

L^AT_EX e la cesura delle parole in fin di riga

Claudio Beccari

Sommario

Questo articolo è destinato a spiegare come fa il sistema T_EX a comporre i capoversi, all'occorrenza dividendo in sillabe le parole in fin di riga.

Questo articolo serve per fornire le indicazioni necessarie per capire certi funzionamenti strani del compilatore L^AT_EX (in realtà dell'interprete T_EX) quando sembra rifiutarsi di spezzare le righe in modo adeguato. L'errore, se c'è, è sempre umano, ahimè, e deriva dall'insufficiente comprensione del funzionamento della macchina che in questo caso è T_EX.

Abstract

This tutorial explains how T_EX (the program) typesets paragraphs, possibly by hyphenating words at line breaks.

This tutorial should explain L^AT_EX's (actually T_EX's) strange behavior in certain circumstances when apparently refuses to correctly break lines. If there is some error, unfortunately this is always a human one, and it is due to an insufficient understanding of T_EX's procedures and algorithms.

1 Composizione dei capoversi

T_EX forma i capoversi leggendo dal file sorgente la sequenza di lettere, parole, spazi, comandi, eccetera, per formare una lunga riga che contiene tutto il capoverso¹. Nel momento in cui questa lunga riga è pronta, T_EX ha già eseguito tutte le macro e i comandi che riguardano il capoverso stesso; in particolare ha già eseguito le scelte dei font, l'inserzione dei caratteri, compresi quelli a cui il compositore accede per mezzo di comandi, ha inserito i richiami di nota e le espressioni matematiche in linea, eccetera.

Generalmente questa lunga riga va spezzata in righe della giustezza specificata per quel determinato capoverso, che potrebbe trovarsi nel testo principale, o dentro una lista, o dentro una `minipage` o una `\parbox`. Le giustezze cambiano quindi a seconda della posizione del capoverso, ma che la giustezza sia pari a `\textwidth`, a `\linewidth`, a `\columnwidth` o ad un altro valore non ha nessuna importanza; resta il fatto che la lunga riga deve

essere spezzata in parti più corte tutte della stessa lunghezza, tranne l'ultima riga del capoverso.

T_EX cerca di trovare dapprima le divisioni della riga in parti senza dividere in sillabe le parole in fin di riga, inserisce la gomma elastica negli spazi interparola e attribuisce un coefficiente di merito al risultato basandolo sul calcolo di quanto ha dovuto allargare o restringere ogni riga parziale: il coefficiente di merito, o meglio di demerito, si chiama *badness* o *bruttezza*. Se questa bruttezza è inferiore ad un certo valore specificato nel file di classe con cui si sta componendo (il parametro `\pretolerance`), bene, la divisione è fatta e il capoverso così formato dalle diverse righe parziali viene aggiunto alla pagina che si sta componendo, prima di attivare l'algoritmo per spezzare la pagina; via via che questa pagina va riempiendosi, viene determinata la sua "bruttezza" in base a quanto bisogna allargare o restringere la gomma verticale per raggiungere l'altezza della gabbia di composizione e poi, quando questa bruttezza è minima, si taglia via la parte superiore insieme alle note e agli oggetti flottanti di cui si è già tenuto conto e si accoda questo materiale al file DVI o PDF che si sta componendo.

Tuttavia, ritornando al capoverso, se nessuna divisione in righe parziali, fra le tante divisioni possibili, della lunga unica riga riesce a dare una bruttezza inferiore a `\pretolerance`, allora T_EX riprende la lunga unica riga e cerca le divisioni spezzando alcune parole in fin di riga; per ogni divisione in sillabe, cioè per ogni cesura eseguita in fin di riga, alla riga viene associata una penalità che fa aumentare la bruttezza. Se due righe consecutive hanno subito la cesura viene aggiunta un'altra penalità; se la penultima riga del capoverso ha subito la cesura si aggiunge ancora un'altra penalità. Di tutte queste penalità si tiene conto insieme ai coefficienti di allargamento o restringimento delle varie righe parziali per determinare la bruttezza complessiva per le varie possibili scelte delle cesure. Viene scelto quell'insieme di cesure che produce la minima bruttezza. Se questa scelta produce un bruttezza inferiore ad un secondo parametro `\tolerance` specificato nel file di classe, bene, il capoverso viene accodato così diviso in righe alla scatola 255, altrimenti T_EX fa un terzo tentativo con una tolleranza maggiore, ma comunque emette un messaggio di avvertimento segnalando la riga peggiore della prescelta divisione in righe e produce i vari messaggi di `overfull hbox` o di `underfull hbox`; le righe (`hbox`) troppo

1. Si ricorda che in italiano si chiama *capoverso* quello che in inglese si chiama *paragraph*; e in italiano si chiama *paragrafo* quello che in inglese si chiama *section*. Bisogna evitare di confondere i concetti.

piene fuoriescono dal margine destro e ne viene comunicato l'ammontare solo se supera la quantità specificata con il parametro `\hfuzz`. Le righe troppo vuote, cioè contenenti troppo spazio bianco, sono state riempite di colla che ha dovuto essere allungata troppo contribuendo in modo sostanziale alla bruttezza del capoverso; per queste righe viene comunicata la bruttezza, solo se è superiore al parametro `\hbadness` specificato nel file di classe. Generalmente `\hfuzz` è dell'ordine di grandezza di un punto tipografico, circa un terzo di millimetro. La bruttezza che passa senza messaggi sullo schermo è dell'ordine di qualche centinaio; una bruttezza di 10000 per T_EX significa "bruttezza infinita".

È evidente che per comporre bene T_EX deve avere le condizioni migliori per farlo; con righe più lunghe di 100 mm la possibilità di dividere bene il capoverso è praticamente assicurata in ogni caso scrivendo in italiano, purché non si usino font della famiglia a spaziatura fissa, per i quali la cesura è interdotta. In italiano si riesce a comporre abbastanza bene fino a giustezze di circa 30 mm, ma ovviamente la cosa dipende molto dal contenuto del capoverso; in altre lingue giustezze così piccole generano una quantità di avvertimenti di `overfull` e `underfull hbox`. È quindi chiaro che la capacità di giustificare bene le righe di un capoverso dipende molto dall'efficienza dell'algoritmo di divisione in sillabe oltre che dalle regole grammaticali di ogni lingua.

2 Le parole del file sorgente

Il programma T_EX non conosce la lingua con cui sta componendo; riesce a distinguere le parole con regole euristiche abbastanza semplici:

1. una parola è una sequenza di lettere; una lettera è un segno di un font al quale è stato attribuito il codice di categoria² 11 o 12 e al quale è associato un codice di lettera minuscola (ritorneremo su questi concetti fra poco);
2. la sequenza di lettere è composta tutta con lo stesso font;
3. la sequenza di lettere comincia dopo uno spazio (colla) o dopo un segno di categoria 12;
4. la sequenza di lettere finisce con il primo carattere o con il primo oggetto diverso da una lettera, sia esso uno spazio, un segno di interpunzione, un comando, una scatola, della matematica, o qualsiasi altra cosa diversa da una lettera;

2. Per una chiara spiegazione dei codici di categoria ci si può riferire all'articolo di GREGORIO (2006), oltre che all'immane *T_EXbook* di KNUTH (1984).

5. dopo la fine di quella che T_EX ha identificato come una sequenza di lettere, T_EX non ricomincia ad esaminare altre sequenze di oggetti (*token*, in gergo T_EX) finché non trova di nuovo un valido inizio di sequenza di lettere.

In generale non è così scontato che una sequenza di lettere che per noi è una parola lo sia anche per T_EX. Per esempio, la parola "elettricità" è completamente una parola se il font di uscita è codificato con la codifica T1³ o equivalente, ma la parola che T_EX riconosce è solamente "elettric" se sta lavorando con la codifica OT1, perché la rappresentazione interna della 'à' è data da "elett_{ri}ci_t\a" ovvero "elett_{ri}ci_t\accent18a"; il comando per collocare l'accento termina la sequenza di lettere che T_EX crede che sia una parola. In definitiva con la codifica T1 T_EX riesce a trovare le possibili posizioni delle separazioni delle sillabe "e-let-tri-ci-tà", mentre con la codifica OT1 ne perde una e nel caso riesce a dividere la parola "elettricità" solo con la possibile divisione "e-let-tri-ci-tà".

In francese la cosa è ancora più critica; la parola "électricité" è divisibile in "é-lec-tri-ci-té" con la codifica T1, ma non è divisibile per niente con la codifica OT1, perché dopo il primo comando per l'accento, T_EX non trova un altro possibile inizio di parola se non con l'inizio della parola successiva.

Va notato che negli esempi riportati sopra si sono indicati i possibili punti di cesura secondo le regole grammaticali. In tipografia le regole di divisione devono rispettare sia le regole grammaticali, sia le regole di buona composizione tipografica, secondo la quale non si va mai a capo in fin di riga lasciando nella riga una sillaba troppo corta o rimandando alla riga successiva una sillaba troppo corta. Per ogni lingua è definito un valore per ciascuno dei due parametri `\leftthyphenmin` e `\rightthyphenmin`; per l'inglese e il francese essi valgono rispettivamente 2 e 3; per l'italiano essi valgono 2 e 2; per il greco 1 e 1; questi numeri rappresentano il numero di lettere minimo accettabile per la sillaba a sinistra della prima cesura tipografica e, rispettivamente, a destra dell'ultima cesura tipografica. Di fatto, se avendo in vigore la codifica T1, si dessero i comandi

```
\showhyphens{elettricità}
\showhyphens{électricité}
```

3. Qui si dà per scontato che il lettore abbia un minimo di conoscenza delle codifiche OT1 e T1 per i caratteri latini che vengono usati dal sistema T_EX. La vecchia codifica OT1 corrisponde a font che contengono solo 128 segni (codifica a 7 bit) e quindi non può contenere le lettere accentate, ma produce l'uscita mediante comandi di basso livello che sovrappongono il segno dell'accento al segno della lettera. La codifica T1 (codifica a 8 bit) contiene 256 segni che comprendono la quasi totalità dei segni accentati che si usano nelle lingue europee, quindi l'uscita è prodotta direttamente con i segni accentati e senza acrobazie per sovrapporre l'accento sulla lettera da accentare.

L^AT_EX fornirebbe sullo schermo e nel file `.log` le informazioni seguenti

```
Underfull \hbox (badness 10000)
      in paragraph at lines 146--146
[] \T1/cmr/m/n/10.95 elet-tri-ci-tà
```

```
\hbox(7.54149+0.0)x16383.99998,
      glue set 8998.3789 []
```

```
Underfull \hbox (badness 10000)
      in paragraph at lines 148--148
[] \T1/cmr/m/n/10.95 élec-tri-cité
```

```
\hbox(7.54149+0.0)x16383.99998,
      glue set 8998.3789 []
```

Il comando `\showhyphens` compone il suo argomento dentro una scatola orizzontale di lunghezza pari a 16384 punti (circa 5 m) e ovviamente la scatola risulta quasi vuota e infinitamente mal composta (`badness 10000`) per cui viene emesso il messaggio di avvertimento e le informazioni relative alla scatola mal composta con il suo contenuto diviso in sillabe come T_EX lo farebbe. Si vede chiaramente che in italiano è persa la prima possibile cesura grammaticale, perché lascerebbe a sinistra un sillaba di una sola lettera. Analogamente per il francese vengono perse sia la prima sia l'ultima cesura grammaticale, perché le sillabe a sinistra e a destra conterebbero meno caratteri di quanto specificato dai parametri `\dots hyphenmin` validi per il francese.

Grazie al fatto che in italiano gli accenti obbligatori cadono solamente sulla vocale terminale delle parole tronche, la convenzione tipografica non produce effetti molto diversi con le codifiche OT1 o T1; in francese, al contrario, l'effetto con la codifica OT1 sarebbe devastante.

Va notato che per l'italiano le regole di sillabazione sono state create assumendo i valori di `\dots hyphenmin` entrambi pari a 1, ma le impostazioni nel file di descrizione della lingua italiana li mettono entrambi a 2. Nessuno vieterebbe al compositore di assegnare i valori unitari ai due parametri suddetti, ma certamente la sua composizione sarebbe soggetta a critiche estetiche non indifferenti.

3 I codici di categoria

I codici di categoria, in particolare i codici 11 e 12 si riferiscono rispettivamente alle lettere dell'alfabeto e ai segni di interpunzione. Le impostazioni di L^AT_EX sono tali che se viene invocato il pacchetto `fontenc` con l'opzione T1, sono dichiarate lettere non solo le 26 lettere minuscole e le 26 lettere maiuscole dell'alfabeto latino, ma anche tutte le loro varianti con tutti i possibili segni diacritici oltre ai segni speciali che compaiono in varie lingue, come

per esempio ß, ŋ, ð, đ, þ. A ogni lettera minuscola è associato il codice della corrispondente lettera maiuscola e viceversa, così che possano essere eseguite le trasformazioni da maiuscolo a minuscolo e viceversa mediante i comandi `\MakeUppercase` e `\MakeLowercase`.

Tuttavia il codice di lettera minuscola è un requisito essenziale per considerare un carattere come facente parte di una parola. Anche i segni di interpunzione possono avere associato il codice di lettera minuscola, sebbene a prima vista non si capisca bene a che cosa possa servire. Serve semplicemente a T_EX per stabilire se quel segno può alterare la divisione in sillabe oppure no. Le regole devono quindi tenere conto della presenza di questi segni considerandoli come parti integranti della parola da dividere in sillabe.

Non è che ci siano tanti segni di questo genere. Lo scrivente è al corrente che solo l'italiano, il francese e il catalano usano l'apostrofo con il significato di elisione di una vocale. In questo caso una parola in senso lato come "bell'accoglienza" è divisibile in sillabe dopo l'apostrofo se e solo se l'apostrofo ha associato il codice di lettera minuscola (un valore qualunque non negativo, e tanto vale allora che questo valore coincida con l'indirizzo dell'apostrofo nella polizza dei caratteri in uso). Si noti che se l'apostrofo non fosse associato al suo valore di lettera minuscola, l'esempio proposto sopra troverebbe T_EX incapace di dividere in sillabe "bell'accoglienza", perché l'unica parola che riconoscerebbe sarebbe "bell", divisibile grammaticalmente in "bel-l", ma l'ultima sillaba a destra sarebbe troppo corta. Dichiarando che l'apostrofo, pur mantenendo la categoria 12 di segno non letterale, ha associato un codice di lettera minuscola, T_EX riconosce "bell'accoglienza" come un'unica parola e deve conoscere le regole di divisione per parole composte anche di apostrofi. La sua divisione, con queste regole, è pertanto "bel-l'ac-co-glien-za"; come si vede, queste regole impediscono di andare a capo dopo l'apostrofo: questa pratica è comunemente accettata in tipografia, anche se si chiude un occhio quando si compone in colonne molto strette; tuttavia T_EX così come è configurato si rifiuta categoricamente di eseguire una operazione del genere.

4 I pattern di sillabazione

Le regole grammaticali di divisione in sillabe variano ovviamente da lingua a lingua: per l'italiano sono molto semplici; in francese esse sono un po' più complesse. Per l'inglese sono estremamente complesse da eseguire a macchina perché dipendono dalla accentazione e parole omografe risultano divise in sillabe differentemente a seconda di come vengono pronunciate: si pensi che *record* (verbo) e *record* (sostantivo) vengono pronunciate rispettivamente *recòrd* e *rècord*; le divisioni in sillabe

diventano pertanto rispettivamente *re-cord* e *rec-ord*; lo stesso succede per parole come *analyses* pronunciato *anàlyses* (plurale del sostantivo *analysis*) o *analýses* (terza persona singolare del verbo *to analyse*); inoltre le regole di sillabazione per la varietà americana sono diverse da quelle della varietà britannica.

In tedesco sono complessissime a causa delle parole composte, dove la cesura non può non cadere nei punti di giunzione delle parole componenti e può provocare anche un cambio di ortografia.

Tenuto conto del fatto che l'algoritmo di divisione in sillabe viene invocato da T_EX molto frequentemente, è necessario che queste regole siano formulate in modo che T_EX le possa consultare con la massima rapidità.

Frank Liang era studente a Stanford quando Knuth stava producendo la prima versione del programma T_EX; egli svolse la sua tesi di master sulla sillabazione mediante calcolatore, LIANG (1983); il suo problema era quello di trovare una struttura dati molto rapida da esplorare e sviluppò una struttura nuova che risultò particolarmente efficace. La sua tesi è accessibile alla lettura di molte persone che abbiano una minima conoscenza dell'informatica, o meglio della "computer science": anche se perdono qualche dettaglio teorico, la lettura è affascinante.

Il metodo di scrittura delle regole di Liang si basa sui *pattern*; questi sono frammenti di parola che contengono le informazioni per inserire o vietare una cesura fra le lettere che compongono il frammento. Usare i frammenti di parola rende l'elenco delle cesure permesse o vietate molto più compatto che non un intero dizionario di parole divise in sillabe; inoltre questo insieme di frammenti può essere messo dentro una struttura dati adeguata per la ricerca rapida dei frammenti.

Ciò richiede che sia creata la lista dei *pattern* e che questa sia predigerita da T_EX in modo che la struttura dati sia già pronta quando T_EX lavora per la composizione.

Il primo punto può essere eseguito con il programma *patgen*, mentre il secondo è quell'aspetto delicato che sfugge alla maggior parte degli utenti del sistema T_EX, cioè che questa predigestione è svolta nel momento in cui si crea il file di formato con estensione *.fmt* o *.efmt*. Spesso gli utenti si lamentano dicendo: "Ho caricato il pacchetto *babel* con l'opzione *italian*, ma poi L^AT_EX divide le sillabe in modo sbagliato". Certo, ciò succede perché T_EX non trova nel file di formato le regole di sillabazione nella struttura dati con i *pattern* per l'italiano, perché il responsabile del sistema, di solito il proprietario o l'amministratore del computer, si è dimenticato di creare o ricreare il formato.

4.1 Il file di formato contiene i pattern di sillabazione

Le varie distribuzioni del sistema T_EX contengono meccanismi vari per creare il file di formato; alla fin fine il tutto si riduce a eseguire il programma *initex* specificando come input il nome del file di cui si vuole predigerire il contenuto e creare il file di formato.

Lanciando *initex* sul file *latex.ltx* viene creato il formato *latex.fmt*; un file di configurazione specifica per quali lingue siano da incorporare i *pattern*; molte delle ultime distribuzioni del sistema T_EX predigeriscono in fase di installazione tutti i *pattern* di tutte le lingue di cui esistono i corrispondenti file di *pattern*. MiK_T_EX ha un comodo programma di configurazione con interfaccia grafica che consente di marcare le lingue desiderate e di eseguire la creazione dei file di formato con pochi click su appositi bottoni.

Ogni volta che si cambia l'insieme delle lingue bisogna ricreare i file di formato; la prima volta che si installa la distribuzione del sistema T_EX bisogna creare il file di formato o almeno bisogna verificare che il file di default creato durante l'installazione contenga le regole di sillabazione per tutte le lingue che interessano, altrimenti bisogna ritornare sulla configurazione, eseguire le modifiche necessarie e, infine, ricreare i file di formato.

Per essere sicuri che il proprio sistema T_EX sia correttamente configurato anche per l'italiano basta creare il semplice file sorgente seguente

```
% file sillabazione.tex
\documentclass{minimal}
\usepackage[italian]{babel}
\begin{document}
\showhyphens{equazione}
\end{document}
```

Lanciando la compilazione di questo brevissimo file non si produce nessuna pagina in uscita ma si ottiene sullo schermo (e nel file *.log*) quello che L^AT_EX pensa sia la divisione in sillabe corretta; se sullo schermo appare *equa-zio-ne*, l'italiano è ben configurato anche come sillabazione; se sullo schermo appare *equ-a-zione* la divisione in sillabe corrisponde alla lingua di default, cioè all'inglese. Bisogna quindi rivedere la questione della configurazione e procedere alla creazione del nuovo formato.

4.2 Come funzionano i pattern

I *pattern*, dunque, sono dei frammenti di parola (scritti in lettere minuscole) dove sono indicati i possibili punti dove la divisione in sillabe è consentita oppure è vietata. Questi punti sono indicati mediante cifre da 0 a 9 intercalate alle lettere: le cifre dispari consentono la cesura in quel punto, mentre le cifre pari la vietano; la cifra zero è sottintesa e quindi non viene quasi mai indicata

b	e	l	l	'	a	c	c	o	g	l	i	e	n	z	a
b0e01010	'0a0c0c0o0g010i0e0n0z0a														
110	0'2	1c0	1g0	1n0											
110	1c0	110	1z0												
21010	2c0c0	0g210	2n0z0												
<hr/>															
b0e21110	'2a2c1c0o1g210i0e2n1z0a														
<hr/>															
bel-l'ac-co-glien-za															

TABELLA 1: Processo di divisione in sillabe mediante i pattern della lingua italiana; la seconda riga della tabella contiene i codici di default, tutti valori nulli a sinistra e a destra di ogni lettera.

esplicitamente. Se ad una parola si applicano diversi pattern e fra due lettere un pattern indica un valore mediante una cifra e fra le stesse due lettere un altro pattern indica un diverso valore, prevale il valore più grande.

Nella tabella 1 è indicata la serie di pattern che si applica alla parola “bell'accoglienza”, intesa da T_EX come un'unica parola, mentre noi umani sappiamo che si tratta di due parole fra le quali è intervenuta l'elisione.

Dalla tabella si può osservare, esaminando ogni riga dall'alto al basso, la parola iniziale seguita dai vari pattern che si trovano nel file `ithyph.tex` e che contengono frammenti della parola da dividere in sillabe: i pattern sono incolonnati insieme ai valori numerici che informano sulla possibilità di inserire o l'obbligo di non inserire una cesura; in questo modo è agevole controllare colonna per colonna qual è il valore numerico più alto e definire quindi l'intera parola con tutti i valori numerici intercalati. In corrispondenza dei valori finali dispari è possibile inserire la cesura, per cui il risultato finale riportato nell'ultima riga è perfettamente coincidente con quanto insegna la grammatica.

4.3 La creazione dei pattern

Per usare il programma `patgen`, creato da Liang a questo scopo, sarebbe necessario disporre di un elenco di parole con le divisioni in sillabe già segnate; Liang ha fatto questo lavoro per l'inglese usando circa 25 000 parole e ha ottenuto per la lingua inglese della varietà americana poco meno di 4800 pattern. Riapplicando questi pattern all'elenco delle 25 000 parole ha constatato che il 90% di queste risultavano divise in modo corretto. Il che la dice lunga sulla difficoltà di dividere in sillabe le parole inglesi.

Il programma T_EX accetta anche un elenco di eccezioni dato come argomento al comando `\hyphenation`:

```
\hyphenation{<lista delle eccezioni>}
```

dove la *lista delle eccezioni* è costituita dall'elenco delle parole scritte in lettere minuscole e con i trattini indicanti le cesure lecite. In italiano si potrebbero per esempio inserire nella lista le parole

composte che si desidera dividere etimologicamente invece che foneticamente come la grammatica consente e suggerisce:

```
\hyphenation{su-per-in-dut-to-re
su-per-in-dut-to-ri
ma-cro-istru-zio-ne
ma-cro-istru-zio-ni}
```

Per l'americano la lista delle eccezioni contiene ora molte centinaia di parole e la rivista TUGboat pubblica gli aggiornamenti regolarmente; l'elenco aggiornato si trova sempre negli archivi CTAN.

Per l'italiano lo scrivente ha inizialmente creato una lista di pattern usando solo la grammatica e ha controllato la correttezza delle cesure su un elenco di parole appositamente creato per eseguire questo controllo; dopo pochi aggiornamenti il file `ithyph.tex`, che accompagna ogni distribuzione del sistema T_EX, è rimasto praticamente inalterato e contiene circa 320 pattern con il livello più alto di consenso/proibizione di eseguire la cesura che arriva alla cifra 4.

Successivamente ha avuto a disposizione una versione del programma `patgen` accompagnato da istruzioni per l'uso abbastanza decifrabili ed ha provato a verificare i suoi pattern alla luce dello stesso elenco di parole appositamente costruito. I pattern ottenuti sono stati marginalmente in numero maggiore e il massimo livello di consenso/proibizione arriva a 5. Tutto questo conferma la bontà del programma `patgen` che arriva quasi a fare altrettanto bene di quello che può fare una persona istruita.

Nella tabella 2 si riporta solo l'elenco dei pattern tratto dal file `ithyph.tex` per eseguire qualche commento; il lettore interessato può avere a disposizione per la lettura l'intero file che è contenuto nella sua distribuzione del sistema T_EX. Ci si ricordi solamente che nei pattern il punto segna l'inizio o la fine di una parola.

Si nota che l'intero contenuto della tabella di pattern è racchiuso fra le parentesi graffe; in questo modo eventuali definizioni o assegnazioni di valori restano confinate al gruppo e non si propagano oltre.

Infatti si assegna subito all'apostrofo il valore di lettera minuscola:

```
\lccode{'}'=\'
```

Le lettere `lc` stanno per le iniziali di “lower case”; la scrittura `'` si riferisce al codice numerico del segno `'`; con l'assegnazione indicata nel file si attribuisce come codice di lettera minuscola all'apostrofo il suo stesso codice. L'apostrofo continua a essere una “non lettera” con codice di categoria pari a 12, ma con il codice di minuscola non negativo esso è considerato da T_EX parte di quelle stringhe che lui considera parole.

Seguono poi come argomento di `\pattern` tutti i pattern per l'italiano: si comincia dall'apostrofo

TABELLA 2: I pattern per la lingua italiana

```

%% file ithyph.tex
%
{\lccode'\='\' % Apostrophe has its own lccode
% so that it is treated as a letter
%> 1998/04/14 inserted grouping
%
\patterns{ % After the Garzanti dictionary:
.a3p2n % a-pnea, a-pnoi-co,...
.anti1 .anti3m2n
.bio1
.ca4p3s
.circu2m1
.di2s3cine
.fran2k3
.free3
.narco1
.opto1
.orto3p2
.para1
.poli3p2
.prel1
.p2s
.sha2re3
.tran2s3c .tran2s3d .tran2s3f .tran2s3l .tran2s3n
.tran2s3p .tran2s3r .tran2s3t
.su2b3lu .su2b3r
.wa2g3n
.wel2t1
alia alie alio aliu aluo alya 2at.
eliu e2w
olia o1ie o1io oliu
'2
1b 2bb 2bc 2bd 2bf 2bm 2bn 2bp 2bs 2bt 2bv b2l b2r 2b. 2b'. 2b''
1c 2cb 2cc 2cd 2cf 2ck 2cm 2cn 2cq 2cs 2ct 2cz 2chh c2h 2chb ch2r 2chn
c2l c2r 2c. 2c'. 2c'' .c2
1d 2db 2dd 2dg 2dl 2dm 2dn 2dp d2r 2ds 2dt 2dv 2dw 2d. 2d'. 2d'' .d2
1f 2fb 2fg 2ff 2fn f2l f2r 2fs 2ft 2f. 2f'. 2f''
1g 2gb 2gd 2gf 2gg g2h g2l 2gm g2n 2gp g2r 2gs 2gt 2gv 2gw 2gz 2gh2t
2g. 2g'. 2g''
1h 2hb 2hd 2hh hi3p2n h2l 2hm 2hn 2hr 2hv 2h. 2h'. 2h''
1j 2j. 2j'. 2j''
1k 2kg 2kf k2h 2kk k2l 2km k2r 2ks 2kt 2k. 2k'. 2k''
1l 2lb 2lc 2ld 2l3f2 2lg 12h 2lk 2ll 2lm 2ln 2lp 2lq 2lr 2ls 2lt 2lv 2lw
2lz 2l. 2l'. 2l''
1m 2mb 2mc 2mf 2ml 2mm 2mn 2mp 2mq 2mr 2ms 2mt 2mv 2mw 2m. 2m'. 2m''
1n 2nb 2nc 2nd 2nf 2ng 2nk 2nl 2nm 2nn 2np 2nq 2nr 2ns 2nt 2nv 2nz n2g3n
2nheit. 2n. 2n'. 2n''
1p 2pd p2h p2l 2pn 3p2ne 2pp p2r 2ps 3p2sic 2pt 2pz 2p. 2p'. 2p''
1q 2qq 2q. 2q'. 2q''
1r 2rb 2rc 2rd 2rf r2h 2rg 2rk 2rl 2rm 2rn 2rp 2rq 2rr 2rs 2rt rt2s3 2rv 2rx
2rw 2rz 2r. 2r'. 2r''
1s2 2shm 2s3s s4s3m 2s3p2n 2stb 2stc 2std 2stf 2stg 2stm 2stn 2stp 2sts 2stt 2stv 2sz 4s.
4s'. 4s''
1t 2tb 2tc 2td 2tf 2tg t2h t2l 2tm 2tn 2tp t2r 2ts 3t2sch 2tt 2tv 2tw t2z
2tzk 2tzs 2t. 2t'. 2t''
1v 2vc v2l v2r 2vv 2v. 2v'. 2v''
1w w2h wa2r 2w1y 2w. 2w'. 2w''
1x 2xt 2xw 2x. 2x'. 2x''
y1ou y1i
1z 2zb 2zd 2zl 2zn 2zp 2zt 2zs 2zv 2zz 2z. 2z'. 2z'' .z2
}} % Pattern end
\endinput

```

isolato, ma guardando più avanti si ritrova l'apostrofo alla fine delle parole, dove svolgerebbe la funzione di virgolette; i pattern con gli apostrofi in funzione di virgolette sono necessari per evitare che le virgolette vadano a capo da sole, lasciando la parola italiana alla fine della riga precedente immediatamente seguita dalla lineetta di cesura. È chiaro che questo sarebbe un modo ridicolo di gestire la cosa, mentre trattando l'apostrofo come una lettera comune e inserendo i codici pari necessari per interdire le cesure si risolve il problema in modo semplicissimo.

La lista contiene poi le divisioni di prefissi e prefissoidi; la grammatica consente di eseguire la cesura in modo fonetico per qualunque parola, tuttavia divisioni come “su-blu-na-re”, “tran-sna-zio-na-le”, che la grammatica consente, sarebbero difficili da leggere. Per altro la grammatica consente anche la divisione etimologica, quindi le divisioni indotte dai pattern specificati saranno “sub-lu-na-re” e “trans-na-zio-na-le”.

Si nota l'assenza di qualunque pattern che consideri vocali isolate o considerate a coppie; per le vocali isolate bastano i valori nulli di default, mentre per i dittonghi veri o per le coppie di vocali si sono lasciati i codici nulli (e quindi non sono stati indicati) per evitare di spezzare fra vocali. È vero che “au” può formare un dittongo e quindi sarebbe inseparabile, tuttavia esso è un dittongo solo se la vocale ‘u’ non è tonica; ma T_EX non conosce la pronuncia dell'italiano e quindi è meglio lasciare sempre la coppia “au” indivisa. Si perderà qualche possibile cesura, ma per il lettore la lettura sarà certamente agevolata: in questo modo egli troverà la divisione “bau-le” invece di “ba-u-le” e non si troverà mai la parola divisa dopo la ‘a’.

Compaiono invece i trittonghi, i gruppi di tre vocali che si pronunciano con un'unica emissione di voce: avremo così, per esempio, le divisioni “aiuo-la”⁴ e “ma-ieu-ti-ca”.

Con questa decisione non ci sono problemi a che i pattern trattino correttamente anche le vocali accentate sia in posizione terminale, come nelle parole tronche, sia quando ricevono una accentazione facoltativa, come quando si vogliono distinguere parole altrimenti omografe quali “séguito” e “seguito”⁵.

Seguono poi i pattern che riguardano le consonanti, per le quali bisogna distinguere le liquide e le nasali dalle altre; bisogna gestire correttamente la ‘s’ impura, caratteristica della sola lingua

4. Di fatto solamente “aiuo-la” perché la prima sillaba grammaticale è troppo corta; tuttavia se la parola è preceduta dall'articolo o da una preposizione articolata, il trittongo riappare: la sillabazione di “l'aiuola” diventa “l'a-aiuo-la”.

5. La grammatica consente l'accento facoltativo solo sulle parole sdrucciole o con l'accento ancora più arretrato. Non consente l'accento sulle parole piane, perché quella è la posizione di default, tuttavia consente l'accentazione delle parole piane per distinguerne il tono come in “cólto” e “còlto”.

italiana fra le varie lingue romanze. Bisogna trattare la ‘w’ e la ‘y’ in modo ragionevole; bisogna prevedere gruppi di consonanti non presenti nelle parole italiane in senso stretto ma derivate da radici straniere. In questo modo, sebbene la grammatica non dica niente al riguardo, vengono divise correttamente parole come maxwelliano, newyorke-se, leishmaniosi, lewisite, wahhabita... ma anche parole italiane dalla grafia insolita, come Santhià o Thiene.

5 Perché T_EX talvolta non divide in sillabe?

Talvolta T_EX si “rifiuta” di eseguire le necessarie cesure pur non essendone impedito dal tipo di font usato; si ricorda, infatti, che il font a spaziatura fissa non consente la divisione in sillabe.

I motivi possono essere diversi, ma spesso si tratta di errori di composizione; per esempio ci si scorda di lasciare lo spazio fra una parola e la susseguente o la precedente espressione matematica in linea: se il segno di dollaro è adiacente alla parola, T_EX non divide in sillabe, ma quasi sempre la mancanza dello spazio è un errore di battitura, e quindi la “colpa” non è di T_EX.

Ma ci sono tre situazioni subdole che il compositore stenta a riconoscere. Una situazione si ha quando la parola segue un cambiamento di font, specialmente se eseguito mediante una dichiarazione. Un'altra situazione si ha quando alla parola è “appesa” una nota, o meglio, il richiamo di nota; quest'ultimo infatti è composto dal comando `\footnote` o da `\footnotemark` alzando il contrassegno mediante il meccanismo degli esponenti matematici oppure mediante la manipolazione di “scatole”. La terza situazione si ha quando la parola è composta mediante elementi separati dal trattino.

In quest'ultimo caso T_EX non divide in sillabe né la prima né la seconda parola legate dal trattino, ma divide solo in corrispondenza del trattino. Questo comportamento è ragionevole, ma talvolta, con parole lunghe, sarebbe desiderabile consentire la cesura anche nel primo o nel secondo moncone. Siccome il problema si incontra in quasi tutte le lingue, la prima cosa da fare è leggere il file di testo che descrive quanto si è realizzato per quella particolare lingua; il pacchetto `babel` è corredato da un file per ogni lingua che riesce a gestire.

Per l'italiano basta fare precedere il doppio apice al trattino, scrivendo, quindi, “- al posto del semplice trattino, e la cesura viene abilitata in qualunque punto della parola composta.

Negli altri casi la soluzione è semplice ma appare come un “dirty trick”, per chiamarlo col nome che Knuth stesso ha attribuito a questi “sporchi trucchi” dedicando loro addirittura l'appendice D del suo T_EXbook, KNUTH (1984).

Basta inserire un grumetto di colla di larghezza

nulla prima o, rispettivamente, dopo l'oggetto che impedisce la sillabazione. Nello stesso tempo si desidera che T_EX non vada a capo in corrispondenza di questo grumetto di colla, specialmente nel caso che esso sia collocato fra la parola e il richiamo di nota.

La soluzione migliore, secondo lo scrivente, è quello di usare una breve macro come la seguente

```
\newcommand*{\hz}{%
  \nobreak\hskip 0pt\relax}
```

così da usarla in situazioni del tipo

```
... capoverso\hz\footnote{Si ricorda ...}
```

Non è il caso di inserire `\hz` prima di ogni nota, ma se si dovesse trovare non giustificata qualche parola per la presenza del richiamo di nota, allora in seconda battuta si può inserire `\hz` solo prima di quella nota.

Analogamente se una parola non risultasse giustificata dopo un cambiamento di font, basta premettere `\hz` alla parola (lasciando uno spazio subito dopo, spazio che T_EX usa solo per capire dove finisce la macro) e il problema è risolto. Si noti che si ha un cambiamento di font implicito all'inizio di ogni capoverso e all'inizio del testo introdotto da `\item` in una lista. Normalmente la prima parola di un capoverso o di un `\item` non richiede la sillabazione, ma se si sta componendo in colonne strette e/o la parola è molto lunga, allora anche in quel caso conviene cominciare il capoverso o l'`\item` con `\hz`.

Non è opportuno usare il comando `\allowhyphens` del pacchetto `babel`, anche se scrivendo in italiano quel pacchetto viene

sempre richiamato, perché esso *non* inserisce il grumetto di colla quando è in vigore la codifica T1, quella che si dovrebbe sempre usare quando si compone in italiano.

6 Conclusione

Questo breve tutorial in merito alla sillabazione dovrebbe essere riuscito a spiegare sia come fa T_EX ad eseguire la cesura delle parole in fin di riga, sia l'importanza di includere i pattern di sillabazione di una data lingua nel file di formato del programma che si intende usare, sia, infine, che cosa fare per consentire l'esecuzione della cesura anche quando T_EX ne è impedito dalle sue regole stringenti per isolare la sequenza di lettere che costituisce una parola.

Riferimenti bibliografici

- GREGORIO, E. (2006). «Codici di categoria». *ArsT_EXnica – Rivista italiana di T_EX e L^AT_EX*, (2).
- KNUTH, D. E. (1984). *The T_EXbook*. Addison Wesley Publ. Co., Reading, Mass.
- LIANG, F. (1983). *Word Hy-phen-a-tion by Computer*. Tesi di Laurea, Stanford University. La tesi originale è stata scandita e resa disponibile nel sito del T_EX Users Group; <http://www.tug.org/docs/liang/liang-thesis-hires.pdf>.

▷ Claudio Beccari