

# Debugging L<sup>A</sup>T<sub>E</sub>X files

## *Illegitimi non carborundum\**

Barbara Beeton

### Sommario

Almeno una volta nella propria attività di utente L<sup>A</sup>T<sub>E</sub>X ci si è trovati di fronte a problemi spinosi quando la compilazione si interrompe per qualche oscuro motivo. Si sa abbastanza bene come comportarsi in presenza di problemi semplici, ma ci sono alcune situazioni nelle quali i metodi tradizionali non bastano.

Questo articolo descriverà le strategie e le tattiche per affrontare i problemi emersi durante un lungo periodo della mia attività, come membro del gruppo di sostegno tecnico presso l’American Mathematical Society (AMS), per gestire i problemi degli autori e del personale tecnico. Si incontreranno sia casi semplici sia casi insoliti, con indicazioni utili per tutti al fine di evitare problemi durante il proprio lavoro.

### Abstract

Every L<sup>A</sup>T<sub>E</sub>X user has, at least once in her career, been faced with a thorny problem when compilation shuts down for some obscure reason. How to deal with simple problems is reasonably well known, but there are situations when the time-honored methods fall short.

This article will present strategies and tactics for dealing with the many types of problems that have arisen during long experience as a member of the AMS technical support staff, handling questions from authors and the editorial staff. Both common and uncommon glitches will be visited,

\*Claudio Beccari ha tradotto BEETON (2017) col permesso dell’autore. Il titolo non è stato tradotto, rimanendo quello originale. La frase in simil-latino è una frase fatta del mondo anglosassone; significa qualcosa come *Non farti demolire dalle avversità*; oppure *È brutto ma tocca farlo lo stesso*. Non ho modificato l’Abstract lasciando l’originale. Inoltre devo segnalare che i *bug* del software inglesi corrispondono per similitudine fonetica ai *bachi* italiani; il verbo inglese *to debug* con i suoi derivati non si può tradurre in italiano con una sola parola più significativa di *correggere*; per non appesantire la traduzione ho lasciato quasi sempre la parola inglese. Come traduttore non mi sono limitato a creare la versione italiana, ma a beneficio del lettore italiano mi sono permesso di aggiungere molte mie note che estendono un poco quanto scritto dall’autore di questo articolo; le mie note sono tutte marcate alla fine con (N.d.T.). In realtà in italiano esiste il verbo *spulciare* che significa, letteralmente, “mondare dalle pulci”. Purtroppo il suo utilizzo nell’accezione di *to debug* è impossibile perché nell’uso corrente significa “dare un’occhiata rapida e distratta”. (N.d.T. e d.R.)

with a bias toward avoiding problems in one’s own work—something for everyone.

### 1 Retrosceca

Nel 2016 l’AMS ha totalizzato la pubblicazione di circa 60 000 pagine di libri e riviste, la maggior parte delle quali da file L<sup>A</sup>T<sub>E</sub>X composti e inviati dagli autori. Il fatto che un articolo per una rivista venga accettato è basato sul valore scientifico del contenuto, valutato da un comitato editoriale e da scienziati competenti che si basano su un documento elettronico, non necessariamente composto con un qualche metodo di videoscrittura, ma talvolta anche scritto a mano. Non viene assegnato nessun valore alla presentazione, ma solo al contenuto. I libri vengono acquisiti sotto contratto dal gruppo editoriale; i membri del gruppo e gli autori dei testi conoscono L<sup>A</sup>T<sub>E</sub>X ma non sono necessariamente T<sub>E</sub>Xnici esperti. Quello che ci arriva da pubblicare è ciò che dobbiamo affrontare.

Supponiamo che un lavoro accettato sia preparato in L<sup>A</sup>T<sub>E</sub>X (se non lo fosse, un tecnico competente provvederebbe a renderlo tale per l’uso successivo); la qualità dei testi inviati è molto varia e offre una ricca serie di opportunità per sperimentare (e migliorare) le proprie capacità di *debugger*.

La produzione si svolge su una rete di sistemi Linux. La libreria disponibile è divisa in tre parti: T<sub>E</sub>XLive, che viene aggiornato al massimo una volta all’anno; una versione locale di font e di file “pubblici” (che talvolta includono versioni aggiornate di ciò che farà parte della successiva collezione di T<sub>E</sub>XLive); macro, font ed altri strumenti che sono di proprietà esclusiva dell’AMS. Tutto è archiviato con Subversions, con gli archivi dei libri e degli articoli che risalgono fino a una ventina di anni addietro. Le versioni di (L<sup>A</sup>)T<sub>E</sub>X e di tutti i pacchetti usati sono registrati insieme ai main file per ciascun lavoro pubblicato usando il pacchetto *snapshot*, così che se si manifestasse la necessità di rielaborare il documento si potrebbe ricreare l’ambiente di composizione originale. Questo insieme di cose fornisce la stabilità necessaria per produrre continuamente nuove pubblicazioni restando in grado di gestire le riproduzioni, le revisioni e le conversioni delle pubblicazioni esistenti in altri formati, compresi gli ebook.

Come descritto sopra, questo metodo di lavoro è efficace e affidabile una volta che i file che rappresentano il manoscritto sono pronti per es-

sere inviati allo stampatore. Ma prima di questo momento può succedere ogni genere di cose. Un principio guida prevale su ogni altro: se qualcosa va storto, deve essere possibile recuperare prontamente e con assoluta affidabilità un punto stabile da cui riprendere il lavoro.

## 2 Preparazione — pianificare in anticipo

Esistono delle convenzioni che, seguite con attenzione, alla lunga possono rendere la vita più facile. Per prima cosa bisogna scegliere buoni strumenti acquisendone la completa padronanza.

Lo strumento di gran lunga più importante è un editor o un IDE.<sup>1</sup> L'autore di questo articolo usa **emacs** ma sono disponibili altre scelte, alcune utili per un solo utente su una singola macchina; altre predisposte per il lavoro cooperativo in rete; altre scelte sono una via di mezzo fra questi due tipi. Una lista di tali strumenti si trova in una risposta data nel sito **tex.stackexchange** (di qui in avanti denominato **tex.sx** (**STACKEXCHANGE**, 2017)).<sup>2,3</sup>

L'autore preferisce elaborare i file usando la linea di comando. Questo le permette di correggere interattivamente semplici errori (come comandi scritti in modo errato) evitando così ritardi e l'eventualità che si manifestino sequele di errori irrilevanti causati dal comando errato. (Ma non bisogna dimenticare di correggere anche il file sorgente prima della compilazione successiva).<sup>4</sup>

Tra le funzionalità più utili per il debugging ci sono le seguenti:

- una buona funzionalità nella ricerca di stringhe;
- il controllo dell'appaiamento corretto sia dei vari tipi di parentesi e altri delimitatori, sia delle coppie `\begin/\end` relative allo stesso ambiente;
- la possibilità di spostarsi direttamente ad una determinata riga di codice;

1. IDE: Integrated Development Environment; ambiente integrato per lo sviluppo del software. Nelle guide del  $\text{\LaTeX}$  è spesso denominato all'inglese come *LaTeX friendly editor* o come *shell-editor*. (N.d.T.)

2.  $\text{\LaTeX}$  Editors/IDEs, <http://tex.stackexchange.com/q/339>

3. L'indirizzo web <http://tex.stackexchange.com/q/339> punta ad un filone di discussione un po' datato; esso contiene informazioni interessanti, ma, per esempio, alcuni editor non esistono più o le pagine web loro collegate non funzionano più. Le caratteristiche dei vari editor non sono tutte ugualmente aggiornate; tuttavia si tratta di un ottimo elenco. Nelle guide del  $\text{\LaTeX}$ , sono indicati alcuni IDE gratuiti che hanno molte delle caratteristiche indicate in questo articolo. (N.d.T.)

4. È anche possibile usare una *shell editor* adeguatamente configurato (vedi più avanti) per seguire una tecnica simile: al verificarsi di un errore ortografico nel nome di un comando, prima si corregge il file sorgente, così che sia già pronto per la compilazione successiva, poi si interagisce con la console dell'IDE per inserire il comando corretto. (N.d.T.)

- la capacità di contare il numero di istanze dei delimitatori di vario genere.

Un altro fattore importante è la disposizione e il relativo metodo di accesso alle cartelle e ai file. È raccomandabile *evitare gli spazi nei nomi dei file*; infatti non tutti i sistemi operativi gestiscono tali spazi correttamente (alcuni non li gestiscono affatto). Inoltre alcuni sistemi operativi distinguono le lettere maiuscole dalle minuscole — per evitare problemi *si usino sempre solo le lettere minuscole nei nomi dei file*; le cifre e i trattini non danno fastidio nei nomi dei file, ma è bene evitare punti in eccesso (per esempio nomi di file che prima del punto dell'estensione contengono altri punti) e i caratteri che hanno un significato speciale per  $\text{\TeX}$ , per esempio il tratto ribassato `'_'`; per essere più chiari, un file dal nome **articolo\_del\_31.05.2016.tex** potrebbe dare molto filo da torcere, mentre se il nome fosse **articolo-del-31-05-2016.tex** non ci sarebbero intoppi di nessun genere.

Se si mantengono i file entro dimensioni ragionevoli alla lunga questo torna molto utile. Per un lavoro importante come un libro, si metta ciascun capitolo in un file a se stante, controllandone la compilazione mediante un main file o un “driver”. Questo permette di lavorare su un capitolo alla volta servendosi utilmente della funzionalità di `\includeonly`. Se si devono introdurre figure, disegni, tabelle di grandi dimensioni, metterli in file distinti importati con `\input` rende possibile escluderli con un semplice `%` per commentarne la direttiva di immissione (inserire questi grossi oggetti con `\input` di file separati rende anche facile, all'occorrenza, l'operazione di inserirli in posti diversi del documento).

Infine, quando si preparano questi file da immettere, è solitamente una buona idea quella di terminarli con un comando `\endinput` in una riga a se stante preceduta da una riga vuota; questo elimina i problemi che nascono (in modo non voluto) trasmettendo un file da un sistema a un altro; non mettere *mai* `\end{document}` se non solo alla fine del main file.

Un ultimo suggerimento: si impari subito dove trovare il file **log**, prima che sia necessario esaminare uno. Alcuni IDE nascondono questo file agli occhi dell'utente; se la compilazione va storta e non si può controllare che cosa sta succedendo leggendo il file **log**, si prospettano tempi molto difficili per scoprire che cosa bisogna correggere.

E ancora:

**Non si aggiorni il sistema  $\text{\TeX}$  mentre si sta lavorando un documento importante.**

Nuove versioni dei pacchetti potrebbero avere funzionalità nuove e incompatibili con quelle delle precedenti versioni, e i vecchi pacchetti non sarebbero più usabili. Naturalmente se l'hardware decide di “defungere” a quel punto, quanto detto

sopra non è una regola da seguire. Ma certamente è stato fatto un backup completo, non è vero?

### 3 Isolate e proteggete il vostro collaudo

**Bisogna usare solo copie dei file da collaudare.**

Se l'errore che si sta cercando di correggere non è semplice come un banale refuso, bisogna proteggersi da possibili disastri creando un ambiente di collaudo speciale. Come minimo si esegua un backup dei file; può andare bene anche un file zip dove si sia salvata l'intera cartella di lavoro con le sue sottocartelle, e che sia riposto in un luogo sicuro, magari su un supporto esterno. Si conosce bene la situazione corrente e all'occorrenza bisogna essere capaci di ripristinarla rapidamente e in tutta sicurezza.

**Per nessun motivo al mondo si modifichi l'unica copia di qualunque file.**

Ancor meglio, se si ha spazio nel disco, si crei un'apposita area di collaudo, identica sotto ogni aspetto importante a quella *vera* di lavoro; si esegua ogni sperimentazione in questa area di collaudo.

Se il lavoro consiste di diversi file, si cominci a copiare *solo* il main file — quello che legge tutti gli altri — nell'area di collaudo; questa sarà la “cavia” su cui lavorare.

Si creino e si usino dei link simbolici per accedere agli altri file. Per i sistemi Linux basta eseguire il seguente comando

```
ln -s <cartella>
```

aggiungendo quella <cartella> al PATH.<sup>5</sup>

Si elabori il documento in modo interattivo. In questo modo errori semplici possono venire corretti al volo prima che producano pletore di messaggi d'errore privi di senso. (Bisogna ricordare sempre di eseguire sia l'immissione interattiva della correzione, sia la correzione nel file che contiene l'errore.) E, se si manifesta un errore non correggibile al volo (come un ambiente ignoto o un ambiente non terminato), la compilazione può essere fermata subito e si può correggere l'errore prima di ricominciare.

Eseguire una compilazione in “nonstop mode” (il modo generalmente predefinito quando si lancia una compilazione attraverso un IDE) vuol dire provocare il verificarsi di una pletora di errori registrati nel file log (fino ad un massimo di 100, dopo di che la compilazione si arresta da sola, ma in modo “traumatico”, quindi talvolta con un file log non completo), ma un singolo errore, che non sia un banale errore ortografico nel nome di un comando, può provocare in cascata una moltitudine di errori, che non si verificherebbero se non

5. L'operazione è identica per Mac OSX; con sintassi leggermente diversa è possibile anche sulle piattaforme Windows recenti. (N.d.T.)

si fosse verificato il primo. Lavorare in “errorstop mode” permette invece di fermare subito la compilazione e interagire con la console immettendo uno dei vari comandi che la console accetta: X: fermare correttamente la compilazione; I: immettere una correzione; S: proseguire senza arrestarsi, se non per gli errori gravissimi; Q: proseguire in silenzio; H: chiedere delucidazioni; eccetera.

Più avanti se ne dirà di più nella sezione 8 “Divide et impera”.<sup>6</sup>

### 4 Alcuni strumenti per la diagnosi interattiva

Sono disponibili alcuni comandi diagnostici per inviare messaggi sia al terminale/console sia al file log.

`\message{...}` scrive il messaggio sia nel file log sia sullo schermo; può essere usato per segnalare quando la compilazione ha raggiunto determinati punti chiave. Per esempio,<sup>7</sup>

```
\message{Ultimo paragrafo,
  pagina \number\thepage}^^J
produce nel file log un messaggio del tipo:
  Ultimo paragrafo, pagina 904
```

`\show` scrive il significato corrente di un comando; l'elaborazione viene sospesa per consentire ulteriore interazione scrivendo nella console il codice I o i, seguito da ulteriori comandi diagnostici. Esempio:<sup>8</sup>

```
\show\LaTeX
produce:
  > \LaTeX= macro:
  -> \protect\LaTeX_{\_}.
Invece:
\show\protect
produce:
  > \protect=\relax.
```

Per inserire ulteriori comandi diagnostici bisogna fare come indicato sopra e mandare in esecuzione il comando premendo il tasto **Enter** o **Invio**; premendo nuovamente questo tasto senza avere inserito nulla nella console fa ripartire la compilazione.

`\showthe` riporta invece il *valore* di un comando che identifichi un registro numerico, dimen-

6. *Separa e governa*, noto precetto dell'impero romano. (N.d.T.)

7. Per la verità `\message` non è interattivo in senso stretto; scrive qualcosa sulla console e nel file log, ma non attende nessun intervento da parte dell'utente. Invece `\typeout` permette una vera interattività; si veda (LAMPOR, 1994). (N.d.T.)

8. Il punto finale ha un valore diagnostico importante; nel primo esempio lo spazio che separa la stringa `\LaTeX` dal punto segnala che quello spazio è parte integrante del nome della macro. (N.d.T.)

sionale, o altro;<sup>9</sup> agisce esattamente come il comando `\the`. Per il resto si comporta come `\show`.

```
\showthe\hfuzz
produce:
  > 1.0pt.
```

Sono disponibili un certo numero di comandi di tracciamento per mostrare i dettagli del flusso di lavoro. (Attenzione: questi comandi di tracciamento possono produrre troppa informazione rispetto a quella che si è disposti a leggere; quindi è bene essere selettivi.) Il risultato del tracciamento è inviato solo al file `log`, a meno che non sia richiesto diversamente. Questi sono i comandi che l'autore usa più frequentemente.

`\tracingoutput` può essere impostato al valore 1 per riportare in forma simbolica il contenuto delle scatole che vengono inviate al file di uscita.

`\tracingcommands` e `\tracingmacros` posti al valore 2 riportano i dettagli completi dell'elaborazione eseguita da  $\text{\LaTeX}$ .

`\errorcontextlines = 200` imposta il massimo numero di righe associate ad un singolo messaggio d'errore; il valore preimpostato è 5 ma spesso mostra troppe poche righe per comprendere l'intera operazione.

`\tracingonline` invia il rapporto di tracciamento sia allo schermo sia al file `log`.

I dettagli relativi a questi comandi (e molti altri comandi `\tracing...` della stessa famiglia) si possono trovare nel *TEXbook* (KNUTH, 1994) o in *TEX by topic*<sup>10</sup> (ELJHOUT, 1991).<sup>11</sup>

## 5 Il file log è amico tuo

Il file `log` riporta la registrazione di ogni azione eseguita — quali file e quali font vengono letti, le assegnazioni di scatole e di contatori, e via di questo passo. Più importante, dal punto di vista del debugging, gli errori sono riportati, spesso con un dettaglio faticoso da leggere, ma accompagnati dal numero della riga del file sorgente nella quale il compilatore crede che si trovino.

9. Al pari di `\the`, `\showthe` accetta come argomento anche una delle espressioni `...\expr` quali sono definite dalla sintassi estesa dei motori di composizione; per esempio `\showthe\dimexpr\topmargin+1in` mostra il valore vero del margine superiore. In questo testo il valore conservato nel registro dimensionale `\topmargin` vale `-52.36449pt` (negativo) mentre il valore vero vale `19.9055pt`  $\approx 7\text{ mm}$ , come il lettore può verificare misurando direttamente sullo stampato. (N.d.T.)

10. Con  $\text{\TeX}$ Live questo testo si può leggere anche con il comando `texdoc texbytopic`.

11. Il pacchetto `trace` offre due altri comandi, `\traceon` e `\traceoff` che eseguono un tracciamento abbastanza completo, omettendo però la maggior parte delle centinaia di righe che ogni cambiamento di font comporta e che di solito danno poche informazioni utili. (N.d.T.)

Ogni volta si controlli nel file `log` il messaggio d'errore, per esempio:

```
! Undefined control sequence.
1.457 \fobx
      {%
```

Gli avvisi (warning) sono segnalati anche loro ma senza l'indicazione della riga:

```
LaTeX Warning; There were undefined references.
```

L'interpretazione di queste informazioni potrebbe risultare una sfida per l'utente, ma questa informazione è la chiave per indirizzare la ricerca. Se il sistema che si sta usando nasconde il file `log` agli occhi dell'utente, si domandi come trovarlo. E non si cancelli mai un file `log` senza prima averlo esaminato.<sup>12</sup>

Non si deve assumere che il numero indicato per la riga dove si è verificato l'errore sia esatto. Per esempio il campo di azione di un tratto di matematica in linea col testo (cioè che è contenuto fra un segno `$` in apertura e un altro segno `$` in chiusura) non può contenere righe vuote; perciò la mancanza di un segno `$` può essere segnalata in corrispondenza di una riga vuota; può darsi che manchi il segno di chiusura oppure può darsi che si sia immessa una riga vuota all'interno del testo matematico.<sup>13</sup>

L'altro errore associato alla mancanza di un dollaro viene indicato con

```
! Missing $ inserted.
```

quando manca il “`$`” di chiusura oppure quando un simbolo matematico è fuori da un ambiente matematico.

Un errore all'interno di una figura, di una tabella, di una equazione di diverse righe, viene di solito segnalato con un numero di riga che corrisponde alla chiusura dell'ambiente, invece che nella riga dove è presente l'errore; ma anche in questi casi la ricerca dell'errore non è difficile poiché di solito lo “scopo” è relativamente breve.

12. Di solito non è difficile trovare dove sia il file `log` perché si trova sempre nella cartella dove risiede anche il main file del documento. Usando quell'applicativo dai nomi più disparati per esaminare il contenuto di cartelle e i file (per esempio, `Explorer.exe` su Windows; `Navigator` su alcune macchine Linux; `Finder` sulle macchine Mac) basta controllare il contenuto della cartella del main file. Si faccia attenzione al fatto che diversi sistemi operativi nascondono l'estensione dei file, se non di tutti, almeno di quelli per i quali esiste un preciso programma per agirvi sopra. Si configuri il programma di esplorazione del disco in modo che mostri sempre l'estensione di tutti i file; con il sistema  $\text{\TeX}$  è molto importante disporre di questa informazione. (N.d.T.) Sulle macchine `*n*x` si può usare anche il comando `find` da terminale. (N.d.R.)

13. È per questo che con  $\text{\LaTeX}$  sarebbe opportuno delimitare la matematica in linea con i delimitatori `\(` e `\)` invece di usare i dollari; l'informazione dell'errore risulta molto più precisa. (N.d.T.)



Un altro caso in cui l'errore viene segnalato lontano dal punto in cui è presente l'errore si manifesta quando un gruppo non è correttamente delimitato, cioè quando si trovano una graffa aperta “{” senza la graffa di chiusura, oppure `\bgroup`, `\begingroup`, `\begin{ambiente}` scompagnati, mancanti, cioè, del corrispondente comando di chiusura. Nel caso di un ambiente con i delimitatori scompagnati il messaggio d'errore potrebbe assumere un aspetto del genere:

```
! LaTeX error: \begin{ambiente}
on input line nnn
ended by \end{ambiente}
```

Questo verrà segnalato appena si incontra la chiusura non corrispondente all'apertura, ma il numero della riga del comando di apertura dovrebbe essere corretto.

Invece un comando di gruppo scompagnato non verrà segnalato fino alla fine del file in cui l'errore si trova, al limite alla fine del processo di composizione. Alla fine di questo processo il messaggio d'errore si presenta su diverse righe:

```
(end occurred inside a group at level m)
### semi simple group (level m)
entered at line nnn (x)
### bottom level
```

Qui  $m$  identifica quanti gruppi aperti restano tali al momento di terminare la compilazione;  $x$  indica qual è l'elemento scompagnato: `\begingroup`, oppure “{” o `\bgroup`. Anche qui il numero della riga  $nnn$  dovrebbe essere corretto ma non è indicato il punto in cui si è ommesso di chiudere il gruppo.

Altri messaggi d'errore sono riportati nella documentazione di molti pacchetti. La maggior parte di questi messaggi include l'indicazione del numero della riga, e in generale l'indicazione della riga incriminata è abbastanza buono. Spesso questo è sufficiente per localizzare l'errore così che possa venire corretto senza dover procedere con metodi più complessi di ricerca. Una volta trovato e corretto l'errore, se ne verifica l'effettiva correzione, poi la si riporta nel *vero* file di lavoro, e si continua con il lavoro principale, essendo lecito dimenticarsi delle copie fatte e dell'ambiente di collaudo, perché questi hanno svolto il loro compito; l'albero di collaudo può essere mantenuto, ma solo come struttura perché vi si potranno riversare altri file da collaudare con la stessa metodologia.

Ma, ci si potrebbe domandare, quando il lavoro consiste di diversi file come fa uno ad essere sicuro di quale sia il file il cui numero di riga è stato indicato come sede dell'errore? Si veda il prossimo paragrafo.

La lezione importante a questo punto è questa

**Non si cancelli il file log finché non se ne sia stata estratta ogni possibile anche minima informazione.**

È stato suggerito all'autore di conservare il file `log` sotto un altro nome anche più a lungo perché potrebbe essere utile; infatti potrebbe essere necessario eseguire il confronto di due successivi file `log` quando si vuol capire che cosa è cambiato fra due compilazioni successive.

## 6 E pluribus unum — ma quale?

Quando si ha a che fare con diversi<sup>14</sup> file, quale bisogna esaminare?

Si supponga che l'errore sia stato segnalato per un file di testo, non per un pacchetto.

Quando il file `log` riporta un numero di riga, la prima reazione è quella di guardare il main file. Ma se quel file è lungo solo 95 righe e il numero riportato è 2345, il conto non torna.<sup>15</sup>

Si faccia allora una copia del file `log` e la si lavori all'indietro partendo dal punto in cui è segnalato l'errore sotto esame. Se alcune pagine sono state prodotte e inviate al file di uscita, il numero della pagina (mostrato fra parentesi quadre: per esempio, [17]) può indirizzare verso un capitolo, che idealmente dovrebbe essere in un file a se stante. Altrimenti bisogna eliminare da questa copia tutto quanto non serve per lo scopo.

I messaggi relativi alle scatole troppo piene (o troppo vuote) non servono e le rispettive righe possono essere eliminate. Le coppie di parentesi tonde ben appaiate normalmente includono il nome del file che viene aperto in lettura e informazioni varie sulla elaborazione del suo contenuto. Si cerchino allora gruppi completi fra parentesi tonde, come per esempio

```
(C:/tech-support/debug/preface.tex
Preface
[1] [2]
)
```

e si cancelli l'intero gruppo. Quello che rimarrà alla fine è una parentesi tonda aperta seguita dal nome di un file — il nome del file durante la cui elaborazione si è manifestato l'errore. Il numero segnalato dovrebbe riferirsi a quel file.

14. *E pluribus unum* è un motto degli USA; il latino rende in modo conciso il concetto di “Un solo popolo da diversi popoli”. (*N.d.T.*)

15. L'autore di questo articolo lavora anche con l'interprete `tex` nella sua forma originale, quella curata da D.E. Knuth e che, tra l'altro, rappresenta il termine di paragone per stabilire se un programma può contenere la stringa `tex` nel suo nome; questo programma non è dotato di molte delle funzionalità diagnostiche dei programmi di compilazione più recenti. Per esempio, sia `latex` sia `pdflatex` sono gestiti dallo stesso interprete, `pdftex`, per il quale si può applicare l'opzione `-file-line-error` (di solito questa opzione è preimpostata nelle IDE) che produce messaggi diagnostici della forma `file:riga:errore` comune al formato dei messaggi di molti programmi di compilazione. Con quel tipo di diagnostica, quanto scritto in questo paragrafo potrebbe sembrare superfluo ma non lo è: esistono alcune situazioni in cui il tipo di diagnostica descritto non è disponibile, in particolare quando si verificano errori nei pacchetti di servizio caricati nel file di formato. (*N.d.T.*)

Ma cosa succede se il numero di riga è stato segnalato alla fine della compilazione, cioè in una situazione di `level m`? Questo è il punto in cui un ulteriore comando `\end{document}` torna utile.

**Si continui a lavorare con i file di collaudo.  
Non si tocchino i file veri finché non si  
trova la fonte del problema.**

Si lavori sul main file cominciando dal fondo e si inserisca un `\end{document}` fra due direttive `\include`. L'approccio "dicotomico" è utile in questo caso: si cominci a metà. (Se ne dirà di più nel paragrafo 8 "Divide et impera".) Si compili ciò che è rimasto. Se il messaggio `level m` viene ancora emesso, il problema è nei file inclusi prima di `\end{document}`, altrimenti in quelli inclusi dopo. Si commentino le righe che contengono gli `\include` dei file "innocenti" e si sposti `\end{document}` a metà dei rimanenti; si continui così finché non rimane il solo file "colpevole". Naturalmente questo processo è più complicato se  $m > 1$ , ma il principio è lo stesso.

## 7 La pulizia del file

Ad un certo punto si è trovato il file dove si ritiene che si trovi l'errore. Può darsi che la posizione dell'errore sia identificata con precisione. Ma può succedere che si abbia ancora solo una vaga idea di dove cercare. Siccome si sta compilando un solo file di collaudo, si elimini la "confusione" togliendo ciò che non serve.

Si modifichi il main file, inserendo nell'argomento del comando `\includeonly` solo il nome del file sospetto. Si commentino anche le direttive che includono parti (irrilevanti) non introdotte mediante i comandi `\include`:

- pacchetti non necessari ai fini del collaudo;
- indici generali e liste di figure e tabelle;
- qualunque cosa legata alla bibliografia;
- comandi `\printindex` per la creazione di uno o più indici analitici.

Si pulisca anche il file sospetto. Non ci si preoccupi di modificare questo file; questa è una copia di collaudo, no? Queste sono le cose che possono essere tolte, sia pure con la dovuta attenzione:

- le righe con un segno di commento `%` all'inizio;
- le righe contenute fra `\begin{comment}` e `\end{comment}` incluse queste due;
- le righe comprese fra `\iffalse` e il corrispondente `\fi` incluse queste due (infatti questo costruito equivale ad un commento).

Attenzione a che i gruppi siano correttamente delimitati. Questo coinvolge tutte le coppie `\begin` ed `\end` di ciascun ambiente e tutti i vari metodi di raggruppamento. Si controllino questi elementi

mediante la funzionalità di conteggio del proprio editor.<sup>16</sup>

- che il numero di graffe aperte `{` sia uguale al numero di graffe chiuse `}` (attenzione al fatto che talvolta viene inserita la stringa commentata `% }` quando una graffa aperta appare solitaria nel codice; perciò si faccia attenzione a questa eventualità);
- il numero di `\begin` sia uguale al numero di `\end`;
- il numero di `\begin{group}` sia uguale al numero di `\endgroup`;
- il numero di `\[` sia uguale al numero di `\]`;
- il numero di `$` sia pari come anche il numero di `$$`.<sup>17</sup>

Si compili quello che è rimasto e si controlli via via il file `log`.

Molti problemi nascono dalla mancanza di accoppiamento degli appositi delimitatori; perciò se si è fortunati non c'è bisogno di cercare altro. Supponiamo invece che la causa dell'errore non sia stata ancora trovata.

## 8 Divide et impera

Quello che si vuole fare è isolare il capoverso, o la più piccola porzione del file, responsabile dell'errore. (Si lavori sempre su copie del file originale; è utile disporre anche di una seconda copia di riserva.)

Si cerchi un buon punto verso la metà del file che costituisca la fine di un capoverso. Vi si inserisca il comando `\endinput` preceduto da una riga vuota. Si faccia attenzione di non inserire tale comando all'interno di un gruppo di qualunque tipo, in particolare che non cada dentro il corpo di un ambiente, cioè fra un `\begin` e il corrispondente `\end`. Si compili questo file ridotto. Se non viene più segnalato nessun errore, il problema risiede nella seconda parte del file (dopo `\endinput`) non ancora elaborata. Si cancelli la parte "sana" del file e si ripeta l'operazione precedente spostando `\endinput` dentro la parte rimasta usando le stesse cautele. Si ripeta questa procedura finché il problema non viene localizzato in un frammento molto piccolo del file che si sta collaudando. Se la soluzione è ovvia, la si corregga e si collaudi il risultato. Si applichi la correzione anche al file di

16. Si potrebbe aggiungere il consiglio di impostare l'editor in modo che usi gli "hard wrap", in modo che ogni riga che compare nella finestra di editing sia numerata per suo conto, e non che i capoversi formino una sola lunga riga che nella finestra di editing compare ripiegata e numerata solo all'inizio del capoverso; questo migliora la diagnostica relativa alla ricerca della riga che contiene l'errore che si sta cercando. (*N.d.T.*)

17. Lavorando con LATEX, le coppie di dollari non dovrebbero essere mai presenti, perché è considerato un "peccato mortale" delimitare le formule in display con i doppi dollari. (*N.d.T.*)

collaudo intero (di cui si era conservata una seconda copia) e lo si compili; quando si è sicuri di avere corretto bene l'errore, si applichi la correzione al *vero* file del documento e se ne verifichi ancora la correttezza.

Ma se la soluzione non è ovvia?

Se quello che rimane è ancora troppo grande per identificare velocemente il problema — per esempio potrebbe essere una lunga dimostrazione i cui passi sono esposti mediante una lista — si faccia una copia del file con un altro nome e si continui a lavorare sul file di collaudo. (A questo autore è successo varie volte di modificare la propria *copia* che non è quella immessa dal main file. Questo genere di distrazioni può portare all'esasperazione.)

Si riduca ancora il file di collaudo commentando quegli elementi che sembrano innocui, ma per ora non si cancelli nulla — quello che si pensa sia innocuo, potrebbe invece essere parte del problema. Si continui ad iterare questo procedimento finché non c'è più la possibilità di eliminare altro materiale senza eliminare l'errore (non ancora identificato). Quello che è rimasto è diventato un esempio minimo (non-)funzionante, quello che in gergo viene indicato con la sigla “MWE” (Minimum Working Example) [o “EMC” (Esempio Minimo Compilabile) (*N.d.T.*)] — anche se non è ancora funzionante perché contiene ancora il problema non identificato.

Si esamini ciò che è rimasto nel file e

#### Si faccia attenzione agli indizi contenuti nel file log.

Naturalmente se si sa cosa correggere e come farlo lo si può verificare subito eseguendo le opportune modifiche nel file di collaudo e rilanciando L $\TeX$  per confermarlo. Se funziona, si possono applicare quelle modifiche al file *vero*, e se non si incontrano altri problemi si è a posto.

Se si incontrano altri problemi si ricomincia da capo, ma questa volta si sa come procedere.

Pero c'è un'area non ancora esplorata, cioè quando un errore si manifesta in qualcosa che si trova prima di

```
\begin{document}
```

Si veda nel paragrafo 10.

Ci si ricordi che ci sono altre tecniche che si possono seguire prima di chiedere aiuto ad altri.

## 9 Talvolta sono necessarie azioni più drastiche

In questo paragrafo continuiamo a riferirci ad errori che compaiono nel corpo del documento.

Una volta che il problema è stato ridotto alla creazione di un MWE, diventa utile servirsi di altri strumenti diagnostici per ottenere maggiori informazioni. In aggiunta ai comandi mostrati nella pagina 7, nella sezione 4 relativa alla diagnosi

interattiva, i seguenti comandi possono risultare molto utili. (Informazioni più dettagliate su questi comandi si possono trovare nel testo *L $\TeX$  by topic* (EIJHOUT, 1991).)

- `\tracingmacros` riporta i dettagli sullo sviluppo delle macro e sui valori dei loro argomenti;
- `\showboxdepth` specifica quanti livelli di “in-scitolamento” mostrare durante il tracciamento dei comandi eseguiti, dove di solito il numero è uguale a `\maxdimen` (in pratica un numero grandissimo che sta per “tutti”);
- `\showboxbreadth` specifica il numero di elementi da mostrare ad ogni livello.

Ne esistono anche altri, ma questi sono generalmente i più utili.

Se si è disperati (e masochisti) si può specificare `\tracingall` ma districarsi fra tutta l'informazione che viene riportata nel file log impegna pesantemente la pazienza e la mente dell'utente; di solito si riesce a trovare un approccio “più semplice”. Si può vedere nel file `plain.tex` quanti e quali tipi di informazione vengono forniti.

Ma se si è forzati ad eseguire il tracciamento, potrebbe esserci una strada più facile.<sup>18</sup>

## 10 Nel caso sia necessario un aiuto maggiore

Alcune risorse utili sono disponibili on-line. Potrebbero esserci altre persone che hanno incontrato lo stesso problema.

Gli archivi nel sito di `tex.sx` (STACKEXCHANGE, 2017) sono un buon punto di partenza.<sup>19</sup> Se non si trova nulla che assomigli al problema che si sta affrontando, si può sempre formulare una nuova domanda d'aiuto. (Bisogna registrarsi nel forum, se non lo si è già fatto in precedenza.) Per ottenere i migliori risultati è molto opportuno accludere un MWE; ma se si è seguita la procedura indicata, se ne dispone già di uno — l'esempio minimo (non) funzionante col quale si è “combattuto” fino al momento di chiedere aiuto ad altri. Si tolga qualunque commento, e se opportuno (e possibile), lo si renda “anonimo” sostituendo al testo vero un testo di riempimento.<sup>20</sup> Si renda questo MWE il più piccolo possibile purché continui a replicare il problema.

18. Con il pacchetto `trace` e i suoi comandi `\traceon` e `\traceoff` si può attivare un tracciamento molto selettivo; inoltre questi comandi sospendono il tracciamento sia della matematica sia della gestione dei font, riducendo di parecchie migliaia il numero delle righe da leggere. Ciò non toglie che se non si esegue un tracciamento selettivo, si rischia di doversi districare fra decine di migliaia di righe nel file log; l'uso intelligente di un IDE permette di ricercare punti particolari nel listato del tracciamento. (*N.d.T.*)

19. Anche il forum del `qJr`, <http://www.guitex.org> è particolarmente utile per gli utenti italiani o che conoscono la lingua italiana. (*N.d.T.*)

20. Fra gli altri esistono i pacchetti `lipsum` e `kantlipsum`; il primo produce interi capoversi scritti in latino maccheronico, mentre il secondo usa capoversi in inglese che assomigliano

Che cosa appariva sullo schermo:

```
Overfull \hbox (23.1113pt too wide) in paragraph at lines 3288-3301
\OML/cmm/m/it/10.95 a$\T1/ptm/m/n/10.05 , as in
```

Il corrispondente contenuto nel file log:

```
Overfull \hbox (23.1113pt too wide) in paragraph at lines 3288-3301
\OML/cmm/m/it/10.95 a$\T1/ptm/m/n/10.05 , as in Ÿ[], is a degree-
```

Che cosa c'era nel file sorgente:

```
..., as in \S\ref{SS:changing}, is a degree-1 ...
```

FIGURA 1: Un bel rompicapo

Si accludano anche righe importanti tratte dal file `log` e una spiegazione di quello che si già è fatto per affrontare il problema. I partecipanti al forum sono competenti e amichevoli verso i richiedenti aiuto; di solito sono ben contenti di risolvere il rompicapo che il problema pone — ma necessita di una quantità di informazione sufficiente che permetta loro di sperimentare; fornire un MWE che essi possano copiare e incollare è un'azione che produce riposte più rapide rispetto a quello che sarebbe necessario tirando ad indovinare.

## 11 Errori nel preambolo

1. Si faccia una copia del file `log` e lo si apra cercando l'ultimo file aperto.
2. Se questo file non è stato aperto tramite un comando `\usepackage`, si torni indietro fino a quando se ne trova uno.
3. Si ha esperienza con i comandi interni di LATEX?
4. No. Allora questo è il momento di cercare l'aiuto di un esperto. Si vada su `tex.sx` (STACKEXCHANGE, 2017). Se nessuno ha mai riferito di questo problema, si inoltri una domanda di aiuto. Bisogna essere specifici includendo il preambolo e il file `log`.
5. Sì. Allora si cerchi di determinare in che cosa consista il problema. Si controllino i rapporti su `tex.sx` (STACKEXCHANGE, 2017). Se il problema non è mai stato segnalato, lo si segnali all'autore del pacchetto.

Questo termina la discussione dei problemi che possono manifestarsi nei *propri* file. Il prossimo paragrafo descrive un problema “tosto” che l'autore ha incontrato e che ha richiesto veramente troppo tempo per essere capito e, alla fine, non era veramente un “problema di LATEX”, sebbene fosse proprio lì che il “mostro” si era manifestato.

a frasi Kantiane di un ipotetico testo *Ideale della ragion pratica*. (N.d.T.)

## 12 Un vero rompicapo

Occasionalmente nemmeno il tracciamento è in grado di indirizzare verso la soluzione del problema.

Due fatti sono importanti.

1. Io vivo e lavoro negli USA. La mia postazione di lavoro è caratterizzata con impostazioni locali (presumibilmente appropriate), cioè con la codifica ASCII.<sup>21</sup>
2. Io compilo usando la linea di comando e non uso le impostazioni `\batchmode` o `\nonstopmode`.<sup>22</sup>

Tempo fa, la compilazione di un file si bloccava sistematicamente prima che il file fosse completamente composto, congelando anche l'operatività del terminale su cui lavoravo. Per riprendere controllo della situazione era necessario aprire un'altra sessione (terminale) per forzare la cessazione del processo di compilazione sul primo terminale. Questo infatti permetteva di inviare il comando da tastiera `ctrl-C` al terminale bloccato perché ritornasse a mostrare il prompt. L'ultima cosa che appariva sullo schermo era un rapporto parziale relativo ad una scatola troppo piena. Sullo schermo appariva sufficiente testo per localizzare il problema nel file sorgente ... solo che il file sorgente aveva un aspetto perfettamente valido (si veda la figura 1). Il file `log` era presente anche se incompleto.

Dopo aver seguito i passi descritti in precedenza, sono riuscita a ridurre il file di collaudo ad una sola riga di testo che formava un semplicissimo capoverso; se toglievo qualsiasi cosa dall'inizio di questo capoverso l'errore spariva. Il problema sembrava collegato alla scatola troppo piena. Ma a questo punto ho chiesto aiuto a qualcuno più preparato di me sui sistemi informatici.

Dopo aver guardato con attenzione quello che appariva sullo schermo e nel file `log` abbiamo notato quello strano carattere — Ÿ. (Questo carattere Ÿ

21. Nell'Europa continentale tutte le lingue usano segni diacritici di vario genere; la codifica ASCII va bene, in pratica, solo per l'inglese. (N.d.T.)

22. Generalmente gli IDE sono impostati con `\scrollmode`; i due modi specificati dall'autore si riferiscono al programma `tex` originale e/o ad una impostazione locale. (N.d.T.)



ha la codifica Unicode U+0178, mentre il comando `\S` corrisponde al segno §, che occupa la posizione "78 nel font `cmsy`.) Siccome sono abituata a lavorare con testi in inglese, e molto raramente maneggio lettere accentate, non sono abituata a vedere caratteri non ASCII, e certamente mai in un file `log` relativo a un documento completamente in inglese.

La soluzione per aggirare il problema che mi è stata indicata consisteva nel mettere la linea seguente

```
LANG="en_US.utf8"
```

in un file chiamato `.i18n` nella mia cartella “home”. Questo non risolve completamente il problema — il file continua a bloccare lo schermo, ma la compilazione va a buon fine e posso dare il comando `ctrl+C` per sbloccare lo schermo e riavere il prompt.

Talvolta quello che uno pensa che sia un baco in L<sup>A</sup>T<sub>E</sub>X, non lo è affatto. Si mantenga quindi la mente aperta.<sup>23</sup>

### 13 Aggiunte successive alla conferenza

Questi che seguono sono problemi facilmente identificabili che capitano spesso, ma la sorgente di ogni problema non è sempre nota. Sembra utile indicare qui come identificarli.

1. L'avviso `Missing character`, per esempio:  
`There is no ; in nullfont!`  
 è quasi sempre il risultato di un errore sintattico — la mancanza di un ‘punto e virgola’ —

23. Non ci sono abbastanza dettagli in questo racconto per poter trovare una spiegazione per quel che succedeva all'autore. Tuttavia il problema dipendeva dalle impostazioni del terminale sulla sua macchina Linux, e da caratteri Unicode che la configurazione del terminale non poteva gestire; se l'autore avesse usato un IDE, come usiamo quasi sempre noi in Italia, e avesse usato per l'input del suo file sorgente la codifica `utf8` che usiamo abitualmente, il problema non si sarebbe nemmeno posto. La raccomandazione delle guide del G<sub>U</sub>IL è di usare sempre l'opzione `utf8` per l'impostazione dei nostri shell editor e di usare font Type 1 o OpenType a seconda del programma di compilazione che usiamo; sempre, anche se dobbiamo comporre un documento interamente in inglese; è vero che questa lingua non fa uso di accenti, ma nel momento in cui si compone una bibliografia inserendo un'opera scritta da un autore non inglese, la probabilità che il suo nome contenga accenti è molto alta; quindi non si sa mai: usare la codifica `utf8` anche quando si compone in inglese è utile. (*N.d.T.*)

in un ambiente `tikzpicture`. Altri messaggi che coinvolgono segni di interpunzione e che citano `nullfont` possono essere collegati a qualche espressione di `tikz`. Se il carattere mancante è indicato mediante un paio di segni strani, la cosa potrebbe dipendere da un errore di codifica in ingresso.

2. Analogamente avvisi di questo genere che citano nomi di font devono essere gestiti con una ricerca accurata. Spesso non è indicato il numero della riga nel file `log` ma in questo file compare il numero dell'ultima pagina spedita al file di uscita. Si confronti il file sorgente con la pagina composta e si controlli che cosa manca.

## 14 Ringraziamenti

Grazie al GUST per aver ospitato la Conferenza TUG'17 a Bachotek insieme al loro incontro annuale; e grazie ai partecipanti le cui domande dopo la mia presentazione mi hanno fornito nuove idee utili e interessanti.

## Riferimenti bibliografici

- BEETON, B. (2017). «Debugging L<sup>A</sup>T<sub>E</sub>X files — Illegitimi non carborundum». *TUGboat*, **38** (2), pp. 159–164.
- ELJHOUT, V. (1991). *T<sub>E</sub>X by topic*. Addison Wesley Publishing Company (UK). Leggibile anche con `texdoc texbytopic`.
- KNUTH, D. E. (1994). *The T<sub>E</sub>Xbook*. Addison Wesley Publishing Company, Reading.
- LAMPORT, L. (1994). *L<sup>A</sup>T<sub>E</sub>X – User's guide and reference manual*. Addison Wesley Publishing Company, Reading, MA, 2<sup>a</sup> edizione.
- STACKEXCHANGE (2017). «`tex.stackexchange.com`». Internet forum con una grande varietà di domande e riiposte. URL <https://tex.stackexchange.com>. Ultima visita ottobre 2017.
- ▷ Barbara Beeton  
 American Mathematical Society  
 Providence, RI, USA  
[bnb at ams dot org](mailto:bnb@ams.org)