esso è definito anche quando si usa $\LaTeX$, ma è poco conosciuto: serve per azzerare l'altezza e la profondità della scatola contenente l'argomento di `\smash`. In questo modo l'altezza e la profondità della prima scatola `\hbox` risultano entrambe nulle e il contenuto di `\vadjust` non altera la spaziatura fra le righe del testo circostante.
7. Il secondo problema, quello di mettere i riferimenti delle figure in appendice nel margine destro, si risolve nello stesso modo sostituendo `\rule` e i suoi argomenti con `\ref` e l'etichetta della figura a cui si vuol fare riferimento.
8. Componendo a due colonne sarebbe desiderabile che `\vadjust` avesse argomenti diversi per la colonna di sinistra e quella di destra; si può fare, naturalmente, ma lo si lascia per esercizio al lettore.

5 L'equivalente di `\showidx`

Il modulo *showidx* serve durante la lavorazione di un documento per marcare nel margine il contenuto del comando `\index`; alcuni autori lo considerano molto comodo. Per ottenere questo risultato il modulo ridefinisce alcuni comandi interni di $\LaTeX$ al fine di usare una nuova categoria di "insert", cioè di oggetti da inserire nel documento finale in modo flessibile, come di fatto viene fatto anche per le note al piede e le note marginali.

Se si vuole evitare di caricare pacchetti che modifichino il cuore di $\LaTeX$ si può ricorrere a `\vadjust`. Definendo un nuovo comando `\indexshow` si può ottenere il medesimo risultato che si ottiene con il modulo *showidx* senza compromettere il cuore di $\LaTeX$.

Il comando potrebbe essere del tipo:

```
\newcommand*\indexshow[1]{\index{#1}%
  \vadjust{\hbox to \linewidth{\hfill
    {\rlap{\enspace
      \smash{\raisebox{0.3ex}{#1}}}}}}}
```

Anche in questo caso, componendo a due colonne si potrebbe mettere il contenuto di `\indexshow` a sinistra della prima colonna e a destra della seconda colonna, ma il trucco è sempre lo stesso che ho lasciato prima al lettore.

È bene ricordare che questo comando potrebbe essere migliorato se l'argomento di `\smash` fosse composto dentro una scatola verticale, cioè, in termini del mark-up di LATEX, mediante una `\parbox`, perché talvolta le stringhe passate come argomento a `\index` possono essere un po' lunghe; così facendo, potrebbero ripiegarsi nella giustezza del margine.

6 `\vadjust` e `\enlargethispage`

Componendo a due colonne può succedere che la prima colonna dell'ultima pagina contenga una nota che va al piede della colonna, anche se si usa il comando `\pareggia` definito sopra.

Questo caso ha una soluzione semplice se si usa in tandem il comando `\pareggia`, definito sopra, con `\vadjust` e `\enlargethispage`⁴, oltre ai comandi `\footnotemark` e `\footnotetext`. Questi ultimi comandi servono per gestire il richiamo della nota separatamente dal testo, cosicché sia possibile mettere il testo della nota in posti diversi da dove li metterebbe il semplice comando `\footnote`.

In sostanza bisogna usare `\footnotemark` nell'esatto posto in cui si vuole mettere il richiamo delle note e inserire il comando `\footnotetext` con il testo della nota alla fine del testo della seconda colonna, cosicché la nota vada a finire al piede della seconda colonna.

Ciò fatto, si compone il documento e poi si contano quante righe della prima colonna non sono affiancate da righe della seconda colonna; si divide per due questo numero e, a partire dalla riga precedente il piede della prima colonna, si inserisce il comando `\pareggia`. Compilando nuovamente il

documento si vede la prima colonna accorciata, ma la nota resta sempre al piede della seconda colonna. Alla fine del testo sorgente della seconda colonna si inserisce poi il codice per diminuirne l'altezza:

```
\enlargethispage{-<n>\baselineskip}
```

dove $\langle n \rangle$ è pressappoco il numero di righe di cui si è accorciata la prima colonna.

Questo valore $\langle n \rangle$ va regolato finemente con qualche compilazione successiva, ma non è difficile trovare qualche decimale che pareggi esattamente le due colonne. A me di solito riesce dopo due o tre ricompilazioni. Come si vede l'ho fatto anche per pareggiare le colonne di questa pagina.

7 Conclusioni

Credo di avere mostrato con esempi significativi l'utilità del comando `\vadjust` nativo di TEX; i semplici esempi che ho mostrato dovrebbero a mio parere essere sufficienti, ma l'appetito viene mangiando, come si suol dire, e i lettori possono giovare per trovare ulteriori soluzioni per situazioni più complesse.

Riferimenti bibliografici

EIJKHOUT, V. (2014). *TEX by topic*. Dante e.V and Lehmanns Media GmbH, Heidelberg and Berlin.

KNUTH, D. E. (1984). *The TEXbook*. Addison Wesley Publ. Co., Reading, Mass.

▷ Claudio Beccari
claudio dot beccari at gmail
dot com

4. Si ricorda che il comando `\enlargethispage` componendo a due colonne è equivalente ad un ipotetico comando `\enlargethiscolumn`, che non esiste. La variante asteriscata in questo caso non deve essere usata, perché sfrutterebbe lo spazio elastico presente nella colonna per cambiarne la lunghezza il meno possibile.

Comporre liste: un viaggio nella grammatica sconosciuta*

James R. Hunt

Sommario

I manuali di scrittura tecnica e le guide di stile pubblicate da università, aziende e altri enti spesso suggeriscono di comporre le liste sulla base di regole che portano a commettere errori grammaticali elementari. Dal punto di vista della grammatica, elencazioni ed enumerazioni sono diverse e gli errori si verificano quando le regole proprie di un tipo di lista vengono applicate alla costruzione dell'altro. Nelle prossime pagine verranno mostrati esempi di liste corrette e scorrette e si indicherà come fare per evitare questi errori. Si vedrà che l'applicazione rigida delle regole grammaticali alla costruzione delle liste rivela alcune interessanti limitazioni di queste strutture.

Abstract

Textbooks on technical writing, and academic, corporate, and other style guides, often prescribe rules for lists that result in basic grammatical errors. Itemised and enumerated lists are grammatically different, and errors arise when the rules for one type of list are used in constructing the other type. Examples of correct and incorrect usage are given, and ways of avoiding errors are described. The strict application of grammatical rules to list construction reveals some interesting limitations of the list form.

1 Introduzione

Tutti conoscono le liste e le utilizzano spesso. Dopo i normali capoversi, esse sono forse le più comuni strutture per mezzo delle quali gli autori di lavori tecnici e scientifici dispongono il testo sulla pagina. Sfortunatamente, i manuali di scrittura tecnica, le guide di stile aziendali e perfino quelle diffuse da enti culturali prescrivono di comporre sulla base di regole che provocano errori grammaticali. Alcune di esse, come quella che prescrive la scrittura

di frasi¹ prive della corretta punteggiatura finale, portano a commettere errori davvero elementari. Lo scopo di questo articolo è di esaminare alcuni di questi errori e cercare di individuare il modo per evitarli.

L'approccio adottato qui è assiomatico: si avanzano alcune ipotesi circa le proprietà che un testo di argomento tecnico dovrebbe avere e se ne esaminano le conseguenze.

1.1 Preliminari e ipotesi di base

Come prima cosa, ecco alcune regole fondamentali preliminari e plausibili da considerare alla stregua di assiomi.

1.1.1 Scrivere in un registro formale

Di solito, le opere di carattere tecnico sono scritte in un *registro formale*. Qui si considererà formale un documento scritto con frasi grammaticalmente corrette e complete e correttamente interpunte. Non verranno presi in considerazione, invece, effettiva qualità e stile della lingua utilizzata.

1.1.2 Migliorare la comprensione

Un'opera di carattere tecnico va scritta con frasi complete e grammaticalmente corrette, che si possono disporre su una pagina a stampa o sullo schermo in qualunque modo aiuti il lettore a comprenderne il contenuto. Per aumentarne ulteriormente chiarezza e comprensibilità, possiamo decorare il testo del documento con ogni mezzo che pensiamo sia necessario: caratteri neri o corsivi; capoversi rientrati; spazi bianchi orizzontali o verticali; simboli, numeri, filetti, illustrazioni e via di seguito.

Analogamente, possiamo applicare al documento alcuni strumenti per “navigare” il testo come intestazioni e sottotitoli, numeri di pagina, didascalie ed elenchi di figure e tabelle, indici, testatine e piedini. Questi strumenti possono consistere di elementi testuali ed essere frasi complete, ma non necessariamente lo sono. Qualunque cosa siano, tuttavia, sono pensati per aiutare il lettore: a comprendere il testo i primi; a trovare elementi specifici nel testo i secondi. Ma né gli uni né gli altri fanno di per sé parte del testo.

1. In questa traduzione ho utilizzato liberamente il termine *frase*, lasciando che di volta in volta significasse *proposizione*, *periodo* ed *enunciato*. Il contesto discriminerà di volta in volta fra un significato e l'altro.

*Questo articolo è una traduzione dell'originale pubblicato su TUGboat 25 (2014), N. 1, pag. 9–15, (HUNT, 2014). Questa traduzione è stata autorizzata dall'autore James Hunt ed è stata eseguita da Claudio Beccari e Tommaso Gordini; se il pensiero di Hunt non è stato tradotto bene, il demerito va attribuito ai traduttori. James R. Hunt vive a Brisbane, in Australia; ha insegnato matematica e ha lavorato nel campo della statistica; ha svolto il ruolo di scrittore tecnico per molti anni nel campo dell'informatica; ora è in pensione e si dedica a L^AT_EX e al software open source.

1.1.3 La forma ellittica

L'ellissi è l'omissione di elementi recuperabili dal contesto e può interessare tanto la punteggiatura quanto intere parole. Ripristinando la prima e le seconde (spesso si tratta di semplici congiunzioni come la *e*) si otterrà una frase completa e grammaticalmente corretta.

Queste frasi vanno costruite con cura, in modo tale che il lettore possa ricostruirne senza sforzo la versione completa e *senza avere la minima impressione di trovarsi di fronte a materiale grammaticalmente scorretto*.

Spesso si ricorre alle frasi ellittiche per ridurre l'apparente complessità delle frasi nei lavori di carattere tecnico e, in particolare, le si utilizza comunemente nella costruzione delle liste.

1.1.4 Non esiste il frammento di frase

Alcuni scrittori si riferiscono alle frasi incomplete o altrimenti sgrammaticate con la locuzione *frammenti di frase*. Il concetto di *frammento di frase* non è realmente utile: frasi incomplete o sgrammaticate non sono accettabili in un lavoro di carattere tecnico, che va scritto con frasi complete o frasi ellittiche ricostruibili con facilità.

Le parole che costituiscono il titolo di un capitolo o di un paragrafo o una didascalia non devono per forza formare una frase completa, e di solito non lo fanno. Questi titoli, infatti, sono solo strumenti di navigazione, ideati per aiutare i lettori a trovare la propria strada in un documento complesso. Li si potrebbe chiamare *frammenti di frase*, allora, ma dato che il loro ambito di applicazione è molto ristretto, non sembra esserci una reale necessità di creare un nome nuovo: possiamo chiamarli semplicemente titoli e didascalie.

1.1.5 Le frasi non possono essere annidate

In inglese² non si possono *annidare* le frasi, cioè non si può inserire una frase completa in un'altra frase come un'entità autonoma. (Naturalmente, si possono costruire frasi elaborate contenenti parti che sarebbero frasi complete se scritte separatamente, ma una frase annidata è altra cosa.³)

1.1.6 Una regola fondamentale di scrittura

Qualunque regola di scrittura provochi un errore grammaticale non serve a nulla e va ignorata. L'applicazione di questa regola alla costruzione di enumerazioni ed elencazioni porta ad alcune conclusioni sorprendenti.

2. Ma anche in italiano. (N.d.T.)

3. L'autore qui si riferisce a frasi coordinate e subordinate a una frase principale. (N.d.T.)

2 Da cosa si vede che è una lista?

Pallini o numeri non fanno una lista. I word processor e gli impaginatori di testi sono in grado di generare simboli da inserire come etichette o una numerazione sequenziale per qualunque parte di testo si voglia specificare, e si possono mettere questi simboli dovunque in un testo. Tuttavia, quei simboli e quei numeri potrebbero non raggiungere lo scopo di aiutare a comprendere i concetti esposti.

2.1 False liste

Esistono liste e *false liste*. Una falsa lista è una raccolta di elementi che non hanno nulla in comune, ma che sono contrassegnati con un pallino o un numero a sinistra della loro prima riga. Nell' "Epoca PowerPoint" le false liste sono molto comuni, perché aggiungere simboli al testo è facilissimo. Piuttosto spesso, però, i pallini, le mani che indicano, le file di formiche in marcia o i numeri non aggiungono nulla alla presentazione: un testo semplice e senza decorazioni sarebbe stato più efficace.

L'esempio errato 1 mostra una falsa lista che si vede comunemente nelle presentazioni PowerPoint.

Esempio errato 1: FALSA LISTA VERTICALE

Argomenti di oggi

- *Cos'è il Compositore automatico di manuali?*
- *Cinque facili passi per il successo*
- *Come funziona*

Non sempre le false liste sono ben allineate in verticale, come mostra l'esempio errato 2.

Esempio errato 2: FALSA LISTA ORIZZONTALE

Sono disponibili tre opzioni, specificatamente: (a) scappare; (b) restare sul luogo ma nascondendosi; (c) restare sul luogo coraggiosamente.

Se si levano parentesi e lettere, la frase è grammaticalmente corretta e anche un po' più facile da leggere. Le lettere fra parentesi non aggiungono nulla alla nostra comprensione e rappresentano semplicemente decorazioni superflue.

2.2 Identificare una lista

Gli elementi di una lista devono avere qualcosa in comune: per esempio, dovrebbero essere tutti collegati ad una singola idea o operazione; e i pallini o i numeri dovrebbero migliorare la chiarezza del materiale esposto. Nel seguito della trattazione insisteremo su questo concetto.

3 Terminologia

L^AT_EX conosce tre tipi di lista: le *enumerazioni*, descritte nel paragrafo 3.2, le *elencazioni*,⁴ descritte nel paragrafo 4, e le *liste di definizioni*.⁵

La lista di definizioni (chiamata anche *descrizione*, *N.d.T.*) rappresenta una stranezza, in un certo senso, perché non sembra nient'altro che una tabella travestita da lista. Se ne parlerà più avanti nel paragrafo 8.

3.1 Altri nomi per le elencazioni

Talvolta ci si riferisce alle elencazioni come *liste puntate* o *non ordinate*. La locuzione *lista puntata* è comune ma non molto utile, dato che non per forza gli elementi di queste liste sono marcati con pallini: al loro posto, infatti, ci potrebbero essere essere lineette, disegni o addirittura nessun contrassegno ed essere disposti semplicemente su righe a sé stanti. Come alternativa più generica di *lista puntata* spesso si utilizza l'espressione *lista non ordinata*.

Nemmeno l'espressione *lista non ordinata* è molto utile, perché comunque gli elementi della lista sono stati composti sulla pagina in un certo ordine che è parso conveniente allo scrittore. Il punto importante è che gli elementi di questa lista possono essere presentati in un altro ordine senza cambiare il significato del messaggio, sebbene la chiarezza della presentazione potrebbe esserne ridotta.

3.2 Enumerazioni

Un'*enumerazione* viene utilizzata per esporre una sequenza di istruzioni, o per elencare componenti o casi, o ancora per segnalare l'ordine d'importanza dei casi elencati. Gli elementi di un'*enumerazione* sono contrassegnati con numeri o lettere o altri simboli disposti secondo una sequenza conosciuta.

Esporre una sequenza di istruzioni è cosa fondamentalmente diversa dagli altri due utilizzi visti poco sopra: enumerando componenti o casi, infatti, l'ordine degli elementi nella lista non è importante, poiché cambiandolo non cambia il significato della presentazione. Talvolta, infatti, si possono scambiare in modo casuale gli elementi di un'*enumerazione* come una lista di componenti per assemblare un dispositivo senza ridurre la chiarezza della presentazione.

4 Elencazioni

Elencazioni ed enumerazioni sembrano simili, ma esiste una differenza fondamentale tra le due, e ciascuna possiede regole di costruzione proprie. Spesso gli scrittori di argomenti tecnici le confondono, nel senso che applicano le regole di costruzione di un tipo di lista all'altro tipo.

4. In italiano vengono spesso chiamate *liste puntate*. (*N.d.T.*)

5. Per queste ultime liste L^AT_EX mette a disposizione l'ambiente standard `description`. (*N.d.T.*)

Un'*elencazione* è semplicemente un mezzo visuale per mostrare sulla pagina una singola frase, di solito complessa. Pallini e altri espedienti tipografici, come rientri e spaziature verticali, non fanno parte della struttura sintattica della frase in sé, ma servono solo ad aiutare la comprensione. Dalla premessa fondamentale esposta nel paragrafo 1.1.2, consegue che una frase scritta in forma di elencazione dovrebbe risultare grammaticalmente corretta dopo aver rimosso questi mezzi visuali. Si consideri l'elencazione mostrata nell'esempio 1.

Esempio 1: ELENCAZIONE SEMPLICE

Sono disponibili tre colori:

- *rosso,*
 - *verde, e*
 - *blu.*
-

Quest'elencazione è soltanto un ridisposizione tipografica della frase seguente.

Sono disponibili tre colori: rosso, verde, e blu.

4.1 Virgole e punti e virgole

Per evitare le ambiguità derivanti dal ripetere la congiunzione *e* in esempi più complessi, possiamo adottare la convenzione di separare gli elementi di un'elencazione con il punto e virgola anziché con la virgola, per cui la lista assume la forma mostrata nell'esempio 2.

Esempio 2: ELENCAZIONE COMPLESSA

Sono disponibili tre combinazioni di colori:

- *rosso, bianco, e blu;*
 - *blu, verde, e giallo; e*
 - *verde, arancio, e bianco.*
-

Questo è il caso classico di una elencazione con la punteggiatura completa.

Dal momento che un'elencazione contiene solo una frase, la prima parola dopo un contrassegno non deve cominciare con la maiuscola, a meno che non si tratti di una qualche forma di nome proprio. L'esempio errato 3 mostra un utilizzo comune ma scorretto di quanto si è appena affermato.

Esempio errato 3: USO ERRATO DELLE INIZIALI MAIUSCOLE

Questo paragrafo fornisce le linee guida per:

- *Creare liste;*
- *Punteggiare le liste; e*
- *Creare liste annidate.*

4.2 Omettere la punteggiatura

Spesso le elencazioni vengono composte in una forma senza punteggiatura, o ellittica. Si consideri l'elencazione mostrata nell'esempio 3. (Alcuni autori di manuali insistono sul fatto che una lista ellittica come questa dovrebbe terminare comunque con il punto fermo. Ma perché preoccuparsene?)

Esempio 3: ELENCAZIONE SENZA PUNTEGGIATURA (FORMA ELLITTICA)

Sono disponibili tre colori:

- *rosso*
- *verde*
- *blu*

4.2.1 Forma ellittica

L'elencazione non interpunta dell'esempio 3 deriva dalla versione completamente interpunta tramite il procedimento dell'*ellissi*.

Alcuni manuali di scrittura tecnica si riferiscono a un'elencazione formata da ellissi come a un'elencazione di frammenti di frase ma, come si è visto nel paragrafo 1.1.4, si tratta di un concetto inutile.

Se non si può ricostruire una versione grammaticalmente corretta della lista appena vista, allora abbiamo a che fare con una falsa lista.

In effetti, molte guide di stile aziendali specificano che invece di un'elencazione completamente interpunta *si deve* utilizzarne la forma ellittica. Non c'è nulla di sbagliato in questa indicazione, naturalmente, a patto di essere consapevoli che quella ellittica è una forma derivata e non la versione completa.

4.2.2 Ellissi e costruzione parallela

Molte guide di stile specificano che gli elementi in un'elencazione dovrebbero rivelare una *costruzione parallela*, cioè dovrebbero essere sintatticamente simili. La costruzione parallela è utile poiché rende più facile comprendere il materiale presentato e più facile recuperare la forma completa di una lista a partire dalla sua forma ellittica.

5 Enumerazioni

Un'elencazione è un mezzo visuale per mettere in mostra una singola frase. Al contrario, un'*enumerazione*, con numeri o lettere come contrassegni, è un mezzo visuale per mettere in mostra un frammento testuale comprendente un certo numero di frasi. Queste frasi possono, ma non necessariamente lo fanno, descrivere tutte insieme una sequenza di eventi nel tempo e nello spazio. Talvolta, la forma di un'enumerazione è utilizzata per indicare il numero dei componenti in una collezione, o dei casi sotto esame. Di solito, la cosa risulta evidente dal contesto.

Se si vogliono generare riferimenti incrociati a singoli elementi di una lista, allora questi ultimi devono far parte di un'enumerazione.

L'esempio più comune di enumerazione è costituito da una serie di istruzioni da eseguire secondo un ordine cronologico rigido. Si consideri la serie di istruzioni nell'esempio 4.

Esempio 4: ISTRUZIONI SEQUENZIALI COMPOSTE IN LINEA

Le istruzioni per eseguire la manutenzione del dispositivo sono le seguenti. Aprire il pannello superiore del roborazer. Inserire il cacciavite nel foro a sinistra. Girare il cacciavite in senso orario finché la pressione si azzerà. Richiudere il pannello superiore.

Di solito, presentiamo una serie di istruzioni come questa in forma di lista numerata o contrassegnata con lettere, introducendola con un'intestazione, o una frase, o entrambe le cose. L'intestazione è solo una decorazione del testo e come tale non ha bisogno di essere grammaticalmente corretta o interpunta. La frase introduttiva deve essere una frase completa, naturalmente, e conclusa con il punto fermo (o interrogativo o esclamativo, se appropriato). L'esempio 5 mostra un'enumerazione con intestazione e frase introduttiva.

5.1 Introdurre una enumerazione senza usare i due punti

Molti scrittori concludono la frase introduttiva prima di un'enumerazione con i due punti. *Questo è un errore grammaticale*: in un lavoro di carattere tecnico, ogni frase deve essere completa e terminare con un punto fermo o interrogativo o esclamativo, ma *mai* con i due punti. Questo errore straordinariamente comune deriva dal confondere le regole per comporre un'enumerazione con quelle per comporre un'elencazione.

Esempio 5: ENUMERAZIONE CON INTESTAZIONE E FRASE INTRODUTTIVA

Manutenzione del dispositivo

Per eseguire la manutenzione ordinaria si eseguano le seguenti operazioni.

1. Aprire il pannello superiore del roborazer.
2. Inserire il cacciavite nel foro a sinistra.
3. Girare il cacciavite in senso orario finché la pressione si azzerà.
4. Richiudere il pannello superiore.

5.2 Decorazioni dei numeri

In un'enumerazione, i numeri accanto agli elementi sono decorazioni — cioè semplici mezzi visuali che aiutano il lettore a comprendere il materiale — senza alcun significato sintattico. La stessa cosa vale per punti fermi, parentesi, caratteri neri, animazioni o altri dispositivi aggiunti ai numeri. L'ordine da seguire nelle azioni è determinato dall'ordine in cui sono esposte le frasi e non da quello dei numeri, che servono solo a rinforzare la sequenza nella mente del lettore.

Il punto fermo o le parentesi dopo i numeri o le lettere potrebbero suggerire al lettore l'esistenza di una struttura sintattica che in realtà non c'è, e questa è una buona ragione addirittura per evitare tali decorazioni. Si fa prima a dirlo che a farlo, però: la maggior parte dei software di scrittura, infatti, le inserisce automaticamente.

Talvolta si potrebbe dover evidenziare con un'adeguata enfasi i numeri: altrimenti, le liste di istruzioni presenti in un manuale per l'utente, per esempio, rischierebbero di annegare in masse di materiale esplicativo (risultando poco visibili).

6 Complicazioni e restrizioni

L'applicazione rigida delle regole esaminate nei paragrafi precedenti produce alcuni risultati interessanti, descritti dettagliatamente nelle prossime sezioni.

Si può attaccare un'elencazione accessoria a un singolo elemento di un'elencazione (paragrafo 6.1).

Mettere un capoverso esplicativo dopo un elemento di un'elencazione è un errore grammaticale (sezione 6.2).

Si può attaccare un'elencazione a un singolo elemento di un'enumerazione (sezione 6.3).

Non si può attaccare un'enumerazione accessoria a un singolo elemento di un'elencazione (sezione 6.4).

6.1 Elencazioni annidate

È possibile, ed è abbastanza comune, inserire elencazioni secondarie sotto singoli elementi di una lista di livello superiore. Esistono addirittura regole largamente accettate su quali contrassegni utilizzare in questo caso: pallini nel primo livello, lineette nel secondo, eccetera. Si ricordi che un'elencazione comprende solo una frase: ne consegue che tutti gli elementi della lista, considerati insieme, devono costituire quest'unica frase e basta. Tentare di seguire questa regola potrebbe portare facilmente a costruzioni complesse che perdono in chiarezza. L'esempio 6 mostra un'elencazione su due livelli costruita secondo la regola e che rimane chiara. (Una frase che si possa mostrare come un'elencazione su tre livelli dovrebbe essere piuttosto complicata, a dir poco.)

Esempio 6: ELENCAZIONE A DUE LIVELLI

Il colore del corpo del dispositivo può essere:

- *un colore primario, che può essere uno fra:*
 - *rosso,*
 - *giallo, o*
 - *blu; o*
- *un colore pastello, che può essere uno fra:*
 - *rosa,*
 - *giallo chiaro, o*
 - *celeste; o*
- *un colore neutro, che può essere uno fra:*
 - *beige,*
 - *avorio, o*
 - *canapa.*

6.2 I capoversi esplicativi non sono ammessi nelle elencazioni

È una pratica comune quella di inserire capoversi esplicativi dopo gli elementi in un'elencazione. *Questo è un errore grammaticale*: un'elencazione comprende solo una frase, e non è permesso annidare altre frasi al suo interno.

6.3 Elencazione dopo un elemento numerato

In un'enumerazione, si possono avere due o più frasi dopo ciascun numero, se lo scrittore lo ritiene necessario — e, naturalmente, una qualunque di queste frasi può essere messa in mostra in forma di elencazione se appropriato (cioè, l'elemento di un'enumerazione può essere a propria volta seguito dai pallini).

Alcune di quelle frasi successive possono essere messe in mostra con i loro propri simboli di ordinamento (cioè, un'enumerazione accessoria può seguire l'elemento di un'enumerazione).

Questa pratica è usuale e non presenta problemi concettuali.

6.4 Le enumerazioni non sono ammesse nelle elencazioni

Come s'è visto, possiamo mettere un'elencazione dopo l'elemento numerato di un'enumerazione. Possiamo, al contrario, mettere un'enumerazione dopo l'elemento puntato di un'elencazione? Come nel caso del leggendario centro sciistico pieno di ragazze in cerca di mariti e di mariti in cerca di ragazze, la situazione non è così simmetrica come potrebbe apparire a prima vista.

Si ricordi che gli elementi di un'elencazione formano una singola frase, e che ogni elemento numerato contiene una o più frasi complete. Come le frasi non possono essere ragionevolmente annidate all'interno di altre frasi, così un'enumerazione *non può* essere messa dopo l'elemento di un'elencazione senza violare le regole grammaticali.

Nei lavori di carattere tecnico, di solito si utilizzano elementi contrassegnati con un pallino al posto di innumerevoli tipi di intestazione: all'elemento in questione, spesso enfaticizzato in nero, seguono capoversi esplicativi. Riflettendoci sopra, è difficile trovare qualche ragione per voler fare una cosa del genere — perché mai si vorrebbero disporre serie di istruzioni su una o più pagine e supporre che il loro ordine di presentazione al lettore sia irrilevante? La migliore soluzione per questo problema è quella di evitarlo: possibilmente riscrivendo questi elementi come intestazioni seguite da testo e lasciando ciascuna enumerazione nel testo per conto proprio.

7 Patologie

Spesso i lavori di carattere tecnico contengono liste costruite in modo strano: enumerazioni travestite da elencazioni, elencazioni travestite da enumerazioni, costruzioni ibride e mostruosità di vario tipo. Spesso questi errori derivano dalla mancata comprensione della differenza fondamentale esistente tra enumerazioni ed elencazioni.

7.1 Enumerazioni travestite da elencazioni

L'errore più comune sembra quello di costruire le enumerazioni come delle elencazioni.

L'esempio errato 4, preso da una guida di stile aziendale, sembra un'elencazione ma in realtà contiene quattro frasi, una delle quali non è conclusa correttamente.

La lista mostrata nell'esempio errato 4 dovrebbe essere presentata più correttamente come un'enu-

merazione introdotta da un capoverso, come si mostra nell'esempio 7.

Esempio errato 4: ENUMERAZIONE TRAVESTITA DA ELENCAZIONE

Usare le seguenti linee guida per creare le appendici:

- *Elencare le appendici nell'indice generale.*
 - *Citare le appendici nella prefazione.*
 - *Per ciascuna appendice scrivere un capoverso introduttivo.*
-

I numeri nell'esempio 7 non rappresentano necessariamente una sequenza di azioni, ma possono semplicemente chiarire il numero di regole da seguire.

Esempio 7: CORREZIONE DELL'ESEMPIO ERRATO 4

Usare le seguenti linee guida per creare le appendici.

1. *Elencare le appendici nell'indice generale.*
 2. *Citare le appendici nella prefazione.*
 3. *Per ciascuna appendice scrivere un capoverso introduttivo.*
-

Esempio errato 5: ELENCAZIONE TRATTATA COME ENUMERAZIONE

Gli oggetti contenuti nel kit di protezione contro in vampiri sono i seguenti:

1. *una pistola affidabile con i suoi soliti accessori.*
 2. *proiettili d'argento;*
 3. *un crocifisso d'avorio;*
 4. *fiori d'aglio in polvere;*
 5. *un paletto di legno;*
 6. *il nuovo siero del professor Blomberg.*
-

7.2 Elencazioni travestite da enumerazioni

L'esempio errato 5 sembra una enumerazione, ma i numeri non servono a nulla: non indicano passi da compiere, o l'ordine in cui gli elementi possono essere utilizzati, o un numero di elementi (quanti elementi sono contenuti nei "suoi soliti accessori"?), o un ordine d'importanza.

L'informazione contenuta nell'esempio errato 5 può essere presentata meglio in forma di

elencazione, come mostra l'esempio 8.

Esempio 8: VERSIONE CORRETTA DELL'ESEMPIO ERRATO 5

Gli oggetti contenuti nel kit di protezione contro in vampiri sono i seguenti.

- *Una pistola affidabile con i suoi accessori abituali.*
- *Proiettili d'argento.*
- *Un crocefisso d'avorio.*
- *Fiori d'aglio in polvere.*
- *Un paletto di legno.*
- *Il nuovo siero del professor Blomberg.*

Esempio errato 6: UNA LISTA SOVRACCARICA

[Le] otto principali lingue in Cina [...]

- *Cantonese (Yúe)* Parlato al sud, specialmente nel Guangdong, nel Guangxi meridionale, Macao, Hong Kong (46 milioni)
- *Gan* Parlato nello Shanxi e nel sud ovest dell'Hebei (21 milioni)
- *Hakka* Molto diffuso, specialmente tra il Fujian e il Guangxi (26 milioni)
- *Mandarino* Una grande collezione di varietà nelle regioni di nord est, del centro e dell'ovest. Il mandarino del nord, quello che si parla a Pechino, è la base della lingua standard moderna (720 milioni)
- *Min settentrionale* Parlato nel nord ovest del Fujian (10 milioni)
- *Min meridionale* Parlato nel sud est, specialmente in parte dello Zhejiang, Fujian, nell'isola Hainan e a Taiwan (26 milioni)
- *Wu* Parlato in parti dell'Anhui, Zhejiang e Jangsu (26 milioni)
- *Xian (Hunan)* Parlato nella regione centro meridionale, nel Hunan (36 milioni)

7.3 Talvolta una tabella è un'idea migliore

Esiste anche la possibilità di avere troppe cose da mettere in una lista, e talvolta le cose sarebbero più chiare se il materiale fosse composto in forma di tabella. L'esempio errato 6, che mostra una lista troppo piena, è tratto da un diffuso manuale di linguistica (CRYSTAL, 2007).

Si potrebbe esporre meglio questo materiale con una tabella di tre colonne — possibilmente con delle elencazioni in alcune celle.

8 Liste di definizioni

La *lista di definizioni*, citata nel paragrafo 3, è un tipo di lista ancora diverso, poiché i suoi elementi non sono contrassegnati né con un pallino né con un numero, ma comprendono due parti: un termine e la sua spiegazione. Di solito, queste due parti sono distinte tipograficamente e possono comparire sulla stessa riga, ma non necessariamente lo fanno.

La maggior parte dei word processor non offre un'opzione corrispondente alla lista di definizioni nei vari menù a tendina, perché una tabella di due colonne, senza didascalia, filetti o intestazioni (cioè una *tabella informale*) fa più o meno lo stesso lavoro. Vedi però più avanti nel paragrafo 8.2.

Si può utilizzare una lista di questo tipo per comporre gli elementi di un glossario, come mostra l'esempio 9. (Questo materiale è tratto da BAICHEL (2009).)

Esempio 9: UNA LISTA DESCRITTIVA

Klingon Questa è probabilmente la lingua più completa per testi di fantascienza. Il Klingon dispone di una grammatica completa e di un vocabolario, e un numero enorme di appassionati l'ha imparato come si impara a scuola il francese o il tedesco.

Qwghimian Nel romanzo *Cryptonomicon* di Neal Stephenson e nella trilogia *Il ciclo barocco* si suppone che questa lingua di fantasia sia parlata in alcune ignote isole britanniche. La lingua ha sedici consonanti e nessuna vocale, ed è perciò particolarmente adatta per rappresentare l'informazione in forma binaria e, ovviamente, la sua pronuncia è quasi impossibile.

R'lyehian Quest'altra lingua con parole strane, a malapena pronunciabili, fa parte del mito di Cthulhu (introdotto nel classico racconto breve Lovecraft *Il culto del Cthulhu*).

Sindarin Mentre Tolkien ha creato diverse lingue per i vari libri della serie di *Il signore degli anelli*, il sindarin, la lingua degli elfi, non è solo la sua migliore invenzione linguistica, ma anche quella maggiormente e completamente sviluppata.

8.1 Uso delle liste di definizioni

Gli usi più canonici di una lista di definizioni si ritrovano nella composizione delle definizioni e nella costruzione di glossari. Esistono altri utilizzi possibili, anche se non così ovvi: per esempio, potrebbero servire per comporre liste costituite di

un solo elemento, vietate da alcune guide di stile, altrimenti scomode.

8.2 I comuni word processor non contemplano le liste di definizioni

Costruire una lista di definizioni con il layout “avvolgente” mostrato nell’esempio 9 è abbastanza facile con L^AT_EX, ma i comuni word processor non offrono voci di menù corrispondenti. Le tabelle informali di due colonne di solito saranno una buona approssimazione.

8.3 Storia delle liste di definizioni

La lista di definizioni è definita in HTML, nella specifica DITA (Darwin Information Typing Architecture: un *data model* XML adatto alla creazione di programmi e database informatici) del 2005, nella specifica DocBook del 1990 e nella specifica L^AT_EX del 1986; sembra che la sua origine vada ricercata nell’impaginatore di testi Scribe del 1978 circa, ora quasi dimenticato. Lo scopo dell’autore di Scribe era quello di fornire un modo semplice per comporre le descrizioni dei comandi nei manuali di programmazione.

Scribe non era in grado di gestire le tabelle, ma L^AT_EX sì, entro un certo limite. Le uniche tabelle che L^AT_EX riusciva a gestire all’inizio erano lunghe meno di una pagina e dovevano essere *galleggianti*: la posizione della tabella nella versione finale a stampa del documento, cioè, era determinata da un algoritmo che determinava la posizione della tabella in modo che non coincidesse con un’interruzione di pagina. Finché esistette questa limitazione della pagina unica, la lista di definizioni rimase comunque utile, perché poteva essere utilizzata per comporre materiale tabulare che occupava più di una pagina.⁶

9 Conclusioni

Applicare poche semplici regole a elencazioni ed enumerazioni porta ad alcune conclusioni inaspettate. In particolare, le elencazioni sono di portata molto più limitata di quanto non sembri a prima vista. In molte circostanze, le enumerazioni sono molto più utili.

Le liste di definizioni sono utilizzate dagli scrittori tecnici meno di quanto dovrebbero esserlo.

Riferimenti bibliografici

- BAICHEL, J. (2009). «Top ten geekiest constructed languages». URL <http://www.wired.com/geekdad/2009/08/top-ten-geekiest-constructe-languages>.
- CRYSTAL, D. (2007). *How Language Works*. Avery, New York.
- HUNT, J. R. (2014). «Making Lists: A Journey into Unknown Grammar». *TUGboat*, **35** (1), pp. 9–15.
- MICROSOFT CORPORATION (2004). *Microsoft Manual of Style for Technical Publications*. Microsoft Press, Sebastopol CA, 3^a edizione.

▷ James R. Hunt
P.O. Box 580
Mt Gravatt
Queensland 4122
Australia
writerlyjames at gmail dot com

6. Questa descrizione ha valore storico; è noto che l’ambiente `longtable` ed altri ambienti simili, esistono almeno da quando esiste L^AT_EX 2_ε, quindi il limite di una pagina per le tabelle non è più veramente tale da molti anni; benché a parere dei traduttori, `longtable` sia il miglior pacchetto per comporre tabelle lunghe, esistono diversi altri pacchetti che permettono di costruire tabelle senza il limite del “fine pagina”; vedi la lista di domande frequenti all’indirizzo web <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=longtab>. Inoltre esisteva fin dall’inizio di L^AT_EX l’ambiente `tabbing` che consentiva, allora come oggi, di comporre materiale incolonnato che non soffriva della limitazione del “fine pagina”. Le tabelle flottanti, come le figure flottanti, sono normalmente usate in tipografia; quindi l’approccio di L^AT_EX era tipograficamente corretto, anche perché L^AT_EX si presentava agli utenti non come un word processor, ma come un motore di composizione tipografica. (*N.d.T.*)

Personalizzazione e riproducibilità nel disegno programmato: introduzione al pacchetto pgfkeys

Claudio Fiandrino

Sommario

Il paradigma chiave-valore è una delle principali potenzialità della famiglia di pacchetti PGF. Cambiando poche opzioni nel codice si ottiene alta personalizzazione e facile riproducibilità dei disegni. L'articolo illustra alcuni metodi per utilizzare in modo efficiente chiavi e stili di TikZ.

Abstract

The key-value paradigm is one of the most important features of the packages belonging to the PGF family. By changing few options it is possible to achieve high customization and easy reproducibility of pictures. This article shows the methods to efficiently use keys and styles with TikZ.

1 Introduzione

Come è ben noto, utilizzare i linguaggi di disegno programmato che $\text{\LaTeX}$ offre come TikZ (TANTAU, 2013) permette di ottenere alta qualità nel disegno. In particolare, un segno distintivo è la coerenza tra i font del testo e delle figure, una caratteristica che è più difficile ottenere lavorando con software esterni.

Dal punto di vista del flusso di lavoro, il dibattito in merito alla qualità e potenzialità dei linguaggi di disegno programmato rispetto ad altri software dedicati, nell'ambito del disegno scientifico, ad oggi è ancora aperto. Un punto a favore di questi ultimi è sicuramente la rapidità nel realizzare disegni semplici, per i quali con pochi colpi di mouse si includono le parti essenziali della figura. Quando si parla di disegni complessi, *probabilmente* le tempistiche di realizzazione sono comparabili. L'incertezza di chi scrive, visibile nel corsivo, è motivata dall'assenza di letteratura scientifica comparativa a riguardo.

Molto spesso la rapidità di realizzazione è l'unica metrica considerata nel confronto tra le metodologie e probabilmente l'ostacolo più serio per chi si affaccia per la prima volta al mondo del disegno programmato. Tuttavia, ci si scorda sempre che le valutazioni basate su una metrica sola sono, quanto meno, mai completamente veritiere. Parlando del disegno programmato, oserei dire *bugiarde*. Il motivo è molto semplice: considerando altri parametri di confronto come la *personalizzazione* e la *riproducibilità*, probabilmente la bilancia pende verso il disegno programmato.

Con *personalizzazione* in questo articolo si intende la capacità di definire in modo consistente le proprietà degli oggetti in una figura e poterle cambiare in modo semplice. Per *riproducibilità* invece, si intende la capacità, in termini di codice e metodologia, di utilizzare interamente o singole parti di figure preesistenti ed eventualmente apportare modifiche o aggiunte.

Questi aspetti fondamentali appartengono ad ogni linguaggio di disegno programmato: non solo TikZ quindi, ma anche PSTricks, Asymptote e Metapost tra gli altri. In questo articolo ci si focalizza su TikZ, evidenziando i meccanismi con cui si ottiene alta personalizzazione e riproducibilità. In particolare, quindi, si prendono in esame le funzionalità base del pacchetto pgfkeys.

2 Personalizzazione e riproducibilità

In questa sezione analizziamo meglio le due metriche personalizzazione e riproducibilità considerando come ambito di riferimento disegni tecnici e scientifici.

2.1 Personalizzazione

Questa indica la capacità di definizione e cambiamento delle proprietà estetiche dei vari elementi caratterizzanti la figura.

Prendiamo in esame la figura 1. Come si può notare, il gruppo di nodi è caratterizzato da uno stile omogeneo che ne definisce il colore, la dimensione e l'ombreggiatura. Ogni proprietà è sempre sotto forma di un input di tipo *chiave-valore*. A volte sia il valore che la chiave possono essere sottintesi: per esempio, `circle` è una forma breve per `shape=circle` mentre in `draw` il valore non è specificato in quanto di default assume, dove presente, un colore indicato fra le opzioni (`violet` in questo caso), oppure il nero.

Prendiamo ora in esame la figura 2. Il risultato mostra i nodi con due differenti colorazioni, un trucco utile per distinguere elementi con proprietà differenti, come ad esempio gli utenti appartenenti a due piconet diversi in Bluetooth.

È importante notare il procedimento con cui si è ottenuta questa personalizzazione. Si osservi che gli stati, `stato 1` e `stato 2` sono del tutto analoghi a `stato` in figura 1, soltanto la colorazione è differente. La seconda osservazione riguarda il posizionamento: mentre in figura 1 si è usato un

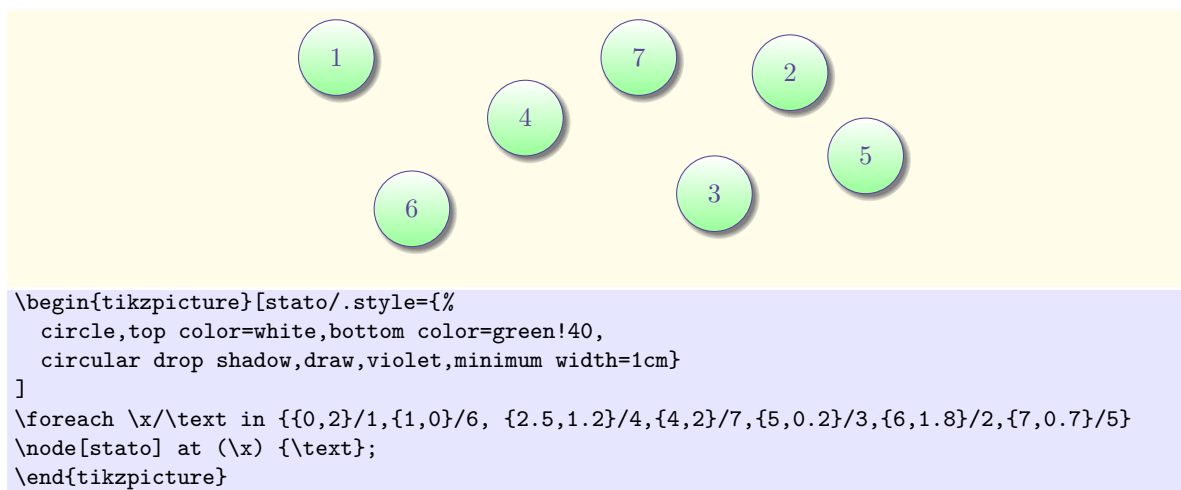


Figura 1: Un gruppo di nodi omogeneo

unico ciclo, in figura 2 è stato necessario spezzare il ciclo in due per poter assegnare i due diversi stili.

Per cambiare le proprietà degli elementi non è quindi stato necessario agire su ogni singolo elemento della figura.

2.2 Riproducibilità

Questa indica la capacità di riutilizzare interamente o parte di disegni realizzati precedentemente da estendere per creare la nuova figura. Quando si ha a disposizione un po' di materiale, la creazione di una figura risulta molto spesso il risultato di un'opera di collage, non solo dei singoli elementi, ma soprattutto delle tecniche utilizzate durante il disegno. Il concetto è ben espresso da Andrew Stacey, di cui riporto un piccolo estratto di un intervento ben più lungo: ¹

Repetition I do a lot of diagrams, but not so many that I'm a graphical designer. Every time that I do a new diagram there's a certain amount of *I've done something like this before, how did I do it?* with subsequent cut-and-pasting. When I use a graphical system then all of the *How did I do it?* information is lost. I can cut-and-paste actual objects, but it's rarely the objects that I want to copy - it's the how did I get that particular effect that I want to copy. When programming a diagram, all of this is laid out in an easily copy-able form.

Supponiamo ora di voler collegare fra loro gli elementi della figura 2 distinguendo i tipi di comunicazione. Il risultato è riportato in figura 3. Come si può notare, l'estensione al codice originale è molto semplice. Tuttavia, quello che è importante osservare è la tecnica utilizzata per l'inserimento

delle frecce: il procedimento con i due cicli è del tutto analogo a quello già visto per il posizionamento dei nodi in figura 2.

3 Aspetti chiave del pacchetto pgfkeys

Numerosi pacchetti L^AT_EX implementano sistemi chiave-valore. In (WRIGHT e FEUERÄSNGER, 2009) gli autori forniscono una panoramica generale, evidenziando chiaramente le distinzioni fra il pacchetto `pgfkeys` e il resto dei pacchetti. Due sono le differenze sostanziali. La prima è la gestione delle chiavi gerarchica con struttura ad albero. La seconda è la capacità di definire *stili*, ossia raggruppare arbitrariamente un certo numero di parametri. Come abbiamo visto nel paragrafo 2, gli stili sono un aspetto fondamentale per garantire personalizzazione e riproducibilità dei disegni.

In questa sezione prendiamo in analisi gli aspetti più importanti di `pgfkeys`, utili non soltanto ad autori di pacchetti basati su TikZ, ma anche a tutti i normali utenti del pacchetto. Capire questi meccanismi, in modo particolare l'uso degli stili, permette di aumentare notevolmente la produttività nel disegno.

3.1 Le chiavi

Il pacchetto `pgfkeys` definisce diversi tipi di chiavi che nel complesso permettono di creare interfacce anche molto complesse. In questo articolo si limita lo scopo dello studio agli strumenti per definire ed utilizzare nuove chiavi.

Le fasi di definizione e utilizzo, concettualmente distinte, devono in realtà essere affrontate insieme data la stretta relazione che intercorre fra le due. Infatti, a seconda di come si utilizza il valore della chiave nel codice, si possono distinguere due famiglie di metodi:

1. <http://tex.stackexchange.com/a/52547/13304d>

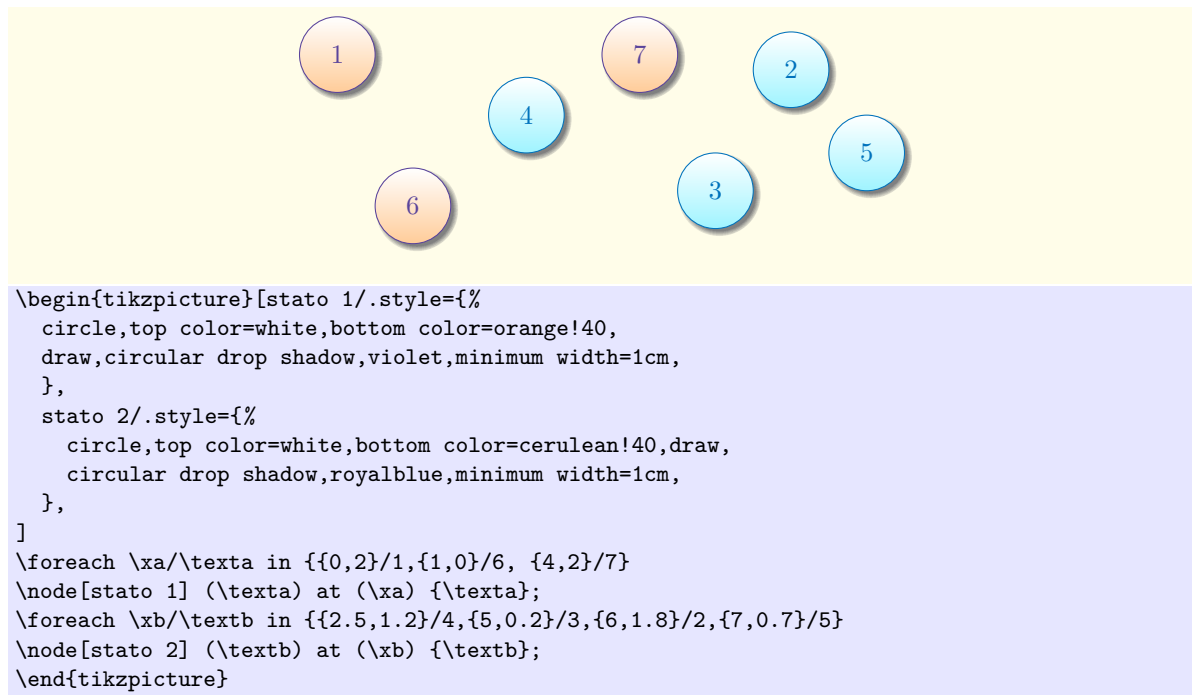


Figura 2: Un gruppo di nodi non omogeneo

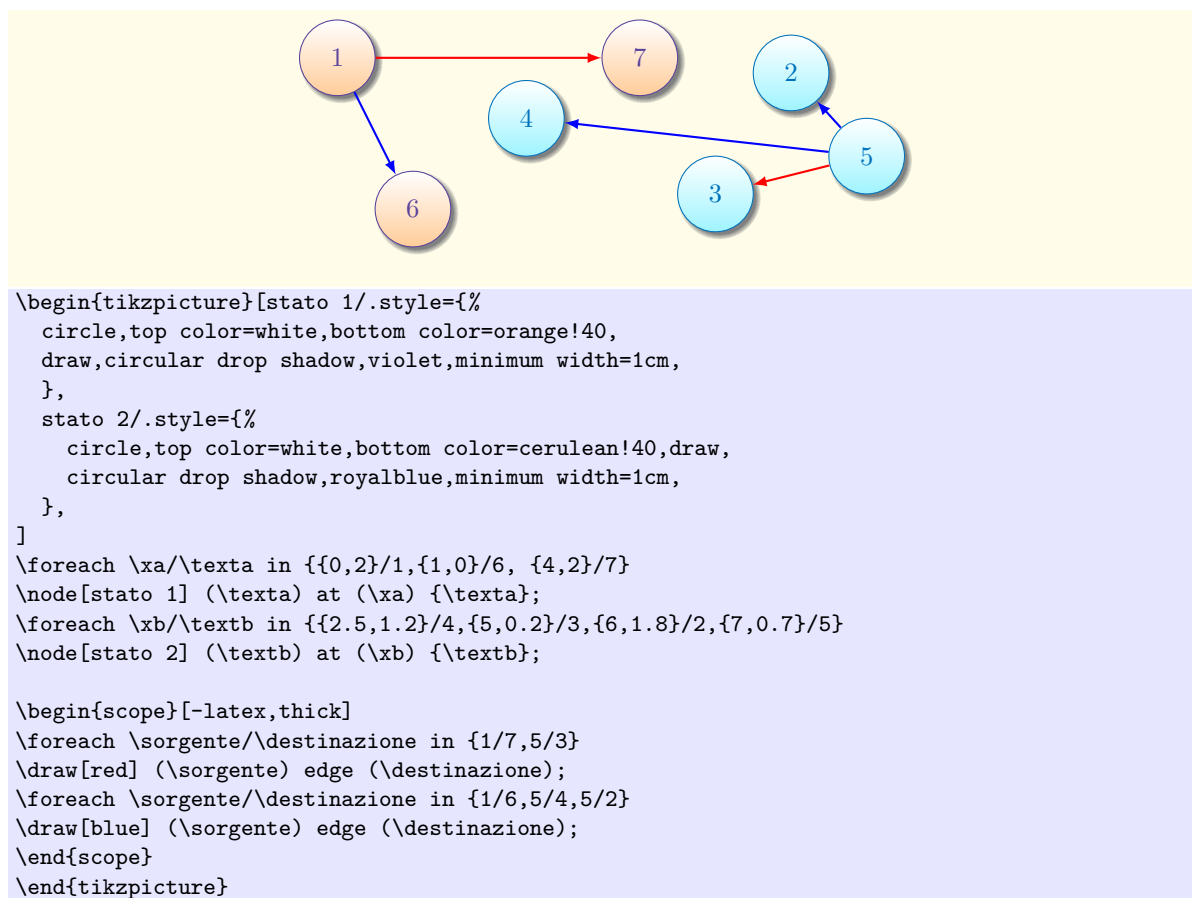


Figura 3: Un gruppo di nodi non omogeneo con aggiunta

- la prima estrae il valore della chiave tramite il comando `\pgfkeysvalueof{\langle chiave \rangle}`;
- la seconda inserisce direttamente nel codice una macro che assume il valore della chiave.

Utilizzando la prima famiglia, è preferibile effettuare la definizione attraverso:

```
\pgfkeys{/percorso/chiaive/.initial = valore}
```

Utilizzando la seconda famiglia, invece, definire le chiavi con il precedente metodo potrebbe non andare bene. Per esempio, nelle prime versioni di Sa-TikZ (FIANDRINO, 2013), il meccanismo utilizzato era:

```
\pgfkeys{/tikz/.cd,%
  N/.initial=10,%
  N/.get=\N,%
  N/.store in=\N,%
}%
```

Come si può notare, anche qui era presente il costruito `/.initial`. Tuttavia, questo metodo genera internamente tre macro:

- `\pgfk@/tikz/N` per `/.initial`, il quale a sua volta richiama internamente `\pgfkeyssetvalue`;
- `\pgfk@/tikz/N/.@cmd` per `/.get`, il quale a sua volta richiama internamente `\pgfkeysgetvalue`;
- `\N` generata da `.store in`.

Per evitare questo inconveniente, è sufficiente modificare di poco la definizione, in modo tale che ad ogni chiave venga associata una e una sola macro:

```
\pgfkeys{/tikz/.cd,%
  N=10,%
  N/.store in=\N,%
}%
```

L'ultima versione di Sa-TikZ il meccanismo di gestione delle chiavi è implementato così.

3.2 Gli stili

Nel paragrafo 2 abbiamo già visto come gli stili permettano di ottenere personalizzazione e riproducibilità dei disegni. Tuttavia, le definizioni degli stili utilizzate per le figure 1, 2 e 3 sono estremamente semplici. In questa sezione, invece, analizziamo più a fondo come sfruttare meglio gli strumenti che il pacchetto `pgfkeys` mette a disposizione. Idealmente, questa sezione sviluppa ed estende i concetti già introdotti in (PANTIERI e GORDINI, 2014).

L'aspetto chiave da tenere presente è che gli stili, come le macro in $\text{\LaTeX}$ possono ricevere argomenti. In questo modo, alcuni parametri dello stile possono variare volta per volta aumentando la flessibilità del codice. Per illustrare questo concetto, si utilizzerà come riferimento lo stile `stato` della figura 1:

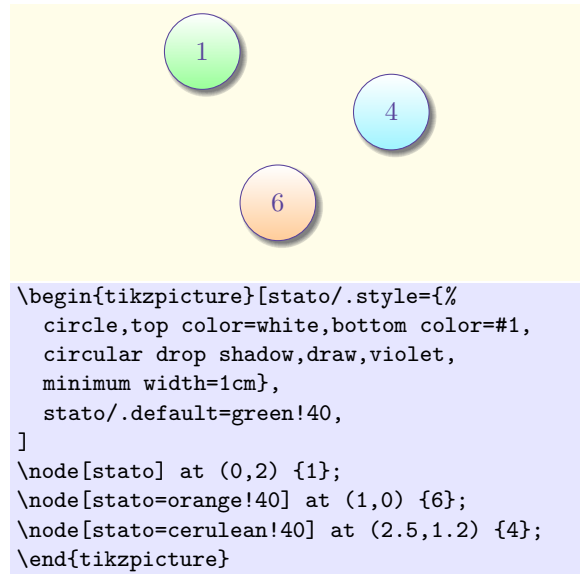


Figura 4: Uso di uno stile con argomento

```
\tikzset{stato/.style={%
  circle,top color=white,
  bottom color=green!40,
  violet, circular drop shadow,
  draw,minimum width=1cm
}}
```

Supponiamo di voler essere liberi di cambiare il colore di riempimento, in questo caso `bottom color=green!40`. Logicamente, avrebbe poco senso creare n stili con n colori diversi. Invece, è molto più comodo fare in modo che la chiave `bottom color` venga impostata tramite argomento:

```
\tikzset{stato/.style={%
  circle,top color=white,
  bottom color=#1,
  violet, circular drop shadow,
  draw,minimum width=1cm
},
  stato/.default=green!40,
}
```

La figura 4 illustra la flessibilità introdotta dal metodo. Si noti come grazie a `stato/.default=green!40` sia possibile dotare lo stile di un argomento di default: questo meccanismo permette di poter utilizzare:

```
\node[stato] at (0,2) {1};
```

senza incorrere in errori.

In realtà, è perfettamente possibile inserire n argomenti agli stili, dove $n < 9$, con la sintassi:

```
stile/.style n args={num arg}{par}
```

dove il primo argomento è il numero di argomenti che si vogliono inserire e il secondo la lista di parametri che definiscono lo stile. Ad esempio:

```
\tikzset{stato/.style n args={3}{%
  circle,top color=#1,
```

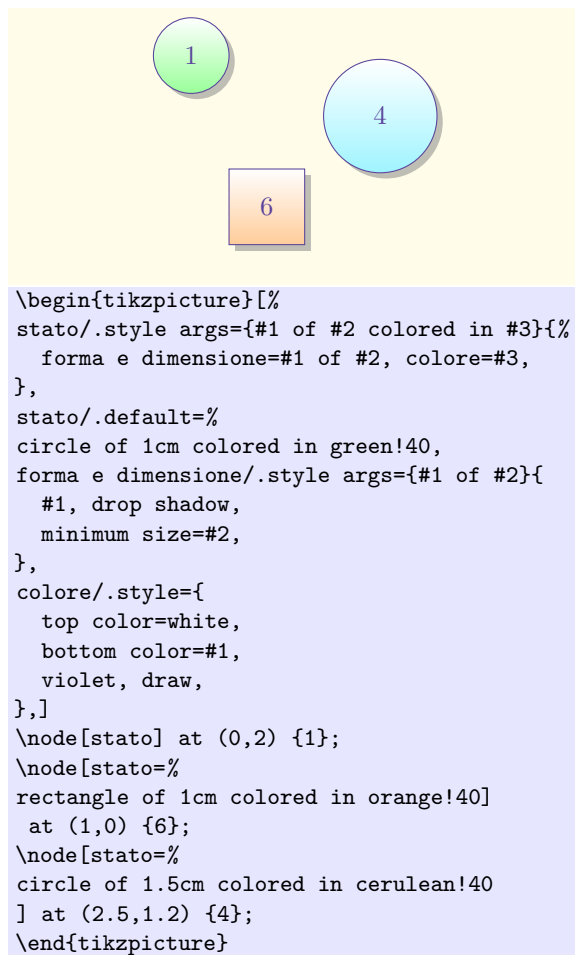


Figura 5: Stili con proprietà complesse

```

bottom color=#2,
violet, circular drop shadow,
draw,minimum width=#3
},
stato/.default={white}{green!40}{1cm},
}

```

Attenzione: sebbene aumentando il numero di argomenti aumenta la flessibilità dello stile, usare un elevato numero di argomenti non è buona pratica in quanto il codice perde in leggibilità e facilità di utilizzo.

Un secondo metodo per migliorare personalizzazione e riproducibilità del codice è definire il più possibile *stili atomici*. Lo stile `stato`, per esempio non è *atomico* in quanto proprietà diverse come colore, forma e dimensione sono mescolate fra loro. Si può rendere più funzionale `stato` in questo modo:

```

\tikzset{stato/.style={%
  forma e dimensione,
  colore,
},
  stato/.default=%
  circle of 1cm colored in green!40,
  forma e dimensione/.style args={#1 of #2}{
    #1, drop shadow,

```

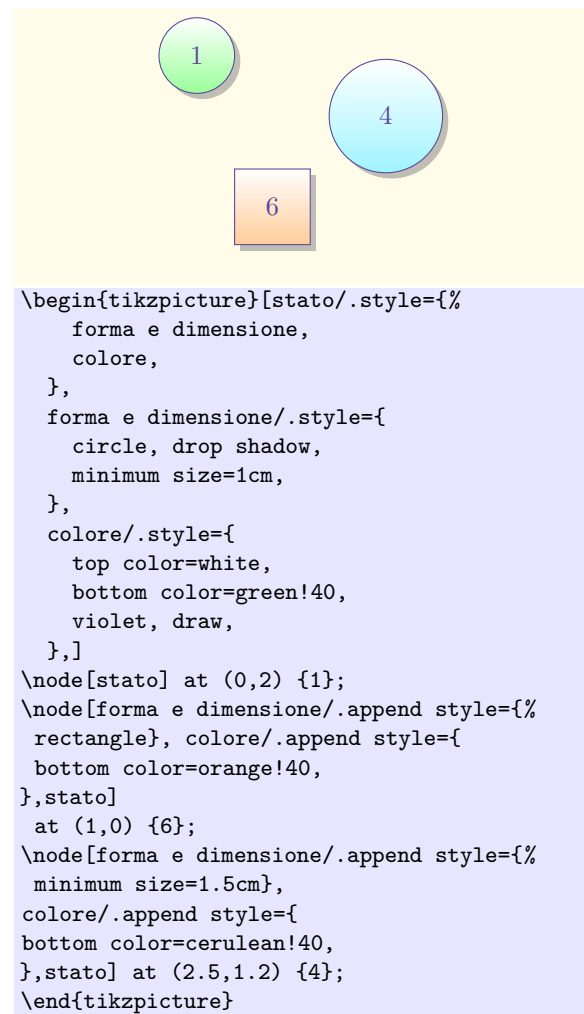


Figura 6: Alterare le proprietà degli stili

```

minimum size=#2,
},
colore/.style={
  top color=white,
  bottom color=#1,
  violet, draw,
},
}

```

Il risultato di questa operazione è riportato in figura 5 utilizzando la proprietà degli stili di definire appositi *pattern* di argomenti. Occorre osservare che questa pratica è un primo passo verso la definizione di stili con proprietà atomiche, ma decisamente complesso. Oltretutto, fornisce soltanto una parvenza di flessibilità: per cambiare una delle proprietà, colore ad esempio, è infatti necessario riscrivere interamente il *pattern* di argomenti.

Invece, sfruttando un'altra proprietà degli stili si possono ottenere soluzioni molto più semplici. Come è noto, gli stili vengono definiti come una lista di parametri chiave-valore. È importante notare che in caso due parametri definiscano diversi valori per la stessa chiave, solo l'ultimo nella lista viene effettivamente considerato. Nel caso di riferimento, quindi, è sufficiente la sintassi:

```
\tikzset{stato/.style={%
    forma e dimensione,
    colore,
},
forma e dimensione/.style={
    circle, drop shadow,
    minimum size=1cm,
},
colore/.style={
    top color=white,
    bottom color=green!40,
    violet, draw,
},
}
```

La figura 6 illustra come utilizzare nella pratica il metodo descritto per alterare le proprietà degli stili. Si noti che è necessario posporre lo stile `stato` come ultimo parametro. `stato`, infatti, esegue entrambi gli stili `forma e dimensione` e `colore`. Quindi, è necessario che le modifiche eseguite a due sotto-stili vengano rese operative *prima* dell'esecuzione dello stile di alto livello.

4 Conclusione

In questo articolo si sono analizzati gli strumenti principali che *TikZ* mette a disposizione per ottenere alta personalizzazione e riproducibilità dei disegni. In particolare modo, si è focalizzata l'attenzione sul pacchetto `pgfkeys` considerando sia il sistema chiave-valore che gli stili, una delle peculiarità del pacchetto.

Dal punto di vista dell'utente normale del pacchetto, imparare ad impostare correttamente gli stili è importante per migliorare la produttività nel disegno. Chi invece desidera *programmare* con *TikZ*, quindi creare nuovi pacchetti o librerie, deve quasi assolutamente confrontarsi con l'uso del sistema chiave-valore.

Riferimenti bibliografici

- FIANDRINO, C. (2013). «Sa-TikZ: a library to draw switching architectures». *ArsTeXnica*, (15), pp. 4–10. URL <http://www.guit.sssup.it/arstexnica/>.
- PANTIERI, L. e GORDINI, T. (2014). *L'arte di disegnare con L^AT_EX*. URL http://www.lorenzopantieri.net/LaTeX_files/ArteTikZ.pdf.
- TANTAU, T. (2013). *The TikZ and PGF Packages*. URL <http://www.ctan.org/pkg/pgf>. Consultabile con: `texdoc tikz`
- WRIGHT, J. e FEUERÄSNGER, C. (2009). «Implementing key-value input: An introduction». *TUGboat*, **30** (1), pp. 110–122.

A Appendice

I colori `cerulean` e `royalblue` non sono presenti fra la collezione standard. Tramite l'opzione `dvipnames` di `xcolor` si può accedere ai colori `Cerulean` e `Royalblue`, perciò si riporta in questa appendice le definizioni di `cerulean` e `royalblue`:

```
\definecolor{royalblue}{cmyk}{1,0.50,0,0}
\definecolor{cerulean}{cmyk}{0.94,0.11,0,0}
```

▷ Claudio Fiandrino
University of Luxembourg
`claudio dot fiandrino at uni dot lu`

L^AT_EX in un'editrice universitaria: come e perché book non è adatta e una possibile alternativa funzionante

Gianluca Pignalberi

Sommario

L^AT_EX è un ottimo sistema di composizione perlopiù ignorato da molti editori. La sua flessibilità permetterebbe loro di risolvere brillantemente e con poche risorse molte ridondanze. L'articolo analizza puntualmente un caso d'uso reale, cerca di mostrare i vantaggi di L^AT_EX rispetto a un normale programma di composizione tipografica e discute le soluzioni adottate.

Abstract

L^AT_EX is an excellent typesetting system mostly ignored by many publishers. Its flexibility would allow them to brilliantly solve many redundancies with few resources. The paper thoroughly analyzes a real use case, tries to show the advantages of L^AT_EX over a usual typesetting program and discusses the adopted solutions.

1 Introduzione

Molte case editrici tradizionali ignorano l'esistenza di T_EX e dei suoi derivati. Un buon numero di quelle utilizzando T_EX ne riservano l'uso a pochi libri specifici preferendo usare programmi "più sicuri" quali QuarkXPress e Adobe InDesign per il grosso della produzione. Sebbene questi programmi semplifichino il lavoro sulle riviste e sui libri dall'aspetto molto complicato, i libri dall'aspetto "tradizionale" trarrebbero grandi vantaggi dalla composizione con L^AT_EX — documenti sorgente scritti su file di testo modificabili con un semplice editor di testo anche da remoto, file di stile adatti a ogni esigenza, combinabili tra loro e riprogrammabili alla bisogna, comandi standard non soggetti a modifiche tra una versione e l'altra del compositore, libertà e gratuità del programma, compattezza e altissima qualità dei *camera-ready*.

La classe *book* non è adatta a soddisfare anche le più semplici esigenze di una casa editrice, dunque possiamo dire che L^AT_EX *off the shelf* non è utilizzabile. C'è da chiedersi quale programma, pur sovraccarico di template, possa soddisfare queste esigenze non appena installato... Una casa editrice ha molte collane diverse, ognuna col proprio aspetto peculiare, e una quantità di informazioni da riportare in varie parti del libro. Possiamo demandare la soddisfazione di tali esigenze a una classe microscopica da completare tramite *plugin*. In questo modello, il file di classe contiene solo i

comandi e le variabili strettamente necessari e, soprattutto, comuni e carica dei file di configurazione realizzati appositamente e atti a definire completamente l'aspetto di ogni collana dell'editore, cioè tutte le funzioni che altrimenti troveremmo in una classe monolitica. Usare una classe microscopica e una serie di plugin comporta circoscrivere le aree di intervento e debugging. È anche vero che ogni plugin non avrà nel suo sorgente il codice della classe microscopica; ciò comporta un lieve risparmio di memoria, la cui quantità è data dalla dimensione del file della classe moltiplicato per il numero dei plugin che altrimenti conterrebbero quel codice.

Qual è il numero di plugin necessari a comporre una collana? Non credo ci sia una quantità universale; io mi sono limitato a due plugin per ogni collana: uno relativo alle funzioni tipo `\chapter` (di fatto il contenuto di un file `.cls` al netto del codice finito nella nuova classe) e uno per i corpi.

L'articolo, descrivendo la mia esperienza come compositore L^AT_EX e *layout designer*¹ per le Edizioni Santa Croce di Roma — dove i miei layout in L^AT_EX e i relativi libri trovano posto a fianco alla produzione InDesign —,² descrive il lavoro di programmazione fatto per ottenere una classe modulare e adatta a soddisfare le esigenze compositive di una decina (per ora) di collane.

2 L'inizio

La mia collaborazione con le Edizioni Santa Croce (EDUSC da ora in poi) iniziò nel 2012 quando l'allora direttore (Jerónimo Leal, nome famoso nella comunità T_EX italiana, specialmente per il suo coinvolgimento nella composizione di edizioni critiche) di una nascente collana chiese fortemente all'editore della sua università di permettermi di progettare la serie e di comporre i suoi libri, molti dei quali sarebbero stati, o avrebbero contenuto, edizioni critiche. L'uso di L^AT_EX fin dall'inizio, cioè dal manoscritto, sarebbe stato cruciale, come riassunto in [LEAL e PIGNALBERI \(2012\)](#).

1. Il *layout designer* è semplicemente il progettista grafico, cioè colui che progetta nei minimi dettagli l'aspetto di un libro: dimensioni della pagina, della gabbia, dei font di qualunque elemento e tutte le altre misure, il tipo di font e dunque il nero della pagina, l'aspetto delle testatine e dei piedini, dei sommari e degli indici e via elencando. Io sono più spesso *layout programmer*, cioè mi limito a replicare in L^AT_EX dei layout già progettati e realizzati in InDesign.

2. Non c'è bisogno di dire che l'editore e gli autori della Santa Croce apprezzano molto le produzioni in L^AT_EX grazie all'aspetto di qualità superiore.

Nell'unica riunione dello staff fatta prima di assegnarmi l'incarico, presenti il direttore (Andrea Zanni), la grafica storica della EDUSC (Liliana Agostinelli), il direttore di collana e io, mi furono dati un paio di vincoli — formato del libro e aspetto desiderato della copertina — e mi fu chiesto che idea avessi per l'aspetto. La descrissi per sommi capi e mi fu richiesta qualche pagina di prova. Nei giorni successivi realizzai alcuni prototipi tra i quali fu scelto quello da cui partimmo per la fase successiva: conversione del manoscritto e impaginazione della prima bozza. L^AT_EX era entrato nella vita attiva della casa editrice.

3 Prima fase, ovvero della scoperta dell'inadeguatezza di book

Avendo cominciato il lavoro con una sola collana di libri, ho ritenuto immediato modificare la classe `book` per adattarla al progetto: gabbia, filetti, forma e dimensioni dei titoli, font, profondità e forma dell'indice generale, testatine e piedini trovavano tutti la giusta collocazione in quella che era la vecchia classe `book`. In sostanza bisognava modificare alcune parti del codice della classe per riprogrammarne i comandi e aggiungere alcuni pacchetti per semplificare parte del lavoro.

Ma prima di riscrivere parte del codice necessario, ho potuto notare che `book` non prevede buona parte degli elementi che caratterizzano un libro propriamente detto: il frontespizio³ è minimale e manca dell'informazione "editore"; il retrofrontespizio⁴ (*copyright page* in inglese) non è previsto così come non lo è la serie di informazioni ivi contenute.⁵ La figura 1 mostra le parti citate.

Dunque ho riprogrammato pesantemente il comando `\maketitle` per prevedere la collocazione del nome dell'editore in calce al frontespizio e per comporre il retrofrontespizio e il pre-frontespizio (questa pagina contiene il nome della collana e il numero del libro nella collana). Il frontespizio e il suo retro sono elementi ricorrenti e il loro aspetto non varia all'interno della stessa collana, mentre cambiano le informazioni da essi mostrate. Durante la (ri)programmazione ho notato la mancanza di molti "slot" di informazione, informazioni altrimenti presenti in molti testi, specie manuali universitari: sottotitolo, ISBN, anno di pubblicazione e molti altri. I preesistenti `\title`, `\author` e `\date` forni-

ti da `book` non sono minimamente sufficienti. Per ogni informazione aggiuntiva, per esempio il sottotitolo, ho aggiunto due righe di codice simili alle seguenti:

```
\def\subtitle#1{\gdef\@subtitle{#1}}
      % Book subtitle
\def\@subtitle{\@latexerror{No%
\noexpand\subtitle given}\@ehc}
```

ed esattamente identiche a quelle scritte per i comandi preesistenti.

Per quanto riguarda i corpi e gli scartamenti,⁶ ho dovuto modificare `bk12.clo`, cioè uno dei file usati da `book` per stabilire, per ogni misura standard (`\normalsize`, `\small`, `\footnotesize` e così via), il corpo e lo scartamento corrispondenti. Sono state piccole modifiche che però hanno "stravolto" la filosofia "proporzionale" di L^AT_EX: non più interlinee crescenti in proporzione al corpo ma fisse per tener conto dell'altezza della gabbia (realizzata in pica, multiplo del punto — 1 pc = 12 pt — per cercare di avere tutte le pagine riempite uniformemente). Visto che lo scartamento di alcuni corpi è diverso da quello di alcuni altri, non abbiamo una composizione a griglia.⁷

Molte delle dimensioni, infine, sono rimaste quelle standard: rientro dell'inizio di un paragrafo e delle note a piè di pagina, rientri del testo citato e degli elenchi, filetti. Ho previsto anche l'introduzione di un *flag* che permetta di scegliere tra la stampa digitale e quella offset. Nel secondo caso il PDF deve stare su un foglio più grande del formato finale e deve avere i crocini di taglio. In base alla LPPL avrei dovuto cambiare i nomi di tutti i file modificati per poterli ridistribuire. Molti file hanno mantenuto il nome originale perché non avevamo ancora discusso della ridistribuzione con l'editore. Avevamo comunque del codice funzionante che ci ha permesso di ottenere il primo libro in L^AT_EX di EDUSC nell'esatta forma progettata. Dopo il normale iter lavorativo abbiamo ottenuto un *camera-ready*, cioè un PDF pronto per la stampa.

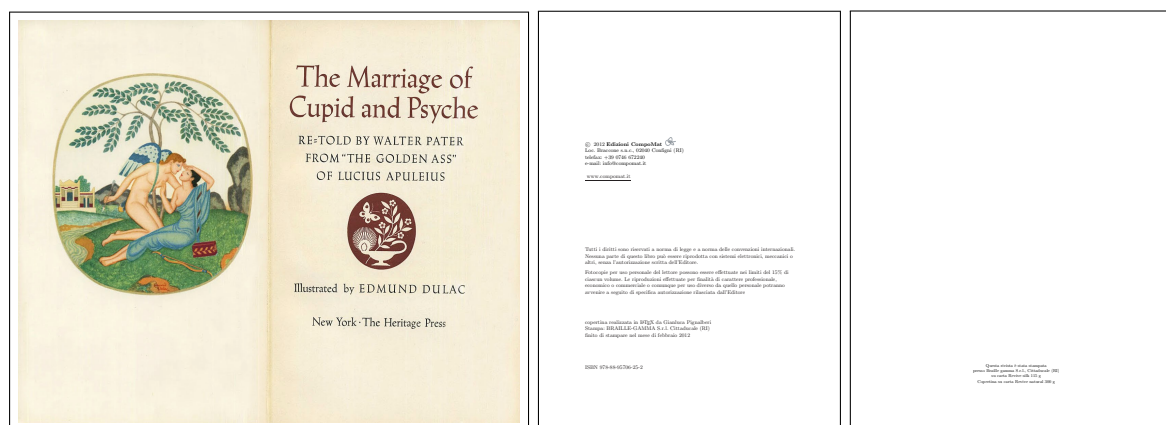
6. Il termine tipografico scartamento indica la distanza tra due linee di base. Tale termine è più comunemente conosciuto come interlinea (cfr., tra gli altri, FIORAVANTI (2002)), che in tipografia "fisica" indicava quella lamina di piombo interposta tra due righe di caratteri per distanziarle e in dattilografia indicava il numero di scatti di avanzamento del rullo della carta. Cfr. <http://www.treccani.it/vocabolario/interlinea/>

7. La *composizione a griglia* è quel tipo di composizione per cui le righe di sinistra di un documento (a due colonne) sono allineate orizzontalmente alle righe di destra. Possiamo immaginare di scrivere su un quaderno a righe: le righe della pagina pari saranno allineate a quelle della pagina dispari perché ogni riga si trova in una posizione fissa. Questo tipo di composizione si ottiene facendo sì che le dimensioni ammesse dei font abbiano la stessa interlinea. Nel caso il documento contenga elementi "extra-testo" (tabelle, figure...), bisognerà fare in modo che l'occupazione verticale di spazio di ogni elemento sia esattamente un multiplo dell'interlinea, eventualmente aggiungendo dello spazio prima e/o dopo. T_EX non è stato scritto avendo in mente la composizione a griglia.

3. In inglese *title page*, da non confondere col *frontispiece* che indica invece il controfrontespizio, cioè la pagina precedente il frontespizio che ora è bianca ma che anticamente soleva contenere un'illustrazione o altri fregi.

4. È comune confondere la pagina del copyright col colophon; quest'ultimo sarebbe il termine riservato alle informazioni sullo stampatore poste nell'ultima pagina del libro.

5. Ne deduco che `book` sia stata scritta il più genericamente possibile, indipendentemente dalle esigenze del mercato editoriale ma col chiaro intento di produrre un libro dall'aspetto compiuto.



(a) Controfrontespizio (frontispiece) e frontespizio (title page) di un libro. (b) Retrofrontespizio (copy-right page) del libro LEAL e PIGNALBERI (2012). (c) Colophon di un vecchio numero di Ar_sT_EX_nica.

FIGURA 1: Elementi di un libro la cui terminologia può ingenerare confusione.

4 Il vecchio e poco lungimirante metodo di lavoro

Neanche il tempo di “gloriarmi”: «Devi iniziare a immaginare un libro di un'altra collana preesistente. Vieni a Roma e ti dò un libro da rifare uguale.» è stata la richiesta dell'editore. Ingenuamente ho pensato: “Mi chiederà questa cosa una volta sola. Copio la classe, faccio le modifiche necessarie e via”. Invece ho sentito questa frase più di una volta e a un certo punto mi sono sorpreso a copiare il file della classe per modificarne inizialmente più o meno il 20%. Uno spreco! e anche un pericolo! Infatti è successo di scoprire servirmi un comando più facile da scrivere che da cercare già fatto; scrivevo il comando nella copia corrente della classe e poi lo dimenticavo là. Quando poi mi serviva per un altro lavoro, non mi ricordavo più in quale copia della classe l'avessi scritto. In pratica non stavo “industrializzando” il processo. In più, avendo trovato sovente dei libri di diverse collane con corpi identici e interlinee diverse, ho avuto la necessità di avere diverse versioni di, per esempio, `bk11.clo` e ho corso l'altissimo rischio di usare quella sbagliata. Puntualmente una volta è successo e ho composto il libro fino alla fase immediatamente precedente alla produzione del camera-ready quando l'editore si è accorto che lo scartamento era più piccolo di un punto rispetto a quello corretto.

Ho avuto l'ulteriore conferma dell'inadeguatezza del metodo di lavoro iniziale quando ho dovuto comporre un libro contenente un'opera collettiva, cioè quella con un curatore (*editor*) e diversi autori. Questa è normalmente una raccolta di atti di convegno o una monografia i cui capitoli siano scritti da diversi autori. Non avendo avuto prima alcun indizio dall'editore che avrei potuto impaginare tali opere, non mi ero preoccupato di programmare la classe in tal senso. La prima cosa da fare è stata ridefinire `\chapter` per fargli accettare il

nome dell'autore del capitolo e la sua affiliazione come ulteriori parametri obbligatori. All'inizio, sbagliando, non ho previsto di usare un flag per differenziare un'opera monoautore da una collettiva e di conseguenza non ho scritto un'altra versione di `\chapter` ma ho semplicemente modificato la versione esistente. Di fatto ho avviato una duplicazione delle classi per ogni collana: uno spreco immane.

Questo approccio *monolitico* infarcito di “ingenuità progettuali” si stava rivelando decisamente scomodo e antieconomico poiché necessitava di duplicare gli stessi file (la classe e la dimensione dei font) troppo frequentemente e di controllare che quelli in uso fossero effettivamente quelli corretti. Era giunto il momento di riprogettare tutto in maniera più razionale e meno soggetta agli errori umani, in maniera *modulare*.

5 La nuova classe e i suoi plugin, ovvero della possibile alternativa funzionante

L'idea portante della nuova progettazione è stata la stessa dei computer e, da qualche anno, degli amplificatori stereofonici: bisogna realizzare una *mother board* con gli appositi slot di espansione in cui inserire delle *daughter board* per espandere le funzionalità della macchina ospite. Nel caso in esame, bisogna realizzare una classe con il codice strettamente indispensabile e un meccanismo per selezionare automaticamente i plugin e i file di descrizione dei corpi da usare di volta in volta. Riguardo al primo elemento, il plugin, questo è nient'altro che un file di configurazione col nome della collana ospite del libro da comporre (potrebbe essere un file `.sty` e non cambierebbe la sostanza) da richiamare tramite `\input` in uno specifico punto della classe. Lo stesso criterio si applica al file dei corpi che però ora avrà il nome legato al file di configurazione, e quindi alla collana, e non

più un nome standard. Abbiamo già evidenziato quanto sia sconsigliabile usare i file .clo coi nomi standard: il caso in cui due collane diverse hanno i rispettivi corpi principali di dimensione identica e le interlinee di dimensioni diverse è frequente e non può essere gestito da un solo file relativo a quel corpo. Inoltre può essere difficile ricordare al volo quali corpi abbia una determinata collana. Una volta stabilita la collana, i comandi e i corpi a essa relativi sono immediatamente identificati senza errore.

La classe attuale conta 335 righe di codice ed è destinata a crescere man mano che aggiungeremo altre collane. Riporto di seguito solo alcuni stralci utili a capire quali siano le parti “innovative”.

```
% Information spanned by all book series
\def\subtitle#1{\gdef\@subtitle{#1}}
\def\@subtitle{\@latex@error{No \noexpand\subtitle
given}\@ehc}
\def\volume#1{\gdef\@volume{#1}}
\def\@volume{\@latex@error{No \noexpand\volume
given}\@ehc}
\def\nvolume#1{\gdef\@nvolume{#1}}
\def\@nvolume{\@latex@error{No \noexpand\nvolume
given}\@ehc}
\def\edition#1{\gdef\@edition{#1}}
\def\@edition{\@latex@error{No \noexpand\edition
given}\@ehc}
\def\isbn#1{\gdef\@isbn{#1}}
\def\@isbn{\@latex@error{No \noexpand\isbn
given}\@ehc}
\def\layout#1{\gdef\@layout{#1}}
\def\@layout{\@latex@error{No \noexpand\layout
given}\@ehc}
\def\typeset#1{\gdef\@typeset{#1}}
\def\@typeset{\@latex@error{No \noexpand\typeset
given}\@ehc}
% Some books (series) have them, some not
\def\coverttext#1{\gdef\@coverttext{#1}}
\def\@coverttext{\@latex@error{No
\noexpand\coverttext given}\@ehc}
\def\coveraut#1{\gdef\@coveraut{#1}}
\def\@coveraut{\@latex@error{No \noexpand\coveraut
given}\@ehc}
\def\imprimatur#1{\gdef\@imprimatur{#1}}
\def\@imprimatur{\@latex@error{No
\noexpand\imprimatur given}\@ehc}
\def\translator#1{\gdef\@translator{#1}}
\def\@translator{\@latex@error{No
\noexpand\translator given}\@ehc}
\def\origlang#1{\gdef\@origlang{#1}}
\def\@origlang{\@latex@error{No \noexpand\origlang
given}\@ehc}
\def\origtitle#1{\gdef\@origtitle{#1}}
\def\@origtitle{\@latex@error{No
\noexpand\origtitle given}\@ehc}
\def\address#1{\gdef\@address{#1}}
\def\@address{\@latex@error{No \noexpand\address
given}\@ehc}
```

Il codice precedente mostra tutti i comandi aggiuntivi agli originali `\title` e `\author` e contenenti le varie informazioni del libro da porre nel frontespizio e/o nel retrofrontespizio. Alcuni di essi non saranno usati da una o più collane. È probabile che alcuni comandi vengano ulteriormente suddivisi o accorpati.

```
% Put here ifs about EDUSC series
```

```
\newif\ifAFT %Atti della Facoltà di Teologia
\newif\ifAS %Apollinare Studi
\newif\ifMCEB %Markets, Culture and Ethics Books
\newif\ifMCEnb
  %Markets, Culture and Ethics Notebooks
\newif\ifPeC %Poetica e Cristianesimo
\newif\ifPF %Prospettive Filosofiche
\newif\ifQSC %Quaderni di Storia della Chiesa
\newif\ifSC %Subsidia Canonica
\newif\ifSISRI %Sisri
\newif\ifStTeo %Storia della Teologia
\newif\ifSuTeo %Sussidi di Teologia
```

Questa parte di codice contiene tutti i test atti a identificare la scelta relativa alla collana. Tale scelta, effettuata tra i parametri di `\documentclass`, indirizzerà il caricamento del plugin e dei corpi. Il seguente codice, invece, specifica le opzioni appena citate. Da qui in poi una serie di puntini starà a indicare che le righe mancanti differiscono dalla prima e dall’ultima per il solo nome dell’opzione e, quindi, del comando di verità/falsità.

```
% What series can be typeset with this class
\DeclareOption{AFT}{\AFTtrue}
.
.
.
\DeclareOption{SuTeo}{\SuTeottrue}
```

Il prossimo codice controlla la presenza o meno di altre opzioni del documento e, se è il caso, definisce altri comandi relativi alle informazioni da porre nel frontespizio o nel retrofrontespizio — in questo caso ricadiamo nel caso della composizione degli atti di una conferenza o di un’opera simile.

```
\newif\ifcropmarks %\cropmarkstrue
\DeclareOption{cropmarks}{\cropmarkstrue}
\newif\ifmultiauthor
\DeclareOption{multiauthor}{%
\def\editor#1{\gdef\@editor{#1}}
\def\@editor{\@latex@error{No \noexpand\editor
given}\@ehc}
\def\conffyear#1{\gdef\@conffyear{#1}}
\def\@conffyear{\@latex@error{No \noexpand\conffyear
given}\@ehc}
\multiauthortrue}

\ProcessOptions
\ifAFT\input{AFT.clo}\fi
.
.
.
\ifSuTeo\input{SuTeo.clo}\fi
```

Le righe di codice precedente mostrano la fase di caricamento del file dei corpi in base alla selezione scelta per la collana.

Le righe di codice seguenti sono relative alla strutturazione del libro. Rispetto a `book` c’è un paio di contatori in più. Torneremo sull’argomento nell’appendice A.4.

```
\newcounter {part}
\newcounter {subpart}
\newcounter {chapter}
\newcounter {subchapter}
\newcounter {section}[chapter]
```

```
\newcounter {subsection}[section]
\newcounter {subsubsection}[subsection]
\newcounter {paragraph}[subsubsection]
\newcounter {subparagraph}[paragraph]
```

Infine, le seguenti righe di codice contengono la parte di selezione del plugin. Effettuiamo tale operazione per ultima in modo da evitare alcune incompatibilità che cambiano l'aspetto del libro rispetto al voluto.

```
\ifAFT\input{AFT.cfg}\fi
.
.
.
\ifSuTeo\input{SuTeo.cfg}\fi
```

Ora che abbiamo visto parte della classe, è opportuno vedere, almeno in minima parte, qualche plugin. Il codice seguente mostra un paio di versioni del comando `\makechapterhead`.

```
% QSC.cfg
\renewcommand\@makechapterhead[1]{%
  \vspace*{8pc}%
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
      \if@mainmatter
        \large\scshape \chapapp\space \thechapter
        \par\nobreak
        \vskip 1pc
      \fi
    \fi
    \interlinepenalty\@M
    \large \scshape #1\par\nobreak
    \vskip 2pc}}
```

```
% MCEnb.cfg
% Alcune delle righe seguenti
% sono state spezzate
% per motivi di impaginazione
\renewcommand\@makechapterhead[2]{%
  \let\oldemph\emph
  \renewcommand{\emph}[1]
    {\textcolor{black!70}{#1}}
  \noindent\begin{tabular}
    {@{}b{9.5cm}@{}b{.5cm}@{}b{2.5cm}@{}}
  \large{\textbf{\textsf{\raggedright#1}}}&&
  \fontsize{84}{84}\selectfont
  \textsf{\textcolor{black!70}
    {\hfill\thechapter}}\\
  \hline
  \end{tabular}
  \vskip3em\noindent
  \textbf{\textsf{\textcolor{black!70}{#2}}}%
  \vskip5em
  \renewcommand\cname{#1}
  \renewcommand\autname{#2}
  \renewcommand{\leftmark}{#1}
  \renewcommand{\rightmark}{\booktitle}
  \let\emph\oldemph}
```

La differenza tra i due comandi è evidente: il primo ha una “forma” più classica e usuale, il secondo veicola un aspetto più ardito con filetti, spazi e corpi molto diversi per il titolo e il numero del capitolo e il corsivo (qui bisognerebbe opportunamente usare il termine “enfaticizzato”) soppiantato dal tondo grigio.

Ogni plugin di configurazione completa la classe base, di per sé incapace di impaginare alcunché poiché mancante di tantissimi comandi. Avrei potuto

scrivere una classe base da usare anche da sola, ma la mia intenzione era proprio far sì che fosse necessario usare un plugin (due, contando quello dei corpi). Dunque ogni plugin di configurazione conterrà il resto dei comandi necessari all'impaginazione di una collana.

Non ci dilunghiamo sui file dei corpi perché, oltre al nome dei file stessi, al loro interno cambia solo qualche dimensione rispetto ai file standard. Nel caso in cui due file `.clo` siano identici, possiamo evitare la copia fisica del file e ricorrere a un link.

È finalmente giunto il momento di mostrare i risultati del lavoro fin qui svolto. La figura 2 mostra le pagine iniziali di un capitolo di ogni collana fin qui realizzata.

L'utente medio di un qualunque word processor o di un programma visuale di composizione tipografica potrebbe chiedersi qual è il vantaggio di usare questo approccio “modulare” rispetto ai loro file di stile: la parte di codice comune a tutti gli stili risiede su un unico file che non deve essere replicato per ogni nuovo stile. Ciò comporta un risparmio — pur modesto — di memoria e un'economia di manutenzione: le modifiche fatte alla classe valgono per tutti i plugin e avere dei file sorgente più piccoli può aiutare a circoscrivere l'area di intervento in caso di modifiche o errori. Dunque l'approccio modulare è più economico rispetto all'approccio monolitico e un file di stile di Word o di InDesign sono monolitici, oltretutto modificabili solo avendo a disposizione i programmi citati.

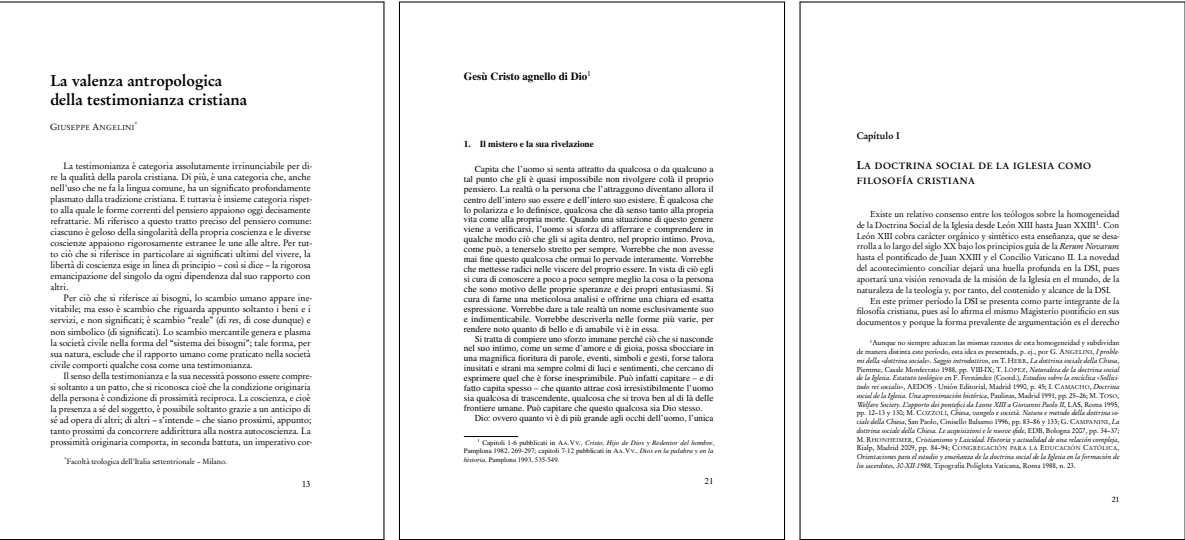
6 Sviluppi futuri

Massimiliano Dominici sviluppò per Aracne, col mio marginalissimo supporto,⁸ un lavoro simile a quello presentato qui ma più articolato: era infatti prevista la diffusione agli autori della classe per comporre da sé i libri da sottoporre all'editore. Tale classe non doveva, correttamente, prevedere la presenza di dati non di competenza degli autori (ISBN e altri dati di competenza dell'editore). Erano quindi previsti un flag e una serie di file aggiuntivi riservati proprio all'editore.

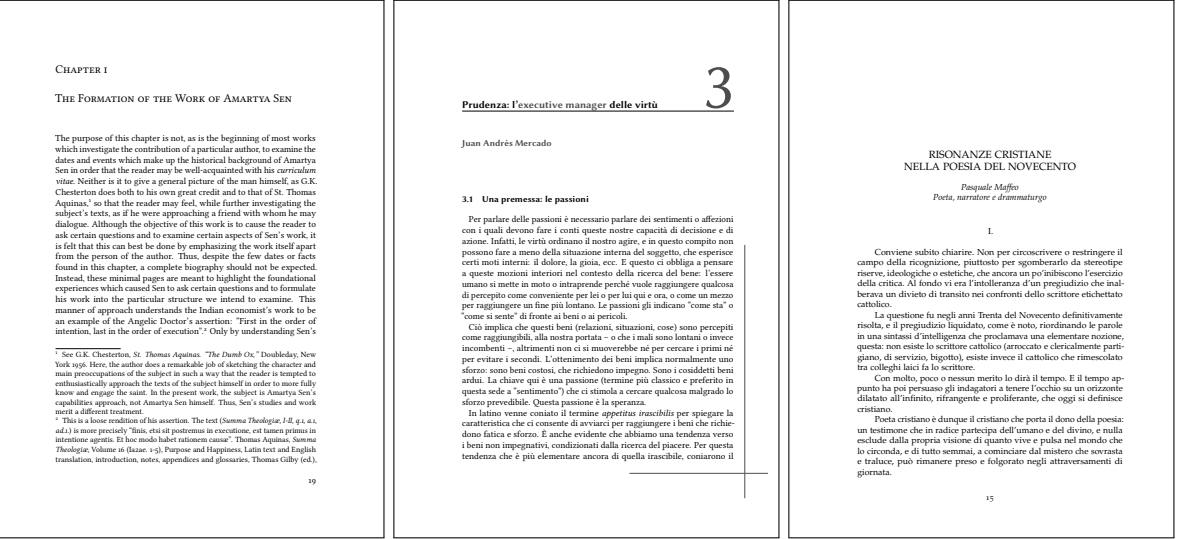
Finora un solo autore delle opere da me impaginate mi ha fornito un manoscritto L^AT_EX.⁹ Proprio a tale proposito non è previsto che la classe sia divulgata a breve. Che sia o no divulgata, implementerò lo stesso la tecnica di Max così da tenere le informazioni editoriali concettualmente separate dal lavoro degli autori.

8. Il mio supporto riguardò il calcolo automatico di un box che Max aveva dimenticato di implementare correttamente e la segnalazione del malfunzionamento in alcuni casi di un comando scritto *ad hoc*.

9. I manoscritti consegnati alla EDUSC sono perlopiù file Word. In un caso ho ricevuto un manoscritto scritto con Open/LibreOffice Writer e un'altra volta uno in L^AT_EX. Però un paio di autori mi hanno chiesto la classe così da poter preparare meglio, e in L^AT_EX, i loro manoscritti futuri.



(a) Atti della facoltà di teologia. (b) Apollinare Studi. (c) Markets, Culture and Ethics Books (vecchia impaginazione).



(d) Markets, Culture and Ethics Books (nuova impaginazione). (e) Markets, Culture and Ethics Notebooks. (f) Poetica e Cristianesimo.

FIGURA 2: Pagine iniziali di capitolo delle varie collane realizzate per EDUSC.

L'altro ovvio prosiegua del lavoro riguarderà la "traduzione" in $\text{\LaTeX}$ di tutte le collane della casa editrice, cioè la scrittura di tutti i file di configurazione necessari a comporre i libri di qualunque collana di EDUSC. Anche se la produzione è per circa due terzi affidata a InDesign, è evidente a tutti che uno dei vantaggi di $\text{\LaTeX}$ è la compattezza dei PDF generati.

L'ultima cosa da fare per completare un progetto dal respiro più ampio è la standardizzazione della conversione da $\text{\LaTeX}$ a $\text{\LaTeX}$ e $\text{\LaTeX}$ per realizzare gli ebook. L'unico esperimento effettuato

finora con $\text{\LaTeX}$ e Calibre ha ribadito le minori dimensioni del file prodotto da $\text{\LaTeX}$ rispetto all'equivalente InDesign. Se la produzione affidata a $\text{\LaTeX}$ continuerà, avremo modo di fare ulteriori comparazioni. In ogni caso sarà opportuna anche una serie di test con Pandoc.

Da tempo sto pensando di rivedere il criterio di assegnazione dei nomi ai comandi relativi ai corpi. Non più $\text{\small}$, $\text{\normal}$ e simili ma, per esempio, $\text{\corpotesto}$, $\text{\corponota}$, $\text{\corpotitolo}$... Purtroppo questo rende un po' più complicato il passaggio all'ebook perché biso-



FIGURA 2: (seguito) Pagine iniziali di capitolo delle varie collane realizzate per EDUSC.

gna modificare di conseguenza dei file specifici di $\TeX$ 4ht, segnatamente almeno `book.4ht`, contenuto nella directory `$\TeXMF/tex/generic/tex4ht`. Ulteriori riferimenti a $\TeX$ 4ht, di cui $\text{ht}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ fa parte, possono essere reperiti in GURARI (2004) e in GOOSSENS e RAHTZ (1999).

7 Conclusioni

Sebbene gli editori ignorino l'esistenza di $\TeX$ e dei suoi derivati, oppure ne ignorino le potenzialità, oppure ancora lo usino solo per comporre testi specialistici, l'uso pervasivo di tale programma in una casa editrice sarebbe molto importante: linguaggio standard dei formati¹⁰ poco soggetto alle

10. Tra questi abbiamo $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}_2\epsilon$ e i suoi *spinoff* e $\text{Con}\mathcal{T}\mathcal{E}\mathcal{X}\mathcal{T}$. $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 2.09 è ancora supportato e $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}_3$ ha un ottimo futuro. $\TeX$ vero e proprio è di fatto congelato e non più soggetto a modifiche del linguaggio.

modifiche “di marketing” e dunque compatibile virtualmente per sempre; documenti, classi e stili salvati su file di testo e dunque modificabili facilmente da remoto con un semplice editor di testo, senza bisogno del programma che li ha generati; un gran numero di pacchetti aggiuntivi realizzati dalla comunità e adatti alle esigenze più diverse; libertà e gratuità del tutto; stabilità dimostrata in oltre trent'anni di esistenza su tutte le piattaforme possibili; altissima qualità del testo composto.

Ho “importato” l'uso di $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ all'interno della EDUSC, facilitato dal fatto che è un programma che non richiede investimenti e che alcuni professori dell'Università della Santa Croce ne erano già grandi estimatori, programmandone la classe e alcuni plugin per replicare l'aspetto delle collane esistenti e ho progettato l'aspetto di nuove collane (mentre scrivo, devo ancora convertire molte delle

vecchie classi). Il successo dell'operazione ha fatto sì che un terzo della produzione annuale di EDUSC sia realizzato in L^AT_EX e il suo aspetto sia molto apprezzato da tutti. L'ultima caratteristica molto apprezzata della produzione in L^AT_EX è l'estrema compattezza dei PDF.

Ringraziamenti

Ringrazio Claudio Beccari che, avendo letto una prima bozza dell'articolo, mi ha corretto buona parte della terminologia tipografica dandomi le relative traduzioni inglesi e mi ha indicato refusi e migliorie. Inconsapevolmente mi ha indotto a inserire la figura 1 e a dilungarmi nelle spiegazioni sulla terminologia. Ringrazio anche i redattori di ArsTeXnica per le correzioni e le osservazioni indicatemi. È grazie a loro se l'articolo è migliore per forma e contenuto.

Riferimenti bibliografici

- FIORAVANTI, G. (2002). *Il nuovo manuale del grafico*. Zanichelli.
- GOOSSENS, M. e RAHTZ, S. (1999). *The L^AT_EX Web Companion*. Addison Wesley.
- GURARI, E. (2004). «T_EX4ht: Html Production». *TUGboat*, **25** (1).
- LEAL, J. e PIGNALBERI, G. (2012). *Edizioni critiche*. Edizioni CompoMat.
- WALLACE, D. F. (2006). *Infinite Jest*. Einaudi. La prima edizione italiana è stata pubblicata nel 2000 da Fandango.

A Vita di redazione¹¹

È banale dire che in ogni organizzazione, anche la più perfetta e roduta, ci possono essere degli intoppi e delle incomprensioni. Soprattutto può succedere di avere delle sovrapposizioni di compiti o di competenze vere e supposte. Quel che segue è un elenco minimo dei consigli di un non-impaginatore a un impaginatore che mi è capitato di ricevere. Tali consigli, a volte buoni, a volte meno, sono l'occasione per discutere alcuni argomenti trascurati nel corpo dell'articolo.

A.1 Vedove e orfane

In un ambiente di produzione quale si presume sia una casa editrice, l'ottimizzazione dei tempi e la non duplicazione dei compiti dovrebbe essere cruciale. Ho trovato invece un luogo in cui la dilatazione dei tempi sembra essere una delle priorità. . .

11. Attenzione: lettura non tecnica.

degli autori e il *throughput*¹² sia una grandezza sconosciuta.

L'eliminazione di vedove e orfane è l'ultimo compito da affrontare durante la composizione di un libro perché ogni modifica (correzione, cancellazione, aggiunta) è potenziale fonte di introduzione o eliminazione di righe di testo. Se a ogni "giro di bozze" si dovessero eliminare vedove e orfane, la lavorazione del libro subirebbe un incremento notevole dei tempi.¹³ Dunque, a meno di indicazioni contrarie dell'editore, lascio che queste "nefandezze tipografiche" sopravvivano a lungo. Puntualmente ogni autore non manca di segnarmi sulle bozze di "eliminare quella riga singola", di "non separare i titoli dai relativi testi", di "portare quelle due righe da sole nell'ultima pagina alla pagina precedente".

A.2 Errori gravi

Come sa ogni utente di L^AT_EX, anche alle prime armi, dimenticare una parentesi graffa chiusa può scatenare una serie di errori in cascata. L'errore che più comunemente commetto è dimenticare la graffa chiusa di un `\footnote`. Cioè, siccome i manoscritti sono in genere scritti con Word e la conversione restituisce un file L^AT_EX pieno di parentesi (e anche inutili comandi *ad hoc*, ahimè), durante la fase manuale di cancellazione e bilanciamento delle parentesi (eseguita solo se strettamente necessaria) mi capita di cancellarne una di troppo.

L'autore che riceve la bozza vede con terrore che una nota a piè di pagina è sparita, che parte del testo è finito in una nota, che un richiamo a una nota sparita si trova in un'altra nota. Sempre l'autore, paziente, scrive sulla bozza tutto quel che trova di errato e si spinge a riscrivere tutto il testo della nota saltata.¹⁴

Ignaro! Non sa che il sorgente L^AT_EX contiene tutto e, al modico costo di scrivere la parentesi mancante e ricompilare, il suo prezioso libro torna splendente e integro. Se lo sapesse, eviterebbe il comprensibile spavento.

A.3 Elementi incoerenti

Tutti gli utenti di L^AT_EX sanno quanto la coerenza dell'aspetto sia un discorso fondamentale per questo programma: è proprio per questo che si è

12. Nell'ambito dei sistemi operativi e relativi *scheduler*, questa grandezza indica il numero di task conclusi nell'unità di tempo; in altri ambiti indica la quantità di dati trasmessi nell'unità di tempo.

13. La dilatazione dei tempi è difficilmente stimabile pur sapendo l'entità delle modifiche: i testi con molte note a piè di pagina offrono un'impressionante serie di difficoltà all'eliminazione delle righe vedove e orfane.

14. La nota 24 di WALLACE (2006) è lunga quasi dieci pagine in corpo 8 (escluse le subnote in corpo 6; ignoro la dimensione in termini di cartelle). Molti autori segnalano la nota "persa" copiandone il contenuto sul PDF della bozza e la difficoltà di fare ciò è indipendente dalla lunghezza della nota. Una sparuta minoranza, però, preferisce ancora annotare le correzioni sulla bozza cartacea. Se tutte le note fossero lunghe come quella di Wallace. . .

passati dal paradigma dei comandi a quello delle etichette.¹⁵ Etichettare con `\chapter` alcune stringhe di testo assicura che l'aspetto di tutte quelle stringhe sia uniforme e che il solo significato all'interno della struttura del documento sia coerente, ricevendo pertanto l'adeguato trattamento anche durante la fase di creazione dell'indice generale.

La stragrande maggioranza degli autori ignora l'uso degli stili e, inconsapevolmente, il concetto di struttura. La cosa peggiore è che ignora anche, e non potrebbe essere altrimenti perché non è il suo compito conoscerla, la coerenza tipografica. Due esempi a tal riguardo.

A.3.1 I titoli

Anche i titoli sono coerenti nell'aspetto. Se scrivo `\section`, il testo che passo come argomento avrà un certo aspetto e una certa distanza dal testo precedente e successivo. Stessa cosa per i titoli degli altri livelli. Se il testo di una `\section` inizia col comando `\subsection`, mi aspetto che questo testo abbia una sembianza diversa da quella che avrebbe se il comando `\subsection` non ci fosse.

In un caso, il testo di un `\subsubsection`, normalmente rientrato,¹⁶ iniziava con un `\paragraph`. In base allo stile adottato, nessun `\paragraph` ha il titolo rientrato e c'è sempre dello spazio bianco sopra di esso. Osservazione dell'autore: *il testo dopo il titolo non è coerente perché c'è una riga vuota di troppo e il testo che segue non è rientrato*. Gli ho fatto educatamente notare che in casi simili, prima del titolo c'era dello spazio vuoto e il titolo era a bordo pagina, senza rientro ma che ero disposto a cambiare questa coerenza. Infatti sono stato obbligato a cambiare la coerenza in base al criterio dell'autore.

A.3.2 Gli elenchi

Molto spesso mi capitano dei manoscritti in cui gli elenchi non siano realizzati come elenchi ma come testo normale. A meno che il testo di ogni punto non sia troppo lungo (e capitano punti lunghi un paio di pagine), uso un ambiente `enumerate`, `itemize` o `description` per rendere al meglio l'elenco. È evidente che ogni elenco avrà esattamente lo stesso rientro a sinistra, in base alla definizione di base o a quella data da pacchetti come `enumitem`.

In un testo mi è capitato che un autore abbia espressamente voluto che un suo elenco, piuttosto corto, non venisse rientrato a differenza di tutti gli altri. La motivazione: *tutti i rientri che io non avevo messo sono stati messi per un'interpretazione "meccanica". Mentre per gli altri elenchi i rientri vanno bene e facilitano la lettura, per questo no. Eliminiamo il rientro*. Rientro eliminato per il solo elenco indicato.

A.4 Situazioni impreviste

Alcuni autori desiderano dividere determinate parti di un libro in sottoparti e queste in capitoli; altri autori decidono invece di dividere alcuni capitoli in sottocapitoli e poi in paragrafi (`\section`). La prima volta che vidi questa cosa ne rimasi contrariato perché non capivo questa struttura. Poi ho capito che invece è molto significativa, tant'è che il web riporta molti esempi di queste suddivisioni e di codice L $\text{\TeX}$ per realizzarle. Ho inserito in pianta stabile questi elementi nella classe, come già discusso nel paragrafo 5.

▷ Gianluca Pignalberi
g dot pignalberi at alice dot
it

15. Sebbene le etichette di L $\text{\TeX}$ siano dei veri e propri comandi, la nomenclatura usata e la realizzazione dei comandi svincolano realmente l'autore dall'obbligo di occuparsi degli aspetti tipografici del documento, mutando così il paradigma di T $\text{\TeX}$.

16. I rientri ci sono in ogni caso visto l'uso di `indentfirst`.

Questa rivista è stata stampata
presso Centro Stampa e Riproduzione S.r.l.,
Via di Pietralata, 157 – 00158 Roma,
su carta vellum white 80 g.
Copertina: vellum white 260 g.



Dodicesimo convegno nazionale su T_EX, L^AT_EX e tipografia digitale

Trento, 17 ottobre 2015

Call for Papers

Sabato 17 ottobre, a Trento, si terrà il dodicesimo Convegno annuale su T_EX, L^AT_EX e tipografia digitale organizzato dal Gruppo Utilizzatori Italiani di T_EX. Il Convegno sarà un momento di ritrovo e di confronto per la comunità L^AT_EX italiana, tramite una serie di interventi atti sia a contribuire all'arricchimento sia a supportarne lo sviluppo.

Maggiori informazioni sul Convegno e sulle modalità di presentazione degli interventi saranno disponibili all'indirizzo:

<http://www.guitex.org/guitmeeting/2015/>

ArsT_EXnica

Rivista italiana di T_EX e L^AT_EX

Numero 19, Aprile 2015

- 5 Editoriale
Claudio Beccari
- 6 Come si può influenzare la posizione degli ambienti galleggianti come `figure` e `table` in L^AT_EX?
Frank Mittelbach
- 14 Una figura larga sotto un testo a due colonne
Barbara Beeton
- 15 `\vadjust`, questo sconosciuto
Claudio Beccari
- 19 Comporre liste: un viaggio
nella grammatica sconosciuta
James R. Hunt
- 27 Personalizzazione e riproducibilità nel disegno programmato: introduzione al
pacchetto `pgfkeys`
Claudio Fiandrino
- 33 L^AT_EX in un'editrice universitaria: come e perché `book` non è adatta e una
possibile alternativa funzionante
Gianluca Pignalberi

