

GNU Emacs e AUCTEX per L^AT_EX

Onofrio de Bari

Sommario

L'intento di questo articolo è raccogliere informazioni e risorse utili per l'impiego dell'editor GNU Emacs con l'estensione AUCTEX nella redazione di documenti L^AT_EX, presentando istruzioni e consigli che fino ad ora non sono stati portati a conoscenza degli utenti con documenti in lingua italiana.

In una prima analisi si introduce l'editor GNU Emacs, soffermandosi poi sulla sua struttura logica e su quali istruzioni possono essere date a Emacs per risultare comodo da usare con T_EX e L^AT_EX; vengono infine analizzati nel dettaglio i moduli software AUCTEX e `preview-latex`, utili nella redazione del codice sorgente e nella visualizzazione preventiva del documento.

1 Emacs e GNU Emacs

La nascita di Emacs, almeno come concetto di editore di macro (Editor MACroS), si può collocare nella metà degli anni '70. All'epoca l'editor più diffuso nei laboratori del Massachusetts Institute of Technology (MIT) era TECO, il quale aveva la peculiarità di trattare come modalità separate la digitazione del testo, la modifica e la visualizzazione del documento, un po' come l'attuale vi. Alcuni sviluppatori di software al MIT, tra cui Guy Steele e Richard Stallman, contribuirono in diversi modi alla nascita di quello che poi sarebbe stato Emacs; il primo unificò le varie macro per TECO che si erano sviluppate in quel periodo, mentre il secondo – dopo una visita allo Stanford AI Lab – implementò una macro che permetteva di avviare un sottoprogramma TECO alla pressione di un tasto; se oggi questo può far quasi sorridere, va chiarito che per l'epoca era un grandissimo successo.

Furono diverse le implementazioni di Emacs per vari sistemi operativi dell'epoca, fino a quando nacque la prima per Unix realizzata in linguaggio C, chiamata Gosling Emacs, dal nome dello sviluppatore James Gosling. Nel 1984, in coincidenza con la nascita del progetto GNU's Not Unix (GNU), Stallman iniziò a sviluppare un'alternativa libera (nel senso definito nella GNU Public License) a Gosling Emacs, realizzando anche un nuovo interprete per il linguaggio LISP, che viene tuttora usato per le estensioni software e per le impostazioni; è così che nacque GNU Emacs, la cui prima versione pubblica è la 13, del 20 marzo 1985. La versione attuale è la 21.4, del 6 febbraio 2005.

Sono oggi disponibili versioni di GNU Emacs per i vari Unix, Linux, Microsoft Windows, Mac

OS X e altri sistemi operativi.

Un'altra variante di Emacs è XEmacs, derivante da un *forking* del progetto Emacs; molte considerazioni contenute in questo articolo (se non tutte) sono valide impiegando XEmacs al posto di GNU Emacs, ma personalmente uso GNU Emacs, quindi non sono a conoscenza di eventuali discrepanze.

2 Fondamenti dell'editor GNU Emacs

Non è facile dire perché si dovrebbe scegliere GNU Emacs come editor; è questione di simpatia iniziale o di abitudine, anche considerando che nel mondo Unix vi è da decenni una rivalità, più o meno scherzosa, tra utenti di Emacs e utenti di vi.

Avviando GNU Emacs si nota subito che lo schermo (o la finestra) è diviso in due *buffer* (memorie di transito); vi è in particolare un'area grande per scrivere delle note e un'area molto più piccola, in basso, detta *minibuffer*, dove vengono visualizzati i comandi e in cui vi è il vero e proprio “dialogo” tra il programma e l'utente.

Un primo strumento d'apprendimento è il *tutorial* a GNU Emacs, disponibile anche in italiano, ottenibile dal menu Help selezionando la voce *Emacs Tutorial (choose language)*; il programma chiederà nel minibuffer quale lingua usare (con la stringa `Language:`), quindi basterà scrivere la parola `italian` e premere INVIO perché venga visualizzato un buffer con la guida a GNU Emacs in italiano.

3 Adattare GNU Emacs a L^AT_EX

Un utente di GNU Emacs che ha intenzione di compilare un documento scritto in T_EX o L^AT_EX vuole evidentemente trovarsi a proprio agio nell'ambiente di programmazione; una prima richiesta è di solito l'evidenziazione delle parole chiave del linguaggio, ottenibile con l'inserimento della riga

```
(add-hook 'tex-mode-hook
'turn-on-font-lock)
```

nel file `.emacs`, presente nella propria *home directory* nel caso in cui si usi Linux. Il file in questione contiene tutte le personalizzazioni di GNU Emacs, che vengono implementate in Elisp, un “dialetto” del linguaggio Lisp.

Nel mio file `.emacs` ho personalmente ritenuto opportuno aggiungere le righe

```
(setq tex-dvi-view-command xdvi -s 4 *)
(setq-default fill-column 72)
```

che servono rispettivamente a impostare xdvi come visualizzatore DVI predefinito e a mantenere le righe in 72 colonne per comodità di visualizzazione.

Una volta completate le operazioni di personalizzazione, non rimane molto altro che imparare dal tutorial e provare; dopo la fase iniziale di apprendimento di Emacs per l'uso con L^AT_EX si inizierà a desiderare uno strumento di editing più potente, ed è allora che viene in aiuto AUCTEX.

4 L'estensione AUCTEX

Il modulo software AUCTEX per GNU Emacs è stato realizzato nel 1992 da Kresten Krab Thorup e da altri studenti dell'Aalborg University Center (Danimarca); attualmente è sviluppato e aggiornato da David Kastrup.

Il sottotitolo "Un sofisticato ambiente T_EX per Emacs", con cui Kastrup descrive il software nella copertina del manuale, rende molto bene l'idea di quello che è AUCTEX. Non si tratta solo di una semplice raccolta di sequenze di tasti in GNU Emacs che facilitano la redazione del documento e l'immissione di comandi, ma di una collezione di funzionalità che coinvolge altresì l'interazione di Emacs con visualizzatori esterni per i formati DVI, Postscript e PDF, l'interpretazione della sintassi T_EX e L^AT_EX e il sostegno a particolarità linguistiche per documenti realizzati in lingue diverse dall'inglese.

AUCTEX è disponibile per sistemi alla Unix (Linux, FreeBSD e simili) così come per Microsoft Windows, Mac OS X e altri sistemi operativi.

Per quanto riguarda Linux, l'installazione di AUCTEX si esegue compilando il codice sorgente o utilizzando pacchetti precompilati per la propria distribuzione, mentre per Windows e MAC OS X è consigliabile procurarsi una versione precompilata di GNU Emacs contenente anche AUCTEX.

Dopo l'installazione, AUCTEX va attivato aggiungendo al proprio file `.emacs` le righe

```
(load auctex.el nil t)
(require 'tex-site)
```

allo scopo di caricare all'avvio il modulo software vero e proprio. La riga

```
(setq-default TeX-master nil)
```

fa sì che AUCTEX chieda quale sia il proprio file *master* per documenti costituiti da più file (un libro diviso in diversi capitoli, ad esempio), mentre le righe

```
(setq TeX-parse-self t)
(setq TeX-auto-save t)
```

abilitano l'interpretazione logica del codice sorgente T_EX o L^AT_EX con il salvataggio in una directory del risultato dell'interpretazione, in modo da non ripetere l'operazione nei successivi caricamenti del file.

4.1 Alcune funzionalità di AUCTEX

Il seguente elenco comprende alcune delle funzionalità aggiunte da AUCTEX alla modalità T_EX e L^AT_EX di GNU Emacs:

- inserimento di macro, ambienti e comandi mediante scorciatoie da tastiera, con un'impostazione "dialogica" nei confronti dell'utente, che viene guidato nelle scelte dei parametri opzionali e obbligatori del comando o dell'ambiente, anche tramite le funzioni di completamento della digitazione ottenibili con il meccanismo Unix della pressione del tasto TAB;
- scrittura di simboli matematici mediante scorciatoie da tastiera, attivabile con la combinazione C-c ~;
- evidenziazione e allineamento speciali per macro ed ambienti;
- esecuzione di T_EX, L^AT_EX, BIBT_EX da GNU Emacs;
- visualizzazione degli errori di compilazione;
- esecuzione di visualizzatori di file DVI, Postscript e PDF da GNU Emacs.

4.2 Impiego basilare di AUCTEX

L'inserimento di macro, ambienti e comandi con l'uso di scorciatoie è, all'inizio, l'aspetto più comodo di AUCTEX; in altri termini è possibile, a partire da un buffer vuoto di Emacs, non dover digitare nulla dei vari comandi

```
\begin{document}
...
\end{document}
```

per creare un documento L^AT_EX. Si può realizzare un documento dal titolo `prova.tex` aprendo un nuovo buffer in GNU Emacs con il comando

```
C-x C-f
```

salvandolo con il nome `prova.tex`; AUCTEX presenta quindi nel minibuffer la richiesta del file *master*

```
Master file: (default /home/guit) /guit/
```

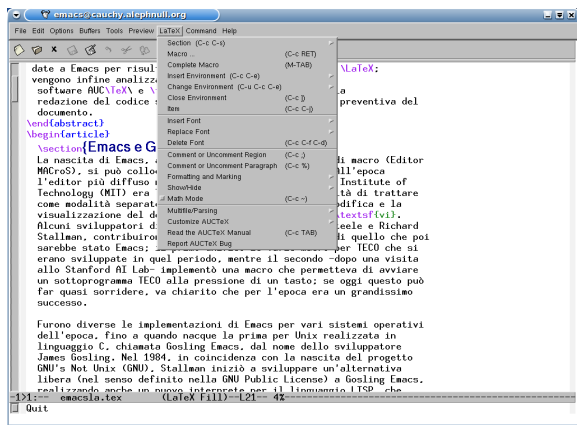
intendendo con questo nome il file principale che contiene il preambolo.

Dopo aver risposto alla richiesta compare sullo schermo un buffer vuoto e digitando la sequenza per creare un nuovo ambiente L^AT_EX

```
C-c C-e
```

nella finestra minibuffer di GNU Emacs apparirà la scritta

```
Environment type: (default document)
```

Figura 1: Il menu L^AT_EX di AUCT_EX.

che chiederà quale ambiente si intende usare (in questo caso è `document` perché si vuole iniziare un documento *ex-novo*).

Dopo la pressione del tasto INVIO si vedrà comparire la scritta

```
Document class: (default article)
```

e quindi si sceglierà la classe di documento; a questo punto nella finestra del minibuffer apparirà la scritta

```
Options:
```

che invita l'utente a digitare le opzioni per la classe di documento.

Il risultato ottenuto sarà qualcosa di simile a quanto segue

```
\documentclass[a4paper,11pt]{article}

\begin{document}

\end{document}
```

e ottenuto senza scrivere codice L^AT_EX.

Quanto è stato ora descritto è solo un banale e primitivo aspetto delle potenzialità di AUCT_EX, ma può servire a dare l'idea di quanto ci si può aspettare da questo modulo aggiuntivo per GNU Emacs.

4.3 Il menu L^AT_EX in AUCT_EX

La prima voce, *Section*, del menu L^AT_EX di AUCT_EX permette di inserire i comandi di sezionamento nel documento.

La voce *Macro* permette l'inserimento di una macro, mentre con *Complete Macro* è possibile scrivere parti di comando L^AT_EX per far completare il nome a AUCT_EX; a titolo di esempio, per scrivere un comando `\renewcommand` basta scrivere "`\renewc`", quindi premere la sequenza M-Tab (nei personal computer il tasto Alt seguito dal tasto Tab) per ottenere il comando completo `\renewcommand`.

Selezionando invece *Insert Environment* si può inserire un ambiente senza dover digitare tutti i caratteri che compongono la sintassi; in questo senso AUCT_EX è un antesignano dei sistemi "visuali" come LyX, con la differenza che in AUCT_EX si possono utilizzare tutte le potenti funzionalità di GNU Emacs e si può facilmente lavorare "a mano" con il codice, possibilità che –personalmente– trovo abbastanza preclusa in ambienti di scrittura L^AT_EX con approccio visuale.

Change Environment si rivela anch'essa molto utile, dato che molto spesso accade di cambiare ambiente in L^AT_EX perché la resa tipografica non soddisfa. Si può pensare al caso in cui, nella redazione di un documento $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX, si sceglie prima un ambiente `align` per scrivere una formula, mentre poi alcune modifiche nel contenuto della stessa fanno optare per un ambiente `gather`; a quel punto basta selezionare *Change Environment* dal menu quando si è con il cursore dentro l'ambiente da sostituire (o servirsi della scorciatoia C-u C-c C-e) per ottenere il nuovo ambiente desiderato.

Per concludere la descrizione del sottomenu riguardante gli ambienti, *Close Environment* permette di aggiungere semplicemente un

```
\end{environment}
```

dove `environment` indica l'ambiente utilizzato.

Il sottomenu successivo riguarda i comandi relativi alle fonti; molto semplicemente, con *Insert Font* si potrà scegliere se inserire il carattere in corsivo, maiuscoletto e così via, mentre con *Replace Font* si possono sostituire tali attributi in modo analogo a quello descritto per gli ambienti; con *Delete Font*, infine, si cancella il punto più interno contenente una specifica di fonte.

In un altro sottomenu troviamo funzionalità inerenti alla gestione del codice sorgente nel suo aspetto; la voce *Comment or Uncomment Region*, in particolare, permette di far precedere ogni riga della regione da un carattere % di commento. In GNU Emacs una regione è un'area selezionata con il mouse o con una *mark* (comando C-<SPC>, documentato a pag. 71 in STALLMAN (2002)). L'operazione *Comment or Uncomment Paragraph* svolge lo stesso compito per un paragrafo.

Con *Formatting and Marking* si possono ottenere degli spazi bianchi all'inizio delle righe che riflettono la struttura sintattica del documento, al fine di permettere una maggiore leggibilità e una più agevole ricerca degli errori (il mancato bilanciamento delle parentesi, ad esempio).

Da *Show/Hide* si può accedere al *Fold Mode*, un'interessante caratteristica che permette di nascondere oggetti che non sono parte del testo che si sta scrivendo, come note o citazioni, oppure di evitare che compaiano nel testo le sequenze di cambiamento degli attributi tipografici delle fonti. Se ad esempio in una parte di testo si è scritto

`\texttt{pippo}`

quando si selezionerà la voce *Show/Hide* → *Fold Mode* per attivare la funzione e quindi *Show/Hide* → *Hide Current macro* si leggerà nel buffer di Emacs solo la stringa `pippo`, opportunamente evidenziata e/o colorata in base alle impostazioni predefinite o personalizzate della modalità *Font Lock*, che permette di evidenziare con un particolare colore o fonte tipografica il codice di *markup* costituito da ambienti o comandi. Tutto questo si può evidentemente estendere al nascondere e al far riapparire tali stringhe “accessorie” in una regione o in un intero buffer.

La scrittura di simboli matematici e testo “matematico” mediante scorciatoie, attivabile selezionando la voce *Math Mode* è uno dei punti di forza di AUCTEX; servendosi di tale modalità si possono, ad esempio, inserire un comando come `\Delta` semplicemente con la sequenza ‘D. È superfluo dire che tutto questo vale anche per inserire in breve nomi di operatori, simboli relazionali, caratteri in grassetto e quanto è utile per indicare oggetti diversi nella tipografia matematica. Per conoscere in dettaglio le sequenze da usare si consulti la *AUCTEX Reference Card*, presente nella distribuzione di AUCTEX nel file `tex-ref.tex` e comunque facilmente reperibile su Internet.

Con *Multifile/Parsing* si può controllare in maniera alquanto agevole un documento con struttura multipla costituito da un file principale (*master file*) e da altri file subordinati.

La voce *Customize AUCTEX* permette di accedere al buffer di personalizzazione di AUCTEX, nello stile grafico delle ultime versioni di GNU Emacs; dalla versione 20 infatti le modifiche al file `.emacs` possono essere effettuate non solo digitando codice Elisp ma anche impostando le variabili di ambiente di Emacs dal buffer di personalizzazione.

È molto comoda la possibilità di leggere come buffer di Emacs il manuale di AUCTEX selezionando la voce *Read the AUCTEX Manual*; verrà visualizzato un buffer con il manuale in formato TeXinfo.

Selezionando infine la voce *Report AUCTEX Bug* è possibile informare gli autori del software di bug presenti (naturalmente avendo cura di controllare che non siano già elencati tra i bug noti).

5 L’estensione `preview-latex`

Il modulo `preview-latex` permette di visualizzare nel buffer di GNU Emacs contenente il sorgente del documento un’anteprima di quello che sarà il risultato finale della compilazione, potendosi così vedere le formule matematiche, le figure, i titoli di capitoli e paragrafi. `preview-latex` è stato realizzato da David Kastrup, Alan Shutko, Jan-Åke Larsson e Nick Alcock; in precedenza veniva distribuito separatamente, ma dalla versione 11.81 è integrato

in AUCTEX almeno nel senso che le procedure di installazione dei due moduli software sono state unificate.

Dopo aver installato `preview-latex` occorre attivarlo con la riga

```
(load preview-latex.el nil t)
```

inserita nel file `.emacs`. All’apertura di un file con estensione `.tex` in GNU Emacs si vedrà quindi un menu *Preview* con dei sottomenu autoesplicativi che seguono la logica del menu L^AT_EX di AUCTEX. Il primo è dedicato alla generazione di un’anteprima per un particolare punto del testo, per un ambiente, per un paragrafo, per una regione o per tutto il documento, mentre il secondo riguarda la rimozione delle anteprime.

Se si usa Emacs in modalità grafica si può, con l’aiuto del mouse, agire sulle singole anteprime. Portando, ad esempio, il puntatore del mouse su un titolo di paragrafo del quale è visualizzata un’anteprima, compare un piccolo menu; se si clicca con il tasto centrale del mouse si disabilita l’anteprima rendendo esplicito il codice “sottostante”, mentre se si clicca con il tasto destro si apre un menu con più opzioni, tra cui quella di disabilitare l’anteprima, rimuoverla per tornare alla normale visualizzazione del codice o rigenerarla.

Il sottomenu *Turn Preamble Cache* attiva o disattiva la funzionalità con la quale, alla prima richiesta di anteprima, il preambolo del documento viene messo in cache per realizzare un “file di formato” che servirà per velocizzare la resa delle successive anteprime, evitando che ogni volta vengano di nuovo esaminati i comandi che compongono la parte dichiarativa del documento.

Le voci *Customize*, *Read Documentation* e *Report Bug* richiamano le medesime funzioni di quelle presenti nel menu L^AT_EX di AUCTEX.

6 Conclusioni

Non sarebbe stato possibile, non solamente per ovvi motivi di spazio, descrivere tutte le potenzialità dell’uso di GNU Emacs e AUCTEX; quello che però voglio sperare sia avvenuto è almeno un semplice interessamento nei confronti di un approccio “vecchio stile” alla redazione di documenti L^AT_EX che, a mio personale parere, ha non solo “fascino” per un utente un po’ *geek* ma contiene in sé una grande funzionalità.

Pur sfuggendo alla banalità del “semplice è bello”, ritengo che in un’epoca in cui si cerca eccessivamente nei software il fascino “visivo” sia una buona cosa servirsi di software che, pur se solo testuale, garantisce un valido compromesso tra l’essere a proprio agio con la tastiera e il mouse e la comoda visualizzazione su schermo delle informazioni, in special modo per l’utente che opera nel settore scientifico e non cerca fronzoli, bensì rigore ed efficienza. GNU Emacs e AUCTEX usati per redigere

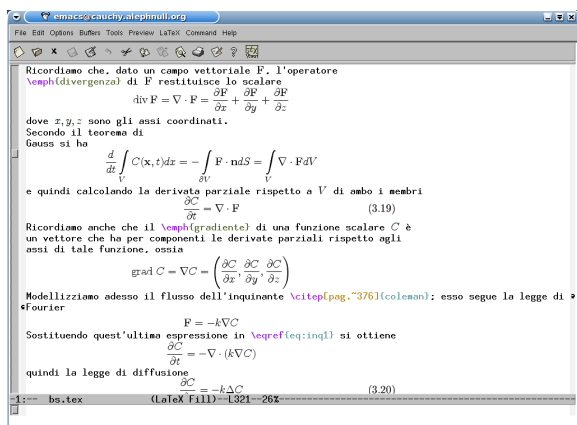


Figura 2: L'estensione preview-latex.

documenti TEX e LATEX sono sicuramente dei buoni candidati a raggiungere questo scopo – magari

dopo una fase di apprendimento iniziale un po' ripida; non è però una sensazione che ognuno di noi ha provato iniziando a imparare TEX o LATEX?

Lascio valutare a chi legge se GNU Emacs e AUCTEX possono essere un buon mezzo per non limitarsi a “lasciar fare tutto al computer” e per avere comodità di uso unita al facile controllo su quel codice sorgente che da anni permette di ottenere i meravigliosi documenti realizzati con il sistema tipografico del professor Donald Knuth.

Riferimenti bibliografici

STALLMAN, R. (2002). *GNU Emacs Manual*. Free Software Foundation.

▷ Onofrio de Bari
onodebari@gmail.com