TeX è un linguaggio di programmazione per la composizione tipografica. La «compilazione» di un sorgente TeX produce un file in formato finale (DVI o PDF) per la stampa.

L'importanza di questo linguaggio richiede anche la conoscenza della pronuncia corretta del suo nome: «t-e-k». Infatti, il creatore di questo linguaggio, D.E. Knuth, voleva fare riferimento alle lettere greche maiuscole «TEX», che così vanno pronunciate.

Dal momento che le istruzioni TeX utilizzano spesso le parentesi graffe, nei modelli sintattici queste vanno intese letteralmente, come parte dell'istruzione rappresentata.

49.1 La distribuzione Unix di TeX

TeX è un linguaggio di programmazione per l'editoria elettronica. Come nei linguaggi di programmazione comuni è possibile realizzare procedure o funzioni, con TeX è possibile costruire delle macro. Nel tempo sono stati realizzati diversi pacchetti standard di macro per TeX; per esempio LaTeX e AmS-TeX.

Semplificando le cose, una distribuzione TeX è un insieme composto da un compilatore TeX (ma forse è più appropriato il termine «compositore»), una serie di file contenenti le informazioni necessarie a produrre i caratteri da stampa e alcuni pacchetti di macro (di solito si tratta almeno di LaTeX).

49.1.1 Collocazione

È importante chiarire che non esiste un modo standard di installare una distribuzione TeX e ogni distribuzione GNU tende a collocarla dove ritiene più opportuno. Il blocco principale di una distribuzione TeX dovrebbe trovarsi in una gerarchia che può inserirsi al di sotto di '`/usr/lib/`' o '`/usr/share/`'. A titolo di esempio, viene mostrato un elenco di alcune di queste possibilità.

'`/usr/lib/teTeX/texmf/`'

'`/usr/lib/texmf/texmf/`'

'`/usr/share/teTeX/texmf/`'

'`/usr/share/texmf/`'

Negli esempi che vengono mostrati, quando si fa riferimento a questa directory, si indicano solo percorsi relativi a iniziare da '`texmf/`'. La sigla «texmf» sta per *TeX and more*, oppure per *TeX and friends*.

49.1.2 Configurazione

Di fronte alla complicazione di una distribuzione TeX, potrebbe sembrare assurda l'idea di metterci le mani, pensando addirittura di modificare le impostazioni generali di TeX. Tuttavia, quando si maneggiano documenti eccezionalmente voluminosi, potrebbe essere necessario modificare anche ciò che non è stato pensato per esserlo.

Alla fine della composizione di un documento TeX, si può leggere nel file delle registrazioni generato, un rapporto delle risorse utilizzate durante l'elaborazione. Si osservi l'esempio.

```
Here is how much of TeX's memory you used:
 2418 strings out of 25906
 45132 string characters out of 446921
 109255 words of memory out of 263001
 5196 multiletter control sequences out of 10000+0
 106774 words of font info for 69 fonts, ↩
↪out of 200000 for 1000
 15 hyphenation exceptions out of 1000
 33i,12n,21p,2494b,1259s stack positions ↩
↪out of 300i,100n,500p,30000b,4000s

Output written on texput.dvi (1844 pages, 7563800 bytes).
```

Questo è proprio il caso di un documento enorme (1844 pagine), ma prima di tale informazione appaiono dei valori, dove alternativamente si vede quanto di una data risorsa è stato usato e quanto di questa sarebbe stato invece disponibile. Se per qualche ragione si esaurisce una di queste risorse, l'elaborazione si interrompe con una segnalazione di errore che indica quale limite è stato superato.

Se succede, si può provare a mettere mano al file di configurazione della distribuzione di TeX che dovrebbe essere '`texmf/web2c/texmf.cnf`'. La prima volta, non è tanto facile capire il senso delle direttive che questo contiene, ma con un po' di tentativi si dovrebbe riuscire a risolvere il problema.

Prima di tutto si può osservare che, seguendo lo stile generale di TeX, i commenti sono introdotti dal simbolo di percentuale ('`%`'). Nella prima parte del file sono annotati i percorsi dei vari componenti della distribuzione.

```
% Part 1: Search paths and directories.

% The main tree, which must be mentioned in $TEXMF, below:
TEXMFMAIN = /usr/share/texmf

% A place for local additions to a "standard" texmf tree.
% For example:
%   TEXMFLOCAL = /usr/share/texmf.local

% A place where texconfig stores modifications (instead of
% the TEXMFMAIN tree). texconfig relies on the name, so
% don't change it.
%   TEXMF_CNF = $TEXMF.cnf

% User texmf trees can be catered for like this...
%   HOMETEXMF = $HOME/texmf
```

La lettura di questa parte può rivelare delle informazioni importanti riguardo la propria distribuzione TeX. Più avanti inizia una parte più delicata: quella che definisce le dimensioni degli array utilizzati da TeX, dimensioni che di conseguenza rappresentano i limiti a cui si accennava all'inizio di questa sezione.

```
% Part 3: Array and other sizes for TeX (and Metafont and
% MetaPost).
…
% Max number of characters in all strings, including all
% error messages, help texts, font names, control sequences.
%  These values apply to TeX and MP.
pool_size.context = 500000
pool_size.cont-en = 500000
pool_size.cont-nl = 500000
pool_size.cont-de = 500000
%pool_size = 125000
pool_size = 500000
```

In questa parte, il valore più importante è quello di '**`pool_size`**', perché può creare problemi soprattutto a pdfTeX. Nell'esempio si vede che è stato quadruplicato.

Alcune modifiche non possono essere prese in considerazione senza un'elaborazione successiva del file. In generale, al termine delle modifiche è bene dare il comando seguente:

```
# texconfig init [Invio]
```


A parte il caso particolare dell'utilizzo appena mostrato, '**`texconfig`**' è un programma interattivo predisposto per configurare gli elementi essenziali della distribuzione TeX. Si avvia semplicemente, senza bisogno di argomenti. La figura 49.4 mostra il menù principale di '**`texconfig`**'.

```
# texconfig [Invio]
```

Figura 49.4. Il menù principale di '**`texconfig`**'.

```
.------------------ teTeX setup utility -------------------.
|                                                           |
|  Hint: all output of external commands (e.g. tex) is      |
|  logged into a file. You can look at this file using      |
|  LOG. If cursor keys make trouble, you may have more      |
|  luck with +/- and TAB.                                   |
| .-------------------------------------------------------. |
| |      EXIT    exit                                     | |
| |      PREF    personal preferences                     | |
| |      CONF    show configuration                       | |
| |      REHASH  rebuild ls-R database                    | |
| |      HYPHEN  hyphenation table (tex/latex)            | |
| |      MODE    default mode (xdvi/dvips/mf)             | |
| |      XDVI    xdvi configuration                       | |
| |      DVIPS   dvips configuration                      | |
| |      FONT    directories for font creation            | |
| |      DOC     rebuild html documentation               | |
| |      FAQ     frequently asked questions + answers     | |
| |      LOG     view logfile                             | |
|-------------------------------------------------------------|
|                  <  OK  >      <Cancel>                   |
'-------------------------------------------------------------'
```

La funzione indicata con la sigla *PREF* serve solo a modificare il comportamento di '**`texconfig`**', permettendo in particolare di selezionare un programma per la modifica del testo alternativo a quello predefinito.

La funzione *CONF* permette di mostrare la configurazione, consistente nella definizione delle directory contenenti i vari componenti della distribuzione TeX.

La funzione *REHASH* permette di ricostruire il file '`texmf/ls-R`', utilizzato per agevolare la ricerca dei componenti installati ad alcuni programmi della distribuzione. In generale, si ricostruisce questo file quando si aggiunte o si toglie qualche file (per esempio i file dei tipi di carattere).

La funzione *HYPHEN* è molto importante, perché permette di stabilire le lingue per cui attivare la suddivisione in sillabe del testo. Selezionando questa funzione si ottiene l'avvio del programma per la modifica di file di testo (presumibilmente VI) con il file '`texmf/tex/generic/config/language.dat`'. Questo file può essere modificato, quindi, dopo averlo salvato, vengono avviate automaticamente le procedure necessarie ad attivare in pratica le scelte fatte.

Di solito, si tratta di commentare le righe che fanno riferimento a linguaggi che non si vogliono gestire e di togliere il commento dalla direttiva di attivazione della sillabazione per la lingua italiana.

```
% File    : language.dat
% Purpose : specify which hypenation patterns to load
%           while running iniTeX

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CAUTION: the first language will be the default if no
%          style-file (e.g. german.sty) is used.
% Since version 3.0 of TeX, hyphenation patterns for
% multiple languages are possible. Unless you know what you
% are doing, please let the american english patterns be the
% first ones. The babel system allows you to easily change
% the active language for your texts. For more information,
% have a look to the documentation in
% texmf/doc/generic/babel.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% The hyphenation pattern files are in the directory:
%       texmf/tex/generic/hyphen
```

```
% The US-english patterns should be loaded *always* and as
% *first* ones.
american ushyph1.tex %

% Let us define USenglish as an alias for american:
=USenglish %

% UK english, TWO LINES!
%british  ukhyph.tex %
%=UKenglish %

% english should always be defined. Either an alias for
% american or british.
=english %

% French, TWO lines!
french          frhyph.tex frhyphex.tex %
=patois %

german          ghyph31.tex %

% The following languages are disabled by default.
% Uncomment, what you need.
%bahasa         inhyph.tex %
%catalan        cahyph.tex %
%croatian       hrhyph.tex %
%czech          czhyph2e.tex %
%danish         dkhyphen.tex %
%dutch          nehyph2.tex %
%finnish        fihyph.tex %
%galician       gahyph.tex %
italian         ithyph.tex %
%magyar         huhyph.tex %
%norsk          nohyph.tex %
%polish         plhyph.tex %
%portuges       pthyph.tex %
%romanian       rohyphen.tex %
%russian        ruhyph.tex %
%serbocroatian  shhyphl.tex %
%slovene        sihyph22.tex %
%spanish        sphyph.tex %
%swedish        sehyph.tex %
%turkish        trhyph.tex %

% A "language" without hyphenation:
nohyphenation zerohyph.tex %
```

Al termine dell'elaborazione si può verificare nel file '`texmf/web2c/latex.log`' la presenza delle righe che dimostrano l'abilitazione della sillabazione per le lingue selezionate nel file di configurazione. In questo caso particolare, la lingua italiana corrisponde al linguaggio numero tre.

```
…
\l@american=\language0
…
\l@USenglish =\language0
\l@english =\language0
\l@french=\language1
…
\l@patois =\language1
\l@german=\language2
…
\l@italian=\language3
…
\l@nohyphenation=\language4
```

La funzione *MODE* permette di predisporre alcuni programmi per la risoluzione della propria stampante. Ciò si ottiene semplicemente selezionando il nome di una stampante che dovrebbe corrispondere, o essere molto simile alla propria.

La funzione *XDVI* permette di configurare Xdvi (sezione 26.3), il programma di visualizzazione dei file DVI, in modo da stabilire il formato del foglio che si utilizza. Basta scorrere un elenco e confermare.

La funzione *DVIPS* permette di configurare Dvips (sezione 26.3), il programma in grado di convertire file DVI in PostScript. Come per

Xdvi, la cosa più importante da stabilire è il formato della carta, ma può anche essere indicata la stampante, o il comando di stampa da utilizzare quando Dvips viene usato per inviare direttamente un file alla stampa.

L'ultima funzione importante è *FONT* che permette di regolare i permessi di accesso alle directory che contengono i tipi di carattere e anche di configurare altre caratteristiche di questi file.

49.1.3 Funzionamento fondamentale di TeX

TeX utilizza un sorgente che si distingue perché di solito il suo nome finisce con l'estensione '`.tex`'; durante il processo di composizione genera un rapporto sull'elaborazione in un file con l'estensione '`.log`' e produce un file finale in formato DVI, con estensione '`.dvi`'. Successivamente, i file DVI vengono convertiti normalmente in PostScript attraverso il programma Dvips.

Eventualmente, è disponibile anche pdfTeX, con cui, invece di una composizione in formato DVI, si ottiene un file PDF senza passaggi intermedi.

Se si suppone che il file '`primo.tex`' contenga il testo seguente

```
Ciao a tutti.

Questo \'e il mio primo documento scritto con il
linguaggio \TeX.
\bye
```

per ottenere la composizione in formato DVI è sufficiente il comando

```
$ tex primo.tex [Invio]

 This is TeX, Version 3.14159 (Web2C 7.3.1)
 (primo.tex [1] )
 Output written on primo.dvi (1 page, 328 bytes).
 Transcript written on primo.log.
```

mentre per ottenere la composizione in formato PDF è sufficiente il comando

```
$ pdftex primo.tex [Invio]

 This is pdfTeX, Version 3.14159-13d (Web2C 7.3.1)
 (primo.tex[/usr/share/texmf/pdftex/config/pdftex.cfg]
 Babel <v3.6x> and hyphenation patterns for american,
    italian, nohyphenation, loaded.
 [1[/usr/share/texmf/dvips/config/pdftex.map]] )<cmr10.pfb>
 Output written on primo.pdf (1 page, 9807 bytes).
 Transcript written on primo.log.
```

Nel primo caso si ottiene il file '`primo.dvi`', mentre nel secondo si ha il file '`primo.pdf`'. Eventualmente, per convertire il file DVI in PostScript, è sufficiente usare Dvips nel modo seguente:

```
$ dvips -o primo.ps primo.dvi [Invio]

 This is dvips(k) 5.86 Copyright 1999 Radical Eye Software
   (www.radicaleye.com)
 ' TeX output 2001.08.30:0835' -> primo.ps
 <texc.pro>. [1]
```

Ecco quello che si ottiene:

Ciao a tutti.
Questo è il mio primo documento scritto con il linguaggio TeX.

Si può anche usare una versione estesa di TeX, e-TeX e pdfeTeX, corrispondenti agli eseguibili '**`etex`**' e '**`pdfetex`**', che in questo caso si comportano nello stesso modo:

```
$ etex primo.tex [Invio]

 This is e-TeX, Version 3.14159-2.1 (Web2C 7.3.1)
 entering extended mode
 (primo.tex [1] )
 Output written on primo.dvi (1 page, 328 bytes).
 Transcript written on primo.log.
```

oppure

```
$ pdfetex primo.tex [Invio]
```

```
This is pdfeTeX, Version 3.14159-13d-2.1 (Web2C 7.3.1)
entering extended mode
(primo.tex[/usr/share/texmf/pdftex/config/pdftex.cfg] [1[
/usr/share/texmf/dvips/config/pdftex.map]] )<cmr10.pfb>
Output written on primo.pdf (1 page, 9807 bytes).
Transcript written on primo.log.
```

Gli eseguibili '**tex**', '**pdftex**' '**etex**' e '**pdfetex**' sono indipendenti, mentre attorno a loro si presentano altrettante serie di collegamenti simbolici:

```
virtex     -> tex
initex     -> tex
latex      -> tex
amstex     -> tex
evirtex    -> etex
einitex    -> etex
elatex     -> etex
pdfevirtex -> pdfetex
pdfeinitex -> pdfetex
pdfelatex  -> pdfetex
pdfvirtex  -> pdftex
pdfinitex  -> pdftex
pdflatex   -> pdftex
```

A seconda del nome usato per avviare uno stesso eseguibile, si può ottenere un comportamento differente. Nel caso di '**virtex**' che è un collegamento a '**tex**', si fa riferimento implicitamente al ***formato*** '**plain**', corrispondente alle dichiarazioni contenute nei file della directory '`texmf/tex/plain/`', così come '**pdfvirtex**' fa riferimento alla directory '`texmf/pdftex/plain/`', ecc. Purtroppo, le cose non sono così semplici in generale, perché le convenzioni non sono perfettamente omogenee; tuttavia, vale la pena di tenere presente che i nomi del tipo '**[pdf][e]virtex**' sono equivalenti ai nomi del tipo '**[pdf][e]tex**'.

I nomi del tipo '**[pdf][e]initex**' fanno riferimento al linguaggio TeX senza la dichiarazione di alcuna macro e sono equivalenti all'uso degli eseguibili del tipo '**[pdf][e]tex**' con l'aggiunta dell'opzione '**--ini**'. In pratica, per usare '**[pdf][e]initex**', oppure '**[pdf][e]tex --ini**', occorre modificare l'esempio già visto nel modo seguente:

```
\input plain
Ciao a tutti.

Questo \'e il mio primo documento scritto con il
linguaggio \TeX.
\bye
```

Questa spiegazione viene data solo per chiarire un po' il funzionamento di TeX e il significato di tutti i collegamenti simbolici che gli stanno intorno. L'uso dell'istruzione '**\input plain**' nel sorgente non funziona sempre come ci si aspetterebbe leggendo queste indicazioni; in pratica, una volta capito il senso della cosa, non va usata affatto.

In modo analogo a quanto visto fino a questo punto, quando si fa riferimento a un collegamento del tipo '**[pdf][e]latex**', è come usare un eseguibile del tipo '**[pdf][e]initex**', oppure '**[pdf][e]tex --ini**', iniziando il sorgente TeX con la riga seguente:

```
\input latex.ini
```

Resta il fatto che il comando '**\input latex.ini**' non rappresenta necessariamente uno standard e quello che conta è sapere solo che i collegamenti '**[pdf][e]latex**' richiamano in qualche modo il formato '**latex**'. In altri termini, i collegamenti '**[pdf][e]latex**' fanno riferimento implicitamente alle macro di LaTeX.

Per completare questa sezione, viene mostrato un esempio di un sorgente LaTeX, ovvero un sorgente TeX fatto per le macro LaTeX. Si fa riferimento al nome ipotetico '`secondo.tex`':


```
\documentclass{article}
\begin{document}
Ciao a tutti.

Questo \'e il mio primo documento scritto con il
linguaggio \LaTeX .
\end{document}
```

Per comporre correttamente questo file, occorre un comando del tipo:

```
$ [pdf][e]latex secondo.tex [Invio]
```

A seconda dei casi si ottiene il file '`secondo.dvi`', oppure '`secondo.pdf`'.

49.1.4 Interazione con TeX

La composizione di un documento scritto con il linguaggio TeX può avvenire anche con qualche forma di interazione. Se si avvia uno degli eseguibili '**[pdf][e]tex**' senza argomenti, si ottiene un invito a inserire il nome del file, attraverso l'indicazione di due asterischi:

```
$ tex [Invio]

 This is TeX, Version 3.14159 (Web2C 7.3.1)
 **
```

Si può inserire così il percorso del file, omettendo eventualmente l'estensione se questa corrisponde a '`.tex`':

```
**terzo.tex [Invio]

 (terzo.tex [1] )
 Output written on terzo.dvi (1 page, 488 bytes).
 Transcript written on terzo.log.
```

Supponendo di avere scritto un file, denominato '`quarto.tex`', in cui non appare l'istruzione '**\bye**' finale, come nel testo seguente, TeX si ferma in attesa di istruzioni, mostrando un invito ridotto a un solo asterisco:

```
Ciao a tutti.

Questo \'e il mio quarto documento scritto con il
linguaggio \TeX, dove non appare l'istruzione
$\backslash$bye alla fine del file.
```

```
$ tex quarto.tex [Invio]

 This is TeX, Version 3.14159 (Web2C 7.3.1)
 (quarto.tex)
 *
```

Naturalmente, se si è in grado di farlo, si può aggiungere anche altro testo:

```
*Saluti! [Invio]

*\bye [Invio]

 [1]
 Output written on quarto.dvi (1 page, 448 bytes).
 Transcript written on quarto.log.
```

Ciao a tutti.
Questo è il mio quarto documento scritto con il linguaggio TEX, dove non appare l'istruzione \bye alla fine del file.
Saluti!

Di solito si evita di interagire con TeX, tuttavia si può essere costretti dal presentarsi di un errore durante la compilazione del sorgente. Per la precisione, il livello di interazione di TeX può essere regolato attraverso delle istruzioni speciali, come descritto nella tabella 49.24. In condizioni normali, il funzionamento avviene in modalità '**errorstopmode**', corrispondente all'istruzione '**\errorstopmode**', in cui TeX si ferma in attesa di indicazioni per qualunque errore si presenti.

Tabella 49.24. Istruzioni per il controllo della modalità di funzionamento interattiva.

Istruzione TeX	Modalità di funzionamento
`\errorstopmode`	La composizione viene sospesa per qualunque errore.
`\scrollmode`	La composizione viene sospesa solo per gli errori più importanti.

Istruzione TeX	Modalità di funzionamento
`\nonstopmode`	La composizione viene sospesa solo in presenza di errori gravissimi.
`\batchmode`	La composizione viene sospesa solo in presenza di errori gravissimi e non si mostrano informazioni sullo schermo.

Per esempio, il file '`quinto.tex`' che contiene il testo seguente, usa erroneamente l'istruzione '**`\tex`**' al posto di '**`\TeX`**':

```
Ciao a tutti.

Questo \‘e il mio quinto documento scritto con il
linguaggio \tex, in cui si provoca volutamente un errore.
\bye
```

```
$ tex quinto.tex [Invio]

This is TeX, Version 3.14159 (Web2C 7.3.1)
(quinto.tex
! Undefined control sequence.
l.3 ...to documento scritto con il linguaggio \tex
                                                , in
?
```

Viene dichiarato sinteticamente il tipo di errore individuato, che in questo caso corrisponde a

```
! Undefined control sequence.
```

ovvero una sequenza di controllo indefinita. Nella riga successiva si indica il numero della riga in cui appare l'errore ('**`l.3`**' sta per *line* 3) con il pezzo di testo che arriva fino all'errore, mentre il pezzo successivo appare staccato, nella riga successiva.

```
l.3 ...to documento scritto con il linguaggio \tex
                                                , in
```

In pratica, secondo TeX l'errore è riferito esattamente alla stringa '**`\tex`**'. Sotto appare un invito composto da un punto interrogativo, con il quale TeX attende un'azione dell'utente. Si può rispondere con un altro punto interrogativo per avere l'elenco delle possibilità:

```
??[Invio]

Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
?
```

La tabella 49.30 descrive brevemente il significato e l'uso dei comandi disponibili, mostrando anche la modalità corrispondente quando una scelta coincide con la richiesta di cambiamento di questa.

Tabella 49.30. Interazione in presenza di un errore.

Tastiera	Istruzione TeX corrispondente	Effetto
[*h*][*Invio*]		Mostra la spiegazione del motivo dell'interruzione.
[*i*][*Invio*]		Richiede di inserire una correzione.
[*e*][*Invio*]		Avvia un programma per la modifica del file sorgente.
[*x*][*Invio*]		Arresta la composizione.
[*s*][*Invio*]	`\scrollmode`	Non si ferma più in presenza di piccoli errori.
[*r*][*Invio*]	`\nonstopmode`	Non si ferma più per alcun errore.
[*q*][*Invio*]	`\batchmode`	Non si ferma più per alcun errore e non mostra alcuna informazione.
[*Invio*]		Cerca di rimediare all'errore e continua la composizione.
[***n***][*Invio*]		Cerca di rimediare automaticamente ai prossimi ***n*** errori.

Nel caso dell'errore mostrato, si vuole provare a capire meglio di cosa si tratta, attraverso il comando [*h*][*Invio*]:


```
?h [Invio]

  The control sequence at the end of the top line
  of your error message was never \def’ed. If you have
  misspelled it (e.g., ‘\hobx’), type ‘I’ and the correct
  spelling (e.g., ‘I\hbox’). Otherwise just continue,
  and I’ll forget about whatever was undefined.
```

Dal momento che si può correggere facilmente l'errore, si richiede di poter inserire del testo sostitutivo:

```
?i [Invio]

insert>\TeX [Invio]

  [1] )
  Output written on quinto.dvi (1 page, 376 bytes).
  Transcript written on quinto.log.
```

Con il comando [*e*][*Invio*] si può avviare un programma per la modifica del file sorgente. Di solito si tratta di VI, ma si può intervenire nella variabile di ambiente ***TEXEDIT*** indicando il nome di un altro programma, usando le metavariabili '**`%d`**' e '**`%s`**' per indicare rispettivamente la riga contenente l'errore e il nome del file. Per esempio, rimanendo nel caso di VI, è possibile usare la sintassi seguente per raggiungere la riga ***n***-esima del file indicato:

```
vi -c n file
```

In questo modo, si dovrebbe assegnare alla variabile di ambiente ***TEXEDIT*** la stringa '**`vi -c %d %s`**':

```
$ TEXEDIT="vi -c %d %s" [Invio]

$ export TEXEDIT [Invio]
```

49.2 Elementi essenziali del linguaggio TeX

Un file sorgente TeX è un file di testo normale; per la precisione dovrebbe trattarsi di un file ASCII standard a 7 bit, dove l'interruzione delle righe avviene secondo le regole del proprio sistema operativo.

Nel linguaggio di TeX si distinguono righe bianche o vuote da righe contenenti istruzioni. Nell'ambito delle righe contenenti istruzioni, possono poi apparire dei commenti che si distinguono per essere preceduti dal segno di percentuale ('**`%`**'), terminati dalla fine della riga. Nell'esempio seguente si può osservare che tutte le righe che contengono del testo sono in pratica delle istruzioni, più o meno articolate. Nella prima riga appare anche un commento che poi scompare nella composizione finale.

```
Ciao a tutti. % Ecco, un commento tanto per gradire.

Questo \‘e solo un piccolo esempio per vedere come funziona
il linguaggio di composizione \TeX.
\bye
```

Gli spazi, verticali e orizzontali, hanno un significato, ma generalmente non si sommano. Normalmente si usa una riga vuota o bianca per separare il testo in paragrafi, ma la presenza di più righe bianche o vuote non cambia la distanza tra questi paragrafi nella composizione finale; nello stesso modo, uno spazio orizzontale serve generalmente a separare le parole di una frase, ma la presenza di più spazi orizzontali non cambia la distanza tra le parole.

Eventualmente si può dichiarare espressamente la separazione tra i paragrafi attraverso l'istruzione '**`\par`**', tenendo presente che anche in questo caso, l'uso di più istruzioni del genere non produce una separazione maggiore tra i paragrafi stessi

L'istruzione finale '**`\bye`**' conclude il «programma» TeX e tutto ciò che appare dopo viene ignorato.

49.2.1 Testo letterale, parole e simboli di controllo

Una riga di testo si traduce generalmente nel testo corrispondente nella composizione finale, tenendo conto però che alcuni simboli hanno un significato speciale e si distinguono contesti differenti. La tabella 49.34 elenca i caratteri che hanno significati particolari, con

l'indicazione del modo per ottenere il loro simbolo originale nella composizione.

Tabella 49.34. Elenco dei caratteri che hanno significati particolari.

Carattere speciale	Utilizzo normale	Trasformazione per l'uso letterale
\	Prefisso di una parola di controllo o di un simbolo di controllo.	`$\backslash$`
{	Apre un gruppo.	`$\{$`
}	Chiude un gruppo.	`$\}$`
%	Inizia un commento fino alla fine della riga.	`\%`
&	Tabulazione orizzontale.	`\&`
~	Spazio non interrompibile.	`\~{}`
$	Inizia e conclude un contesto matematico.	`\$`
^	Apice, in un contesto matematico.	`\^{}`
_	Pedice, in un contesto matematico.	`\_{}`
#	Definisce la collocazione di un parametro.	`\#`
<*SP*>	Uno o più spazi vengono ridotti a uno solo.	`\<SP>`
<	In condizioni normali, genera un punto esclamativo rovesciato.	`$<$`
>	In condizioni normali, genera un punto interrogativo rovesciato.	`$>$`

La barra obliqua inversa ('\') viene usata come prefisso per delle ***sequenze di controllo***, nell'ambito delle quali si può distinguere tra ***parole di controllo*** e ***simboli di controllo***. Una parola di controllo è formata esclusivamente da lettere alfabetiche (dalla «a» alla «z», maiuscole e minuscole, escluse le lettere accentate); per esempio, '**\TeX**' è una parola di controllo con cui si ottiene la rappresentazione del nome TeX secondo lo standard stabilito dal suo autore originale. Un simbolo di controllo è invece un solo carattere che non sia una lettera alfabetica; per esempio, '**\'**' è un simbolo di controllo con cui si ottiene l'aggiunta di un accento grave sopra il simbolo successivo.

In base al fatto che le parole di controllo si distinguono perché composte esclusivamente da lettere alfabetiche, si può porre il problema di delimitarle correttamente quando si trovano incorporate in parole che compongono il testo normale. Nell'esempio già mostrato, la parola di controllo '**\TeX**' si individua correttamente perché è seguita da un punto fermo, ovvero un simbolo che non è una lettera alfabetica, ma volendo scrivere un gioco di parole, come «il mio primo TeX-documento», sarebbe necessario usare uno strattagemma. Si osservi l'esempio seguente che si traduce in un errore nella composizione del documento:

```
Ciao a tutti.

Questo \'e il mio primo \TeXdocumento.
\bye
```

TeX cerca di interpretare la parola di controllo '**\TeXdocumento**' e non dovrebbe riuscirci. Per questa ragione, lo spazio che dovesse seguire una parola di controllo viene ignorato; così, diventa più facile inserire queste parole di controllo in parole del testo che si vuole comporre. Ecco l'esempio corretto:

```
Ciao a tutti.

Questo \'e il mio primo \TeX documento.
\bye
```

Tuttavia, rimane da chiarire in che modo inserire veramente uno spazio dopo una parola di controllo. Questo problema viene risolto con l'uso dei raggruppamenti.

49.2.2 Raggruppamenti

In diverse situazioni è utile raggruppare parte delle istruzioni (il testo) all'interno di parentesi graffe ('**{**…**}**'). L'effetto del raggruppamento non si nota nella composizione finale, ma permette di circoscrivere l'effetto di istruzioni particolari.

È ammissibile anche l'uso di raggruppamenti vuoti, '**{}**', che di solito vengono usati per separare parole o simboli di controllo dal testo che segue. Per esempio, scrivendo '**\TeX{}**' si riesce a evitare che lo spazio successivo venga inghiottito. Naturalmente si può usare anche il simbolo di controllo '****<***SP***>' (barra obliqua inversa e spazio) per indicare espressamente uno spazio in quel punto. Nel seguito vengono mostrati diversi esempi che si traducono nella stessa composizione finale.

- Raggruppamento vuoto alla fine della parola di controllo:

```
Il linguaggio di composizione \TeX{} sembra complesso
a prima vista...
\bye
```

- Raggruppamento vuoto per separare gli spazi:

```
Il linguaggio di composizione \TeX {} sembra complesso
a prima vista...
\bye
```

- Raggruppamento per individuare uno spazio:

```
Il linguaggio di composizione \TeX{ }sembra complesso
a prima vista...
\bye
```

- Raggruppamento per individuare uno spazio dopo la fine della parola di controllo:

```
Il linguaggio di composizione \TeX { }sembra complesso
a prima vista...
\bye
```

- Raggruppamento per isolare la parola di controllo:

```
Il linguaggio di composizione {\TeX} sembra complesso
a prima vista...
\bye
```

- Raggruppamento per isolare lo spazio e la parola successiva:

```
Il linguaggio di composizione \TeX { sembra} complesso
a prima vista...
\bye
```

Nella tabella 49.34 sono stati mostrati alcuni simboli di controllo che sono conclusi evidentemente da un raggruppamento vuoto: '**\~{}**', '**\^{}**' e '**_{}**'. In questo caso, il raggruppamento vuoto serve a impedire che la sequenza di controllo produca qualcosa di diverso da ciò che ci si aspetta in quel contesto particolare. Per esempio, '**\^o**' oppure anche '**\^ o**' genera la lettera accentata «ô».

Viene chiarito in seguito che le istruzioni '**\~{}**' e '**\^{}**' sono delle macroistruzioni che servono a generare un accento, utilizzano un parametro, costituito dal carattere o dal gruppo successivo. Nel momento in cui si vogliono rappresentare questi simboli senza abbinarli ad altro, il loro parametro deve essere un gruppo vuoto o lo spazio inteso come carattere ('****<***SP***>'). Pertanto, in alternativa a '**\~{}**' e '**\^{}**' si potrebbe usare la forma '**\~**<***SP***>' e '**\^**<***SP***>'.

Oltre all'uso delle parentesi, è possibile usare in alternativa la coppia di parole di controllo '**\bgroup**' e '**\egroup**', in sostituzione di '**{**' e '**}**' rispettivamente.

49.2.3 Inclusione di file esterni

È possibile articolare un sorgente TeX in più file separati che vengono inclusi con l'istruzione '**\input**':

```
\input nome_file
```

L'inclusione avviene inserendo in quel punto le righe del file indicato. Se il file esterno contiene l'istruzione '**\bye**', la composizione termina senza continuare nel file di partenza.

49.3 Variabili e tipi di dati

TeX ha una gestione molto particolare dei dati. Nelle sezioni seguenti vengono descritti solo i tipi di dati più comuni, ma questo dovrebbe bastare per far comprendere la logica di fondo.

Vale la regola per cui può essere usato solo ciò che è già stato dichiarato; inoltre, il campo di azione di queste variabili può essere controllato attraverso i raggruppamenti con le parentesi graffe.

In generale, per TeX una variabile ha l'aspetto di una parola di controllo a cui si assegna un valore secondo la sintassi seguente:

```
\nome=valore
```

L'espansione di una variabile avviene inserendo la parola di controllo corrispondente nel punto in cui il contesto lo richiede. Tuttavia, se lo scopo è quello di espandere la variabile in modo che appaia nel testo normale, occorre usare un accorgimento, con cui si trasforma il suo contenuto in una stringa. Di solito si usa per questo la parola di controllo '**\the**':

```
\the\nome
```

49.3.1 Stringhe

Per TeX la stringa è ciò che può essere reso tipograficamente; così, un'espressione stringa è ciò che alla fine si trasforma in una stringa nella composizione tipografica. Le variabili di tipo stringa si dichiarano nel modo seguente:

```
\newtoks\nome
```

Si può assegnare una stringa alla variabile nel modo seguente:

```
\nome={espressione_stringa}
```

L'espansione di una variabile stringa non può avvenire inserendo semplicemente la parola di controllo '***nome***' nel testo, perché occorre dichiarare espressamente questa intenzione con la parola di controllo '**\the**'.

```
\newtoks\data
\data={9 settembre 2001}
Treviso, \the\data \par
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla  bla bla bla bla bla bla. \par
Ciao.
```

Treviso, 9 settembre 2001
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.
Ciao.

Nell'esempio precedente, si crea una variabile stringa corrispondente alla parola di controllo '**\data**', a cui si assegna la stringa '**9 settembre 2001**', per poi ottenerne successivamente l'espansione nel testo.

49.3.2 Trasformazione delle stringhe

Le trasformazioni più comuni sulle stringhe sono il cambiamento in maiuscole o minuscole. Ciò si ottiene facilmente con le macro '**\uppercase**' e '**\lowercase**':

```
\uppercase{espressione_stringa_da_trasformare_in_maiuscole}
```

```
\lowercase{espressione_stringa_da_trasformare_in_minuscole}
```

49.3.3 Contatori

La variabile numerica più semplice di TeX è il contatore. Ne esistono due tipi: uno deve essere inizializzato subito, con un valore non negativo, l'altro no:


```
\countdef\nome_contatore=n
```

```
\newcount\nome_contatore
```

Il primo di questi due modelli riguarda il tipo di contatore che deve essere inizializzato in fase di dichiarazione. Il valore di inizializzazione è rappresentato da ***n***.

Per assegnare un valore a una variabile contatore, si usa la forma seguente:

```
\nome=valore
```

È importante sottolineare che l'inizializzazione di un contatore definito attraverso '**\countdef**' potrebbe in pratica non tradursi nell'assegnamento corrispondente alla variabile, pur essendo obbligatorio. Pertanto, conviene poi assegnare nuovamente il valore richiesto.

Si può assegnare un numero espresso usando cifre numeriche, con un segno anteriormente nel caso sia necessario, senza separatore decimale, come nell'esempio seguente, in cui si assegna al contatore '**\conteggio**' il valore -345.

```
\conteggio=-345
```

49.3.4 Lunghezze

Un tipo specifico di variabile numerica è adibita a contenere delle lunghezze. Per TeX, la lunghezza è un'informazione numerica particolare, riconducibile al concetto di variabile a virgola mobile di altri linguaggi. Si dichiara una lunghezza nel modo seguente:

```
\newdimen\nome_lunghezza
```

Una lunghezza è un valore che si può rappresentare in forma costante solo specificando l'unità di misura, composta da due lettere secondo lo schema che appare nella tabella 49.49. Pertanto, un valore che esprime una lunghezza deve avere la forma seguente:

```
[+|-]numero[true]unità_di_misura
```

In particolare, il valore che precede l'unità di misura può contenere una virgola decimale, espressa attraverso il punto ('.').[1] Per esempio, per esprimere una lunghezza di 10 cm, si deve scrivere '**10cm**'.[2] Nell'esempio seguente si assegna alla variabile '**\distanza**' una lunghezza positiva di 4,5 mm:

```
\distanza=4.5mm
```

La parola chiave '**true**' consente di indicare una lunghezza che non può essere ingrandita o ridotta attraverso l'istruzione '**\magnification**', come viene descritto nella sezione 49.11.1. Pertanto, se nell'esempio precedente si vuole indicare una lunghezza positiva corrispondente esattamente a 4,5 mm, in ogni situazione, si deve scrivere così:

```
\distanza=4.5truemm
```

oppure anche in modo più leggibile:

```
\distanza=4.5 true mm
```

La rappresentazione interna delle lunghezze è di un solo tipo; in pratica, TeX converte sempre i valori nell'unità di misura più piccola che è in grado di gestire. L'unità di misura più piccola è definita *scaled point* ed è stata creata appositamente per TeX.

Tabella 49.49. Unità di misura secondo TeX.

Sigla	Denominazione	Corrispondenza	Annotazioni
mm	millimetro		

Sigla	Denominazione	Corrispondenza	Annotazioni
`cm`	centimetro		
`in`	pollice	2,54 cm	
`bp`	*big point*	0,352 777 7 mm 1/72 pollici	Il punto tipografico usato dal linguaggio PostScript.
`pt`	punto	0,351 459 8 mm 1/72,27 pollici	Punto tipografico usato negli Stati Uniti.
`dd`	punto didôt	0,376 065 mm 1/67,54 pollici	Punto tipografico europeo.
`sp`	*scaled point*	1/65 535 punti '**`pt`**'	L'unità di misura più piccola gestibile da TeX.
`pc`	pica	12 punti	
`em`	M	variabile	Quadratone, pari alla larghezza della lettera «M» maiuscola.
`ex`	x	variabile	Altezza della lettera «x» minuscola.
`mu`	*math unit*	18 quadratoni	

Si osservi che TeX non semplifica la tradizione tipografica, consentendo di utilizzare ben tre tipi diversi di punto tipografico. Il punto a cui si è abituati comunemente con i programmi di composizione, è quello corrispondente alla sigla '**`bp`**', ma TeX utilizza in modo predefinito l'unità '**`pt`**', la quale comunque non si discosta di molto.

49.3.5 Lunghezze elastiche

In varie situazioni, TeX è in grado di gestire delle lunghezze elastiche. Le variabili che contengono informazioni del genere sono in grado di annotare tre indicazioni distinte: la distanza normale, una tolleranza in più e una tolleranza in meno. Questo tipo di informazione si esprime secondo la forma seguente:

```
lunghezza_richiesta [plus lunghezza_in_estensione] [minus lunghezza_in_contrazione]
```

In pratica, è come dire che si fa riferimento a una certa lunghezza, a cui si può aggiungere quanto appare dopo la parola chiave '**`plus`**' e si può togliere quanto appare dopo '**`minus`**'. Come si può intuire, quando non si indicano i valori che danno elasticità, si sottintende in corrispondenza un valore zero.

L'elasticità fissata attraverso le parole chiave '**`plus`**' e '**`minus`**' non è tassativa. Di solito, il solo fatto che si consenta un'estensione, anche di un solo punto, fa sì che il salto sia allungabile in modo indefinito, in caso di necessità.

Eventualmente, si dichiara una variabile del genere con la forma seguente:

```
\newskip\nome
```

L'assegnamento, come si può intendere, ha la forma seguente:

```
\nome=lunghezza_elastica
```

Ovvero:

```
\nome=lunghezza [plus lunghezza] [minus lunghezza]
```

Le indicazioni sull'elasticità in estensione e in contrazione sono formate normalmente da lunghezze, come per esempio '**`plus 1pt`**', ma si possono usare anche delle definizioni astratte, rappresentate da tre parole chiave, precedute da un numero intero:


```
nfil
nfill
nfilll
```

Generalmente, il numero ***n*** è sempre 1 e va inteso come moltiplicatore della parola successiva; in pratica, '**`2fil`**' rappresenta un'elasticità doppia di '**`1fil`**'.

La parola chiave '**`fil`**' rappresenta un'elasticità di grado minimo, '**`fill`**' un'elasticità di grado medio e '**`filll`**' un'elasticità molto grande.

49.3.6 Operazioni con i valori numerici

La realizzazione di espressioni numeriche con TeX diventa abbastanza complicata. Si utilizzano fondamentalmente tre tipi di istruzione per modificare il valore di una variabile:

```
\advance\nome [by] valore
```

```
\multiply\nome [by] valore
```

```
\divide\nome [by] valore
```

In pratica, nel primo caso si incrementa la variabile del valore indicato (se il valore in questione è negativo, la variabile viene ridotta di conseguenza); nel secondo caso si assegna alla variabile il prodotto tra quanto contenuto prima per il valore indicato; nel terzo caso si divide il contenuto della variabile per il valore, assegnando alla stessa il risultato della divisione.

Come si può osservare, la parola chiave '**`by`**' è facoltativa e si può usare per facilitare la lettura umana dell'istruzione.

Il valore usato deve essere del tipo adatto alla variabile con cui si esegue l'operazione. Viene mostrato un esempio complessivo che dovrebbe essere comprensibile a sufficienza:

```
\countdef\pagina=0
\pagina=1
\newcount\contatore
\contatore=-7
\newdimen\lunghezza
\lunghezza=100pt
\newskip\elastico
\elastico=10pt plus 2pt minus 1pt
pagina = \the\pagina \par
contatore = \the\contatore \par
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
\multiply\pagina by 2
\advance\contatore by 1
\advance\lunghezza by 10pt
\advance\elastico by 5pt plus 3pt minus 2pt
pagina = \the\pagina \par
contatore = \the\contatore \par
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
\divide\lunghezza by 2
\divide\elastico by 2
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
```

pagina = 1
contatore = -7
lunghezza = 100.0pt
elastico = 10.0pt plus 2.0pt minus 1.0pt
pagina = 2
contatore = -6
lunghezza = 110.0pt
elastico = 15.0pt plus 5.0pt minus 3.0pt
lunghezza = 55.0pt
elastico = 7.5pt plus 2.5pt minus 1.5pt

Va tenuta in considerazione una scorciatoia importante per rappresentare il prodotto tra una costante numerica e il valore di una lunghezza; questa scorciatoia si usa di solito per gli assegnamenti:

n**nome

Qui il numero ***n*** non deve esprimere una lunghezza, pertanto non può contenere l'indicazione dell'unità di misura. Per esempio, se '**\hsize**' contiene la lunghezza 15 cm, con l'istruzione seguente si assegna alla variabile '**\mezza**' la metà di questa lunghezza, ovvero 7,5 cm:

```
\mezza=0.5\hsize
```

Nell'esempio seguente si creano due variabili: una contiene una lunghezza e l'altra una lunghezza elastica. Dopo aver controllato il valore iniziale di queste, si riassegna loro la metà del loro valore di partenza, controllando successivamente il risultato ottenuto. Si può osservare che '**\elastico**' perde l'informazione sull'elasticità, diventando una lunghezza normale.

```
\newdimen\lunghezza
\lunghezza=100pt
\newskip\elastico
\elastico=10pt plus 2pt minus 1pt
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
\lunghezza=0.5\lunghezza
\elastico=0.5\elastico
lunghezza = \the\lunghezza \par
elastico = \the\elastico \par
```

lunghezza = 100.0pt
elastico = 10.0pt plus 2.0pt minus 1.0pt
lunghezza = 50.0pt
elastico = 5.0pt

49.3.7 Trasformazione in stringa di dati numerici

I dati numerici, quando devono essere convertiti in stringhe, possono essere rappresentati in forme differenti. La parola di controllo '**\the**' consente di ottenere una trasformazione «normale» di qualunque variabile in stringa:

\the*parola_di_controllo*

Per esempio, se '**\lunghezza**' è stata dichiarata come lunghezza contenente il valore 10 cm, l'espansione di '**\the\lunghezza**' può generare la stringa '**284.45274pt**'.

Un valore numerico può essere trasformato in numero intero, ammesso che ciò abbia senso, con la parola di controllo '**\number**':

\number*parola_di_controllo*

In questo caso, se si tenta di trasformare una lunghezza in numero, si ottiene la dimensione in punti *scaled point*; nel caso dell'esempio precedente, si otterrebbe la rappresentazione del valore 18646798, corrispondente ai 10 cm di prima espressi secondo l'unità di misura minima che TeX è in grado di gestire. Per quanto riguarda le lunghezze elastiche, non è possibile usare la trasformazione attraverso '**\number**'.

Quando il valore da trasformare è un intero positivo maggiore di zero, si può ottenere la rappresentazione in numero romano, con lettere minuscole, attraverso la parola di controllo '**\romannumeral**':

\romannumeral*parola_di_controllo*

L'esempio seguente riepiloga l'uso delle forme di trasformazione dei dati numerici in stringa che sono appena state descritte:

```
\newdimen\lunghezza
\newcount\contatore
\newcount\pagina
\lunghezza=1cm
\contatore=-7
\pagina=123
lunghezza = \the\lunghezza\ = \number\lunghezza scaled point \par
contatore = \the\contatore\ = \number\contatore \par
pagina = \the\pagina\ = \number\pagina = \romannumeral\pagina \par
```


lunghezza = 28.45274pt = 1864679scaled point
contatore = -7 = -7
pagina = 123 = 123= cxxiii

49.3.8 Controllo del campo di azione delle variabili

Le parentesi graffe, oppure le parole di controllo '**\bgroup**' e '**\egroup**', servono a delimitare e isolare una zona rispetto al testo che si trova al loro esterno. Una dichiarazione o un assegnamento fatto all'interno di una zona delimitata da parentesi graffe ha effetto in quell'ambito e in tutte le altre zone che possono essere annidate al suo interno, mentre all'esterno non esiste più. Si osservi l'esempio seguente:

```
\newdimen\lunghezza
\newcount\contatore
\lunghezza=1cm
\contatore=7
{
    \lunghezza=10cm
    \contatore=14
    {
        \lunghezza=100cm
        \contatore=21
        lunghezza = \the\lunghezza{}
        contatore = \the\contatore \par
     }
    lunghezza = \the\lunghezza{}
    contatore = \the\contatore \par
}
lunghezza = \the\lunghezza{} contatore = \the\contatore
\bye
```

Si ottiene il testo che segue:

lunghezza = 2845.27559pt contatore = 21
lunghezza = 284.52756pt contatore = 14
lunghezza = 28.45274pt contatore = 7

Perché un assegnamento abbia valore in modo globale, si usa la parola di controllo '**\global**'. Si osservi l'esempio seguente:

```
\newdimen\lunghezza
\newcount\contatore
\lunghezza=1cm
\contatore=7
{
    \lunghezza=10cm
    \contatore=14
    {
        \global\lunghezza=100cm
        \contatore=21
        lunghezza = \the\lunghezza{}
        contatore = \the\contatore \par
     }
    lunghezza = \the\lunghezza{}
    contatore = \the\contatore \par
}
lunghezza = \the\lunghezza{} contatore = \the\contatore
\bye
```

In questo caso, la lunghezza appare essere sempre di 100 cm (2845,27559 punti normali), anche quando si ritorna al di fuori dei raggruppamenti. In pratica, si ottiene il testo che segue:

lunghezza = 2845.27559pt contatore = 21
lunghezza = 2845.27559pt contatore = 14
lunghezza = 2845.27559pt contatore = 7

49.3.9 Ordine nell'espansione delle sequenze di controllo

Esiste un problema con TeX, legato all'ordine in cui vengono espanse le parole di controllo. L'esempio più comune è dato dalla difficoltà con cui si riesce a trasformare un numero romano ottenuto da '**\romannumeral**' in maiuscolo. Per esempio,

```
\newcount\pagina
\pagina=4
\uppercase{\romannumeral\pagina}
```

genera solo:

In pratica, '**\uppercase**' si trova a intervenire su una stringa che ancora non c'è, per cui non succede nulla. Per risolvere il problema si usa la parola di controllo '**\expandafter**' che anticipa l'esecuzione di ciò che segue:

```
\newcount\pagina
\pagina=4
\uppercase\expandafter{\romannumeral\pagina}
```

IV

Purtroppo il funzionamento di questa parola di controllo non è intuitivo e spesso si devono fare vari tentativi prima di riuscire a fare ciò che si intende.

49.4 Dichiarazione di macroistruzioni[3]

Si può dare un nome a un'espressione stringa attraverso la sintassi seguente:

```
\def\nome{espressione_stringa}
```

Si osservi che, al contrario di altre situazioni, qui TeX richiede che la parentesi graffa aperta segua immediatamente il nome (o il simbolo dell'ultimo parametro, come viene descritto nel seguito), senza alcuna spaziatura intermedia.

Questo tipo di dichiarazione serve in generale per realizzare delle macroistruzioni; tuttavia, utilizzandola solo così, si fa in modo di ottenere l'espansione di ciò che è contenuto fra le parentesi graffe nel momento in cui si inserisce nel testo l'istruzione '***nome***'. Si osservi l'esempio:

```
\def\eTeX{e-\TeX}

Bla bla bla bla \eTeX{} bla bla...
\bye
```

In questo caso si vuole dichiarare la parola di controllo '**\eTeX**', con cui diventa facile uniformare la scrittura di questo nome nel testo. Se ci fosse un ripensamento sulla forma da dare al nome, basterebbe modificare la sua dichiarazione iniziale.

Una macro del genere può essere modificata solo riassegnandole un altro valore, nello stesso modo usato per la sua dichiarazione iniziale.

È importante ricordare che in fase di composizione, le parole di controllo assorbono gli spazi alla loro destra, per cui è necessario usare una tecnica per evitarlo, come l'inserimento di un gruppo vuoto alla fine della stessa.

Nel momento in cui si utilizza una parola di controllo corrispondente a una macroistruzione dichiarata in questo modo, si ottiene l'espansione del suo contenuto. In altri termini, la parola di controllo diventa una forma abbreviata per scrivere un testo più articolato, il quale può contenere a sua volta altre sequenze di controllo che vengono espanse solo all'ultimo momento. Si osservi l'esempio seguente:

```
\def\resistenza{\valore{} ohm \tolleranza{} \%}
\def\valore{100}
\def\tolleranza{5}
25 resistenze \resistenza \par
\def\valore{300}
\def\tolleranza{1}
30 resistenze \resistenza \par
\bye
```

Si ottiene un testo simile a quello seguente, con cui si comprende il fatto che le parole di controllo '**\valore**' e '**\tolleranza**' vengono espanse per ultime:

25 resistenze 100 ohm 5 %
30 resistenze 300 ohm 1 %

L'espansione di una parola di controllo del genere avviene in modo letterale, salvo naturalmente l'espansione successiva del suo contenuto, ma questo fatto significa che vengono riprodotte anche le interruzioni delle righe e gli spazi. Alle volte si preferisce strutturare il contenuto di una dichiarazione del genere, per cui si utilizzano dei commenti per evitare di dare un significato agli spazi che si inseriscono. Si osservi l'esempio precedente dopo una piccola modifica:

```
\def\resistenza{

\valore{} ohm \tolleranza{} \%}

\def\valore{100}
\def\tolleranza{5}
25 resistenze \resistenza \par
\def\valore{300}
\def\tolleranza{1}
30 resistenze \resistenza \par
\bye
```

In questo caso, ciò che si ottiene è diverso, perché la parola di controllo '**\resistenza**' si espande inserendo inizialmente una riga vuota, ovvero ciò che poi si traduce nell'inizio di un paragrafo nuovo:

25 resistenze
100 ohm 5 %
30 resistenze
300 ohm 1 %

Per evitare questo tipo di inconveniente, si può mettere un commento all'inizio della riga vuota, che così perde questa sua particolarità:

```
\def\resistenza{
%
\valore{} ohm \tolleranza{} \%}
```

Di solito, in queste situazioni si mette un commento anche dopo la parentesi graffa aperta:

```
\def\resistenza{%
%
\valore{} ohm \tolleranza{} \%}
```

Oltre a questo è da tenere in considerazione che nel momento dell'espansione, ciò che si ottiene non risulta contenuto in una zona separata; in altri termini, ciò che viene dichiarato o modificato all'interno di questa definizione, continua a valere anche al di fuori. Eventualmente, se si intende che l'espansione debba generare una zona isolata, vanno usate le parentesi graffe come già mostrato. Per esempio:

```
\def\attenzione{{\bf ATTENZIONE!!!}}
```

In questo caso, l'istruzione '**\bf**' inizia la scrittura in neretto; così, la parola di controllo '**\attenzione**' permette di inserire la scritta che si vede, in neretto, senza interferire con il testo successivo.

Gli esempi usati fino a questo punto sono riconducibili all'idea di una funzione che non prevede parametri per la chiamata, ricevendo i dati attraverso variabili globali. Per dichiarare una macroistruzione in grado di ricevere degli argomenti si usa una dichiarazione come quella seguente:

```
\def\nome#1[#2...[#9]]{espressione_stringa}
```

In una dichiarazione del genere si possono indicare un massimo di nove parametri (parametri formali, a cui si abbinano nella chiamata gli argomenti attuali), rappresentati da '**#1**', '**#2**',... '**#9**', che possono essere inseriti nella stringa contenuta tra le parentesi graffe. Volendo modificare l'esempio già visto, le cose si potrebbero semplificare nel modo seguente:

```
\def\resistenza#1#2{#1 ohm #2 \%}
25 resistenze \resistenza{100}{5} \par
30 resistenze \resistenza{300}{1}
\bye
```

Nella dichiarazione, i simboli '**#*n***' che si inseriscono all'interno del testo contenuto tra parentesi graffe possono essere indicati an-

che più volte, ottenendo sempre l'espansione del parametro ***n***-esimo corrispondente.

È ormai evidente il modo in cui deve essere usata la parola di controllo che può essere chiamata con dei parametri:

```
\nome{parametro_1}[{parametro_2}[...{parametro_n}]]
```

È importante sottolineare che, contrariamente a quanto si potrebbe immaginare, la stringa utilizzata come argomento della chiamata di una macroistruzione non può essere separata in paragrafi.

Si osservi che molte macroistruzioni predefinite di TeX isolano l'espressione stringa che restituiscono all'interno di un raggruppamento, in modo tale che ciò che si cambia al suo interno non si rifletta nel testo successivo. Di solito questo fatto è un comportamento «logico», o intuitivo, ma non si deve pensare che la definizione di una macro implichi automaticamente questa forma di isolamento. In pratica, il fatto di usare una parola di controllo del tipo '***nome*{*espressione*}**', non significa implicitamente che quanto inserito come argomento della chiamata non debba anche influenzare il testo successivo.

49.4.1 Chiamata di macroistruzioni che richiedono l'indicazione di argomenti

È stata mostrata la sintassi per la chiamata di una macroistruzione che richiede di fornire uno o più argomenti:

```
\nome{parametro_1}[{parametro_2}[...{parametro_n}]]
```

Tuttavia, si tratta di una semplificazione. Infatti, gli argomenti possono anche non essere racchiusi tra parentesi graffe, ma in tal caso, il primo argomento diventa il primo carattere che segue. Si osservi l'esempio seguente, in cui si dichiara una macro con tre parametri e poi la si chiama senza raggruppare gli argomenti:

```
\def\ciao#1#2#3{%
    Primo argomento della chiamata: #1\par
    Secondo argomento della chiamata: #2\par
    Terzo argomento della chiamata: #3\par
}

\ciao, come stai?
```

Primo argomento della chiamata: ,
Secondo argomento della chiamata: c
Terzo argomento della chiamata: o
me stai?

Nella sezione 49.10 viene mostrato l'uso di macro come '**\'**' che servono ad aggiungere un accento alla lettera successiva. Queste si usano solitamente senza circoscrivere la lettera che segue entro parentesi graffe.

49.5 Riferimenti ad altre parole o simboli di controllo

È possibile creare dei riferimenti a una parola o a un simbolo di controllo, usando la sintassi seguente:

```
\let\nome_nuovo=\nome
```

In pratica, in questo modo si crea una parola di controllo alternativa a un'altra già esistente: '***nome_nuovo*** può essere usato al posto di '***nome***'.[4] Si osservi l'esempio:

```
\newdimen\altezza
\altezza=10cm
\let\grandezza=\altezza
\altezza=20cm
La scatola ha una grandezza di \the\grandezza.
\bye
```

Si ottiene in pratica il testo

La scatola ha una grandezza di 569.05511pt.

dove 569,055 11 punti corrispondono esattamente a 20 cm, ovvero l'ultimo valore assegnato alla lunghezza '**\altezza**', a cui punta anche '**\grandezza**'.

Tuttavia, se si fa un esperimento simile con una parola di controllo corrispondente a una macroistruzione definita con l'istruzione '**\def**', il riferimento che si genera con l'istruzione '**\let**' è quello che punta alla macroistruzione di quel momento, mentre una ridefinizione della parola di controllo di partenza non si riflette nel riferimento:

```
\def\resistenza#1#2{#1 ohm #2 \%}
\let\prova=\resistenza
25 resistenze \resistenza{100}{5}

25 resistenze \prova{100}{5}

\def\resistenza#1#2{#2 ohm #1 \%}
30 resistenze \resistenza{300}{1}

30 resistenze \prova{300}{1}
\bye
```

Quello che si ottiene dalla composizione di questo esempio è il testo seguente:

25 resistenze 100 ohm 5 %
25 resistenze 100 ohm 5 %
30 resistenze 1 ohm 300 %
30 resistenze 300 ohm 1 %

49.6 Testo normale e ambienti matematici

Una caratteristica molto importante di TeX è la distinzione tra due modalità di funzionamento. In pratica, si distingue un contesto «normale» da un contesto matematico. L'ambiente matematico si introduce e si conclude con il simbolo '**$**' e in tale situazione diventano disponibili delle istruzioni che non si possono utilizzare al di fuori di questo ambito, mentre alcune istruzioni dell'ambiente normale non lo sono più.

Per esempio, alcuni caratteri esistono solo nell'ambiente matematico; è già stato visto in che modo vanno indicate le parentesi graffe quando si scrive in un ambito normale, attraverso le istruzioni '**$\{$**' e '**$\}$**'. In pratica, il dollaro iniziale e finale di queste istruzioni serve ad aprire e a chiudere l'ambiente matematico.

49.7 Modalità orizzontale e modalità verticale

Nel suo lavoro di composizione, TeX distingue due situazioni, definite come ***modalità orizzontale*** e ***modalità verticale***. Per comprendere la differenza tra queste due situazioni, occorre pensare alla struttura di ciò che si inserisce in una pagina.

Ogni oggetto che viene inserito in una pagina è una scatola, con le sue dimensioni. Queste scatole si inseriscono a loro volta all'interno di altre; per esempio, una lettera è una scatola che si inserisce in una riga, ovvero un'altra scatola, che a sua volta si inserisce in un paragrafo, ovvero ancora un'altra scatola, ecc.

In base al contesto, orizzontale o verticale, TeX si occupa di inserire spazi orizzontali o verticali: tra le lettere di una parola, tra le parole, tra le righe, tra i paragrafi, ecc.

Quello che conta comprendere di tutto questo è che alcune istruzioni possono essere inserite solo in modalità orizzontale, altre solo in modalità verticale. Volendo sperimentare se un certo contesto si trovi in modalità orizzontale o verticale, si può realizzare la macro seguente e collocarla nel testo dove si ritiene opportuno; nella composizione finale si ottiene alternativamente la frase «modalità orizzontale», oppure «modalità verticale»:

```
\def\verificamodo{%
\ifvmode
    modalit\'a verticale
\else
    modalit\'a orizzontale
\fi}
```

Nella sezione 49.8 viene spiegato l'uso dell'istruzione '**\ifvmode**'.

49.8 Strutture di controllo

Il linguaggio di TeX possiede una serie di strutture di controllo condizionali, in cui parte della condizione è implicita nel nome dell'istruzione con cui la si introduce. Questo insieme di strutture ha una sintassi comune riconducibile alla semplificazione seguente:

```
\if nome_condizione [argomento]
    testo_e_altre_istruzioni_se_vero
\else
    testo_e_altre_istruzioni_se_falso
\fi
```

Spesso, per completare la struttura anche quando una delle due ipotesi non deve generare alcun risultato, si utilizza l'istruzione '**\relax**', la quale rappresenta proprio l'operazione nulla.

Nelle sezioni seguenti vengono descritte solo le strutture condizionali più comuni.

49.8.1 Istruzione «\ifodd»

È possibile verificare se un numero intero è dispari o pari con la struttura seguente:

```
\ifodd n
    testo_e_altre_istruzioni_se_dispari
\else
    testo_e_altre_istruzioni_se_pari
\fi
```

Di solito, al posto del numero ***n*** si inserisce l'istruzione '**\pageno**' che restituisce il numero della pagina corrente, permettendo così di verificare se la pagina è dispari o pari:

```
\ifodd \pageno
    testo_e_altre_istruzioni_se_pagina_dispari
\else
    testo_e_altre_istruzioni_se_pagina_pari
\fi
```

49.8.2 Istruzione «\ifmmode»

È possibile verificare se ci si trova in modalità normale o in modalità matematica con la struttura seguente:

```
\ifmmode
    testo_e_altre_istruzioni_se_modalità_matematica
\else
    testo_e_altre_istruzioni_se_modalità_normale
\fi
```

49.8.3 Istruzione «\ifnum»

È possibile mettere a confronto due numeri con la struttura seguente:

```
\ifnum m=|>|<n
    testo_e_altre_istruzioni_se_confronto_valido
\else
    testo_e_altre_istruzioni_se_confronto_non_valido
\fi
```

In pratica, il confronto può avvenire solo con gli operatori '**=**', '**>**' e '**<**', per indicare rispettivamente se i due valori sono uguali, se il primo è maggiore del secondo oppure se il primo è minore del secondo.

49.8.4 Istruzioni «\ifhmode» e «\ifvmode»

È possibile verificare se la composizione si trova in modalità orizzontale o verticale con una delle due strutture seguenti:

```
\ifhmode
    testo_e_altre_istruzioni_se_modalità_orizzontale
\else
    testo_e_altre_istruzioni_se_modalità_verticale
\fi
```

```
\ifvmode
    testo_e_altre_istruzioni_se_modalità_verticale
\else
    testo_e_altre_istruzioni_se_modalità_orizzontale
\fi
```

49.9 Verifica del significato di un'istruzione elementare

Per TeX, un'istruzione elementare può essere il singolo carattere di una parola, oppure una sequenza di controllo. Volendo comprendere il senso di qualcosa, si può verificare come intenda TeX questa o quell'istruzione. Per questo, occorre avviare la composizione indicando un file vuoto; per esempio, in un sistema Unix si potrebbe fare così:

```
$ tex /dev/null [Invio]

  This is TeX, Version 3.14159 (Web2C 7.3.1)
  (/dev/null)
  *
```

A questo punto, dall'invito, si può usare l'istruzione '**\show**' nel modo seguente:

```
\show istruzione_elementare
```

Per esempio, ci si può domandare se il carattere '**@**' abbia un significato particolare:

```
*\show @ [Invio]

  > the character @.
  <*> \show @

  ?
```

A questo punto, la composizione si ferma in attesa di altre indicazioni, mostrando un invito differente. Questo permette di comprendere anche che non conviene usare '**\show**' in un file normale del quale si vuole ottenere la composizione, perché in quel punto ci sarebbe una sospensione con richiesta di intervento. A ogni modo, per proseguire basta premere [*Invio*], quindi si può chiedere di conoscere qualcosa di diverso:

```
?[Invio]

*\show \TeX [Invio]
```

In questo caso si vuole conoscere in cosa consiste la macro '**\TeX**' ed ecco il risultato che si ottiene:

```
> \TeX=macro:
->T\kern -.1667em\lower .5ex\hbox {E}\kern -.125emX.
<*> \show \TeX

?
```

Pertanto, questo significa che '**\TeX**' è definita come:

```
\def\TeX{T\kern -.1667em\lower .5ex\hbox {E}\kern -.125emX}
```

Per completare l'esempio, conviene vedere come appare una macro che prevede l'indicazione di un parametro, come nel caso di '**\'**', che serve a mettere un accento grave su una lettera:

```
?[Invio]
```

*\show \‘ [*Invio*]

```
> \‘=macro:
#1->{\accent 18 #1}.
<*> \show \‘

?
```

In pratica, la definizione originale è la seguente:

```
\def\‘#1{{\accent 18 #1}}
```

Per terminare l’uso interattivo basta inserire alla fine l’istruzione ‘**\bye**’:

?[*Invio*]

*\bye [*Invio*]

```
No pages of output.
Transcript written on null.log.
```

49.10 Caratteri

In condizioni normali, il testo scritto nel sorgente viene riprodotto nello stesso modo nella composizione finale, dopo essere stato reimpaginato opportunamente. Tuttavia, alcuni simboli hanno significati speciali, per cui si devono usare simboli di controllo al loro posto, inoltre le lettere accentate e altri caratteri che non fanno parte dell’ASCII standard non possono essere usati direttamente, per cui servono altri simboli o parole di controllo per generarli.

A fianco di questo problema, nasce poi l’esigenza di poter scrivere utilizzando stili, forme e corpi differenti.

49.10.1 Caratteri speciali

Come è descritto nella tabella 49.34, alcuni caratteri hanno un significato speciale e non producono il simbolo corrispondente in fase di composizione. Per porre rimedio a questo problema e alla mancanza di altri caratteri, si usano generalmente delle sequenze di controllo. Tuttavia, dal momento che si distinguono due contesti di composizione differenti (quello normale e quello matematico), certi caratteri o certe sequenze di controllo esistono solo nella modalità matematica.

In generale, salvo altra indicazione, si fa sempre riferimento alla modalità di composizione normale, per cui un carattere o una sequenza di controllo che può apparire solo in modalità matematica, viene mostrata generalmente delimitandola tra due simboli ‘**$**’. Per esempio, quando si afferma che le parentesi graffe si ottengono con le istruzioni ‘**$\{$**’ e ‘**$\}$**’, è chiaro che in modalità matematica non serve più la delimitazione con i simboli ‘**$**’.

La tabella 49.88 riepiloga le sequenze di controllo per i caratteri speciali disponibili.[5]

Tabella 49.88. Elenco delle sequenze di controllo necessarie a ottenere i caratteri speciali che non richiedono la sovrapposizione di accenti.

Simbolo	Codice	Simbolo	Codice
\	\$backslash$	{	$\{$
}	$\}$	$	\$
&	\&	#	\#
^	\^{}	_	_
%	\%	~	\~{}
<	$<$	>	$>$
œ	\oe	Œ	\OE
æ	\ae	Æ	\AE
å	\aa	Å	\AA
ø	\o	Ø	\O
ł	\l	Ł	\L
ß	\ss	ı	\i
ȷ	\j	¡	!`
¿	?`	†	\dag
‡	\ddag	§	\S
¶	\P	£	{\it\$}
©	\copyright	…	\dots
TEX	\TeX		

49.10.2 Accenti

TeX ha la capacità di collocare un accento sopra ogni simbolo, anche se ciò che si ottiene può non avere senso per qualunque lingua. Questa apposizione di accenti si ottiene con sequenze di controllo che precedono la lettera da accentare. Quando si tratta di simboli di controllo, la lettera successiva può essere unita alla sequenza, mentre quando si tratta di parole di controllo occorre lasciare uno spazio di sicurezza. Per esempio, si scrive ‘**\‘o**’ per ottenere la lettera «ò», mentre si scrive ‘**\c c**’ per ottenere la lettera ‘**ç**’.[6] La tabella 49.89 riepiloga le sequenze di controllo necessarie a ottenere tutti gli accenti disponibili, usando la lettera «o» come esempio.

Tabella 49.89. Elenco delle sequenze di controllo necessarie a ottenere le lettere accentate.

Simbolo	Codice	Simbolo	Codice
ò	\`o	ó	\´o
ô	\^o	ö	\"o
õ	\~o	ō	\=o
ȯ	\.o	ŏ	\u o
ǒ	\v o	ő	\H o
o͡o	\t oo	o̧	\c o
o̱	\d o	o̲	\b o

Il fatto che TeX aggiunga l’accento sopra un simbolo, significa che occorre poi provvedere a eliminare il puntino sopra le lettere «i» e «j» minuscole. Pertanto, per scrivere la lettera «ì», occorre usare l’istruzione ‘**\‘\i**’.

49.10.3 Apici, trattini e legati automatici

TeX ha una gestione estetica particolare per alcuni caratteri. Purtroppo mette a disposizione solo virgolette alte, che vanno usate opportunamente a coppia, nella forma ‘***‘testo’***’ e ‘***‘‘testo’’***’. Si osservi l’esempio seguente e in che modo TeX lo traduce tipograficamente:

```
"virgolette doppie normali" \par
‘virgolette singole, aperte e chiuse’ \par
‘‘virgolette doppie, aperte e chiuse’’ \par
```

”virgolette doppie normali”
‘virgolette singole, aperte e chiuse’
“virgolette doppie, aperte e chiuse”

Evidentemente, è bene non usare le virgolette doppie normali (‘**"**’), a meno che ci sia una ragione precisa per farlo.

TeX è in grado di gestire tre lunghezze differenti per il trattino o la lineetta orizzontale.[7] Questi si ottengono con il trattino normale,

ovvero il segno meno, usato ripetutamente. In pratica, '**-**' genera un trattino corto; '**--**' genera un trattino medio e '**---**' genera un trattino lungo. Questo tipo di automatismo può creare dei problemi quando il trattino deve assumere proprio il significato di un segno meno e non è molto bella l'idea di usare due trattini per ottenere una lineetta media. In questo caso, si usa preferibilmente la sequenza '**$-$**', dal momento che nella modalità matematica questo comportamento non ha luogo. Si osservi l'esempio:

```
Trattino corto: - \par
Trattino medio: -- \par
Trattino lungo: --- \par
Segno meno: $-$ \par
```

Trattino corto: -
Trattino medio: –
Trattino lungo: —
Segno meno: −

Per motivi estetici, la tradizione tipografica richiede che alcuni accoppiamenti di lettere vengano legati assieme. Si tratta dell'unione della lettera «f» con «i», «l» e «f». TeX rispetta questa tradizione, ma solo nella modalità normale, perché in modalità matematica le esigenze sono differenti. Si osservi l'esempio:

```
ff fi fl ffi ffl \par
$ff fi fl ffi ffl$ \par
```

ff fi fl ffi ffl
$ff\,fi\,fl\,ffi\,ffl$

49.10.4 Istruzioni alternative per generare simboli particolari

Alcune tastiere non consentono di utilizzare facilmente dei simboli che per TeX sono invece importanti. In particolare, ci possono essere difficoltà a ottenere un apostrofo corrispondente all'accento grave. Quando l'intento è solo quello di scrivere degli apici, singoli o doppi, si possono usare le parole di controllo '**\lq**' e '**\rq**' per aprire e chiudere rispettivamente. Per esempio,

```
\lq\lq ciao\rq\rq
```

racchiude tra apici doppi la parola «ciao». Tuttavia, questo sistema non può sostituire i simboli di controllo usati per ottenere l'accento grave e acuto. Per ovviare all'inconveniente, si può leggere il codice delle macroistruzioni standard di TeX per scopre un metodo alternativo, oppure si può usare l'istruzione '**\show**' in modalità interattiva:

```
\def\‘#1{{\accent18 #1}}
\def\’#1{{\accent19 #1}}
\def\v#1{{\accent20 #1}}
\def\u#1{{\accent21 #1}}
\def\=#1{{\accent22 #1}}
\def\^#1{{\accent94 #1}}
\def\.#1{{\accent95 #1}}
\def\H#1{{\accent"7D #1}}
\def\~#1{{\accent"7E #1}}
\def\"#1{{\accent"7F #1}}
```

In pratica, è sufficiente definire una parola di controllo alternativa alle sequenze che non si possono riprodurre facilmente. Ecco un esempio concreto per gli accenti grave a acuto:

```
\def\graveaccent#1{{\accent18 #1}}
\def\acuteaccent#1{{\accent19 #1}}
```

In questo modo, al posto di '**\‘e**' si può scrivere '**\graveaccent{e}**', anche se ciò richiede una digitazione un po' lunga.

Per quanto riguarda le parentesi quadre, queste possono essere rappresentate con la coppia '**\lbrack**' e '**\rbrack**' per rappresentare rispettivamente la parentesi aperta e quella chiusa. Per le parentesi graffe le cose si complicano: per inserirle nel testo come simboli tipografici, si possono usare le istruzioni '**$\lbrace$**' e '**$\rbrace$**', che come si vede richiedono un contesto matematico, come già avveniva con '**$\{$**' e '**$\}$**', ma questo significa anche che non c'è un'alternativa alle parentesi graffe usate nelle istruzioni con finalità differenti, salvo la possibilità di usare '**\bgroup**' e '**\egroup**' quando si tratta di raggruppamenti.

Tabella 49.99. Alternative ad alcune sequenze di controllo che possono creare difficoltà con tastiere incomplete.

Sequenza normale	Alternativa
‘	\lq
’	\rq
\‘	{\accent 18 *x*}
\’	{\accent 19 *x*}
[	\lbrack
]	\rbrack
$\{$	$\lbrace$
$\}$	$\rbrace$

49.10.5 Caratteri da stampa standard

In condizioni normali, TeX mette a disposizione un carattere tondo con grazie. È possibile cambiare il gruppo stilistico e la forma del carattere con comandi generici, che non fanno riferimento a un nome preciso, ma a un aspetto di massima. Questi comandi hanno effetto dal momento in cui vengono inseriti, fino a quando ne viene incontrato un altro, oppure fino alla fine del raggruppamento in cui si trovano. Si tratta di parole di controllo di due sole lettere, elencate brevemente nella tabella 49.100.

Tabella 49.100. Selezione standard del tipo di carattere.

Sequenza	Nome	Significato	Nome corrispondente
\rm	*roman*	tondo	cmr10
\bf	*boldface*	neretto	cmbx10
\it	*italic*	corsivo	cmti10
\sl	*slanted*	inclinato	cmsl10
\tt	*typewriter*	dattilografico	cmtt10

È importante osservare che l'uso di queste parole di controllo non va a sommarsi con le precedenti, cambiando il gruppo stilistico e la forma simultaneamente. In pratica, in condizioni normali, non si può ottenere un neretto-corsivo o un neretto-inclinato combinando i comandi relativi. Si osservi l'esempio che segue:

```
Testo iniziale, \bf testo in neretto, \it testo in corsivo,
\sl testo inclinato, \tt testo dattilografico, \rm testo
tondo normale.
L'uso delle parentesi graffe {\it consente di circoscrivere}
l'effetto dei comandi che intervengono sul tipo di
carattere.
```

Testo iniziale, **testo in neretto,** *testo in corsivo, testo inclinato,* `testo dattilografico,` testo tondo normale. L'uso delle parentesi graffe *consente di circoscrivere* l'effetto dei comandi che intervengono sul tipo di carattere.

Prima di proseguire sull'argomento è il caso di osservare un problema che si presenta nel momento in cui si utilizza un carattere inclinato o corsivo: lo spostamento verso destra può creare degli inconvenienti, quando si incontra con un testo non inclinato. In caso di necessità si può aggiungere un piccolo spazio, inteso come correzione del corsivo, con il simbolo di controllo '**\/**', che rappresenta simbolicamente il concetto. Si osservi l'esempio seguente:

```
(testo normale tra parentesi tonde che finisce con una lettera l) \par
({\it testo corsivo tra parentesi tonde che finisce con una lettera l}) \par
({\it testo corsivo con correzione dopo la lettera l}\/) \par
```

(testo normale tra parentesi tonde che finisce con una lettera l)
(*testo corsivo tra parentesi tonde che finisce con una lettera l*)
(*testo corsivo con correzione dopo la lettera l*)

Le parole di controllo '**\rm**', '**\it**', '**\sl**', '**\bf**' e '**\tt**' sono delle piccole macroistruzioni standard di TeX. Infatti, le informazioni sui caratteri da stampa sono contenute in file particolari che possono essere utilizzati con l'istruzione seguente:

```
\font\nome=carattere
```

In pratica, in questo modo si dichiara una parola di controllo a cui si attribuisce il carattere indicato a destra. Il nome del carattere corrisponde solitamente alla prima parte del nome del file che lo contiene (la radice che rimane togliendo l'estensione) e si può esprimere usando indifferentemente lettere maiuscole o minuscole. Nell'esempio seguente, si dichiara un carattere tondo senza grazie, corrispondente al nome '**cmss10**':

```
\font\sans=cmss10
\sans Ecco un testo senza grazie, scritto con il carattere
''Computer Modern Sans Serif'' da 10 punti.
```

Ecco un testo senza grazie, scritto con il carattere "Computer Modern Sans Serif" da 10 punti.

Il carattere da stampa che si seleziona in questo modo ha già un corpo stabilito. In questo caso si tratta di un carattere da 10 punti. Per cambiare corpo, si può cambiare carattere, oppure si può ingrandire e ridurre espressamente, intervenendo nell'istruzione di dichiarazione:

```
\font\nome=carattere scaled n
```

Il numero ***n*** che si indica alla fine esprime un rapporto in millesimi dell'adattamento che si vuole ottenere. Per esempio,

```
\font\sans=cmss10 scaled 1200
```

richiede espressamente di abbinare alla parola di controllo '**\sans**' il carattere '**cmss10**' al 120 %, cioè ingrandito di un quinto in più.

```
\font\sans=cmss10 scaled 1200
\sans Ecco un testo senza grazie, scritto con il carattere
''Computer Modern Sans Serif'' da 10 punti, ingrandito
al 120 \%.
```

Ecco un testo senza grazie, scritto con il carattere "Computer Modern Sans Serif" da 10 punti, ingrandito al 120 %.

Naturalmente, con lo stesso criterio si può anche ridurre un carattere, sempre tenendo presente che 1000 rappresenta la dimensione di partenza. Tuttavia, in generale non si usa questa forma, sostituendo il numero con un'istruzione apposita, utile a livello mnemonico:

```
\font\nome=carattere scaled \magstep n
```

In questo caso, il numero ***n*** può andare da zero a cinque, dove zero corrisponde al corpo attuale, mentre valori maggiori rappresentano piccoli incrementi percentuali. A ogni modo si può verificare facilmente cosa succede, con il codice TeX seguente:

```
\newcount\ingrandimento
\ingrandimento=\magstep 1
\font\sans=cmss10 scaled \magstep 1
\sans Ecco un testo senza grazie, scritto con il carattere
''Computer Modern Sans Serif'' da 10 punti, ingrandito al
\the\ingrandimento{} per mille.
```

Ecco un testo senza grazie, scritto con il carattere "Computer Modern Sans Serif" da 10 punti, ingrandito al 1200 per mille.

A questo punto si pone il problema di comprendere l'utilità di usare un carattere da stampa adatto alla dimensione del corpo che interessa. In pratica, se il carattere corrispondente alla sigla '**cmr5**' è fatto per un corpo da cinque punti, quando serve un corpo da 12 è bene usare il carattere '**cmr12**', se disponibile, invece di ingrandire il primo 2,4 volte (240 %). Si osservi l'esempio seguente, in cui si mettono a confronto diversi caratteri di piccole dimensioni che vengono ingranditi tutti in modo da avere un corpo finale di 12 punti:

```
\font\cinque=cmr5 scaled 2400
\cinque Il carattere cmr5 ingrandito al 240 \% \par
\font\sei=cmr6 scaled 2000
\sei Il carattere cmr6 ingrandito al 200 \% \par
\font\sette=cmr7 scaled 1714
\sette Il carattere cmr7 ingrandito al 171,4 \% \par
\font\otto=cmr8 scaled 1500
\otto Il carattere cmr8 ingrandito al 150 \% \par
\font\nove=cmr9 scaled 1333
\nove Il carattere cmr9 ingrandito al 133 \% \par
\font\dieci=cmr10 scaled 1200
\dieci Il carattere cmr10 ingrandito al 120 \% \par
\font\normale=cmr12
\normale Il carattere cmr12 senza ingrandimenti \par
```

Il carattere cmr5 ingrandito al 240 %
Il carattere cmr6 ingrandito al 200 %
Il carattere cmr7 ingrandito al 171,4 %
Il carattere cmr8 ingrandito al 150 %
Il carattere cmr9 ingrandito al 133 %
Il carattere cmr10 ingrandito al 120 %
Il carattere cmr12 senza ingrandimenti

Tabella 49.114. Nomi dei caratteri da stampa standard a disposizione (Computer Modern).

Sigla	Denominazione originale	Corpo
cmbsy10	Computer Modern Bold Symbols	10 punti
cmbx5 cmbx6 cmbx7 cmbx8 cmbx9 cmbx10 cmbx12	Computer Modern Bold Extended Roman	5, 6, 7, 8, 9, 10 e 12 punti
cmbxsl10	Computer Modern Bold Extended Slanted Roman	10 punti
cmcsc10	Computer Modern Caps and Small Caps	10 punti
cmdunh10	Computer Modern Dunhill Roman	10 punti
cmex9 cmex10	Computer Modern Math Extension	9 e 10 punti
cmff10	Computer Modern Funny Roman	10 punti
cmfi10	Computer Modern Funny Italic	10 punti
cmfib8	Computer Modern Fibonacci Font	8 punti
cminch	Computer Modern Inch-High Sans Serif Bold Extended Caps and Digits	1 pollice
cmitt10	Computer Modern Italic Typewriter Text	10 punti
cmmi5 cmmi6 cmmi7 cmmi8 cmmi9 cmmi10 cmmi12	Computer Modern Math Italic	5, 6, 7, 8, 9, 10 e 12 punti
cmr5 cmr6 cmr7 cmr8 cmr9 cmr10 cmr12 cmr17	Computer Modern Roman	5, 6, 7, 8, 9, 10, 12 e 17 punti
cmsl8 cmsl9 cmsl10 cmsl12	Computer Modern Slanted Roman	8, 9, 10 e 12 punti
cmsltt10	Computer Modern Slanted Typewriter Text	10 punti
cmss8 cmss9 cmss10 cmss12 cmss17	Computer Modern Sans Serif	8, 9, 10, 12 e 17 punti
cmssbx10	Computer Modern Sans Serif Bold Extended	10 punti
cmssdc10	Computer Modern Sans Serif Demibold Condensed	10 punti

Sigla	Denominazione originale	Corpo
cmssi8 cmssi9 cmssi10 cmssi12 cmssi17	Computer Modern Slanted Sans Serif	8, 9, 10, 12 e 17 punti
cmssq8	Computer Modern Sans Serif Quotation Style	8 punti
cmssqi8	Computer Modern Sans Serif Quotation Style Slanted	8 punti
cmsy5 cmsy6 cmsy7 cmsy8 cmsy9 cmsy10	Computer Modern Math Symbols	5, 6, 7, 8, 9 e 10 punti
cmtcsc10	Computer Modern Typewriter Caps and Small Caps	10 punti
cmtex8 cmtex9	Computer Modern TeX Extended ASCII characters	8 e 9 punti
cmti7 cmti8 cmti9 cmti10 cmti12	Computer Modern Text Italic	7, 8, 9, 10 e 12 punti
cmtt8 cmtt9 cmtt10 cmtt12	Computer Modern Typewriter Text	8, 9, 10 e 12 punti
cmu10	Computer Modern Unslanted Italic	10 punti
cmvtt10	Computer Modern Variable-width Typewriter Text	10 punti

A fianco delle parole di controllo generiche per la selezione del carattere da stampa, esistono altre parole di controllo standard che contengono anche l'informazione sulla dimensione. Si tratta dell'elenco riportato nella tabella 49.115. Ciò permette di non utilizzare una definizione dettagliata del carattere quando quello che serve cambiare è solo la dimensione (entro i limiti normali già previsti).

Tabella 49.115. Parole di controllo per la selezione di un carattere da stampa generico, con l'indicazione della dimensione.

Parola di controllo	Effetto
`\tenrm`	Tondo, 10 punti.
`\sevenrm`	Tondo, sette punti.
`\fiverm`	Tondo, cinque punti.
`\tenit`	Corsivo, 10 punti.
`\tenbf`	Neretto, 10 punti.
`\sevenbf`	Neretto, sette punti.
`\fivebf`	Neretto, cinque punti.
`\tentt`	Dattilografico, 10 punti.
`\tensl`	Inclinato, 10 punti.
`\teni`	Corsivo matematico, 10 punti.
`\seveni`	Corsivo matematico, sette punti.
`\fivei`	Corsivo matematico, cinque punti.
`\tensy`	Simboli matematici, 10 punti.
`\sevensy`	Simboli matematici, sette punti.
`\fivesy`	Simboli matematici, cinque punti.
`\tenex`	Estensione ai simboli matematici, 10 punti.

49.10.6 Sottolineatura

La gestione della sottolineatura con TeX è separata da quella che riguarda la forma e la dimensione dei caratteri. Si tratta solo di una linea orizzontale che viene collocata al di sotto del testo, a una distanza che varia in funzione delle caratteristiche del testo, che si ottiene con la macro '**`\underbar`**':

```
\underbar{testo_da_sottolineare}
```

La linea con cui si ottiene questa sottolineatura crea implicitamente una scatola orizzontale (sezione 49.14) che non può essere suddivisa in più righe. Si osservi l'esempio seguente:

```
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla.

bla bla bla bla bla bla bla bla bla bla bla \underbar{bla
bla bla bla bla bla bla bla bla bla bla bla} bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla.
```

Dal momento che il testo sottolineato è molto lungo e si trova verso la fine di una riga, TeX non può separarlo e si crea un problema estetico molto spiacevole:

bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

bla bla bla bla bla bla bla bla bla bla bla <u>bla bla bla bla bla bla bla bla bla bla bla bla</u>
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla.

49.11 La pagina

In condizioni normali, la pagina a cui si fa riferimento con TeX è il tipo Lettera, larga 8,5 pollici e alta 11 pollici. Nell'ambito di questo spazio, il testo occupa normalmente un rettangolo di 6,5 pollici per 8,9 pollici, con un margine uguale di un pollice (salvo il margine inferiore che è di 1,1 pollici).

Al di sopra e al di sotto di questo rettangolo è possibile collocare una riga, con lo scopo normale di indicare il nome della sezione o il numero della pagina. In condizioni normali, in basso appare proprio il numero della pagina, ma può essere fatto sparire con l'istruzione '**`\nopagenumbers`**'.

Le dimensioni della pagina possono essere modificate, ma non si fa riferimento a una pagina e ai suoi margini; si interviene piuttosto sulla posizione di partenza, sulla larghezza (ovvero la giustezza) e sull'altezza dello spazio a disposizione dei paragrafi.

49.11.1 La parte centrale della pagina

La parte centrale della pagina è controllata da quattro variabili, rappresentate dalle parole di controllo '**`\hoffset`**', '**`\voffset`**', '**`\hsize`**' e '**`\vsize`**', le quali controllano rispettivamente le coordinate iniziali, la giustezza e l'altezza dell'area di scrittura.

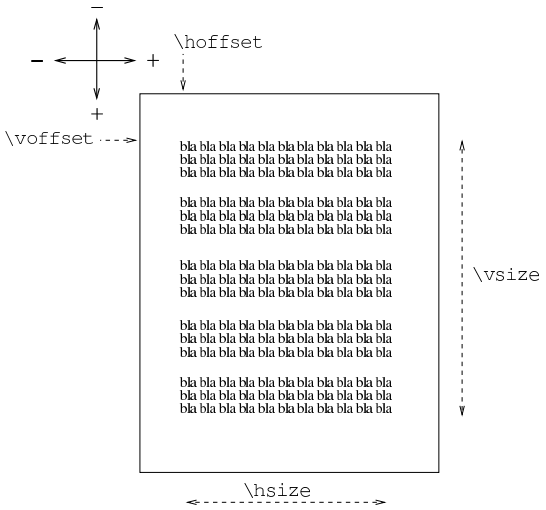
Tabella 49.118. Parole di controllo che regolano le dimensioni e la collocazione dell'area di scrittura centrale nella pagina.

Parola di controllo	Competenza	Valore predefinito
`\hoffset`	Posizione orizzontale iniziale	0
`\voffset`	Posizione verticale iniziale	0
`\hsize`	Giustezza, a partire da '**`\voffset`**'	6,5 pollici
`\vsize`	Altezza, a partire da '**`\hoffset`**'	8,9 pollici

È importante osservare che le coordinate zero di '**`\hoffset`**' e

‘**\voffset**’ rappresentano un punto in alto a sinistra del foglio, collocato a destra e in basso di un pollice rispetto all’angolo superiore sinistro del foglio stesso. In pratica, per indicare valori inferiori del margine superiore e di quello sinistro, si devono attribuire a queste parole di controllo dei valori negativi.

Figura 49.119. Area del testo nella pagina.



L’esempio seguente consente di verificare il valore predefinito che viene restituito da queste parole di controllo:

```
$\backslash$hoffset = \the\hoffset \par
$\backslash$voffset = \the\voffset \par
$\backslash$hsize = \the\hsize \par
$\backslash$vsize = \the\vsize \par
```

$\backslash\text{hoffset} = 0.0\text{pt}$
$\backslash\text{voffset} = 0.0\text{pt}$
$\backslash\text{hsize} = 398.33858\text{pt}$
$\backslash\text{vsize} = 643.20255\text{pt}$

I valori vengono mostrati in punti, ma si verifica facilmente la corrispondenza con quanto mostrato nella tabella 49.118.

È importante sottolineare che le coordinate di partenza rappresentano l’angolo superiore sinistro dell’area in cui si inserisce il testo; pertanto, l’altezza espressa dalla parola di controllo ‘**\vsize**’, si sviluppa in basso, a partire da quel punto.

Il testo che si scrive nel sorgente TeX parte dalle coordinate iniziali e, salvo l’uso di comandi appositi, viene impaginato automaticamente; quando lo spazio verticale viene esaurito, si passa automaticamente a un’altra pagina.

È importante sottolineare che le modifiche apportate alle coordinate iniziali e all’altezza del testo della pagina hanno effetto a partire dalla pagina corrente; inoltre, se i valori vengono cambiati più volte, contano le ultime modifiche che si possono ricondurre alla pagina corrente. Fa eccezione a questa regola la giustezza del testo, controllata dalla parola di controllo ‘**\hsize**’, che può essere modificata in qualunque momento, intervenendo a partire dal paragrafo in cui appare, senza coinvolgere il testo precedente nella stessa pagina.

Eventualmente, se prima di scrivere alcunché, si colloca un’istruzione come quella seguente,

```
\magnification=n
```

si ottiene un ingrandimento o una riduzione di tutte le dimensioni, compreso il corpo dei caratteri. In pratica, il numero ***n*** che si assegna a ‘**\magnification**’ è un valore che esprime la riduzione o l’ingrandimento in rapporto a 1000, come già avveniva con il ridimensionamento dei caratteri da stampa. In questo senso, al posto del numero si può usare anche la parola di controllo ‘**\magstep**’, come è già stato mostrato in precedenza:

```
\magnification=\magstep n
```

In questo caso, ***n*** esprime un valore intero, da zero a cinque.

La definizione di un ingrandimento o di una riduzione ha effetto su tutti i comandi che definiscono una dimensione; per esempio, se si usa un ingrandimento di 2000, pari al doppio, volendo indicare una dimensione di un centimetro, si ottengono in pratica due centimetri.

Quando si intende indicare una lunghezza esatta, che non possa essere ridimensionata, si usa la parola chiave ‘**true**’ davanti all’unità di misura. Per esempio, ‘**5cm**’ è una lunghezza adattabile, mentre ‘**5truecm**’, oppure ‘**5 true cm**’, indica sempre cinque centimetri.

In condizioni normali, TeX cerca di occupare tutto lo spazio orizzontale e tutto lo spazio verticale, giustificando orizzontalmente e verticalmente (ovvero allineando simultaneamente a sinistra e a destra, in alto e in basso). In particolare, l’allineamento verticale del testo viene controllato da due parole di controllo: ‘**\normalbottom**’ e ‘**\raggedbottom**’. Nel primo caso si ha la situazione «normale», ovvero l’allungamento del testo in modo da completare lo spazio verticale di ogni pagina, mentre nel secondo caso questo allungamento non ha luogo.

TeX mette a disposizione un’istruzione per il salto pagina anticipato che si ottiene con la parola di controllo ‘**\eject**’. Quando l’impaginazione è quella normale, ovvero quella corrispondente alla parola di controllo ‘**\normalbottom**’, l’inserimento di un salto pagina costringe TeX a fare in modo che il testo esistente, per quanto breve possa essere, finisca esattamente alla fine di ‘**\vsize**’. Per evitare di produrre degli allungamenti sgradevoli del testo, pur senza disabilitare globalmente la funzionalità di allineamento verticale, si può usare un comando per l’inserimento di uno spazio verticale allungabile in modo indefinito, il quale si ottiene con la parola di controllo ‘**\vfill**’. Pertanto, quando si vuole inserire un salto pagina si usa generalmente un’istruzione composta da entrambe le parole di controllo, come nell’esempio seguente:

```
\vfill \eject
```

Inizialmente è stato indicato l’uso dell’istruzione ‘**\bye**’ per concludere un sorgente TeX. In realtà, ‘**\bye**’ è una macroistruzione realizzata per concludere bene un testo, in cui si utilizza anche ‘**\vfill**’ per non allungare il contenuto della pagina stessa. L’istruzione primitiva per concludere il documento è invece ‘**\end**’.

Anche ‘**\eject**’ è una macro, che utilizza in pratica ‘**\break**’, dopo un passaggio in modalità verticale, in modo da dare a ‘**\break**’ il contesto corretto.

Eventualmente, esiste anche un’altra macro, oltre a ‘**\eject**’, il cui scopo è quello di garantire il salto pagina in ogni circostanza. Si tratta di ‘**\supereject**’.

49.11.2 Intestazione e fondo pagina

Oltre alla parte centrale della pagina, è possibile accedere a una riga di intestazione e a una riga alla base della pagina, le quali appaiono rispettivamente sullo spazio del margine superiore e del margine inferiore. Queste righe vengono annotate in due variabili apposite, a cui si accede con le parole di controllo ‘**\headline**’ e ‘**\footline**’:

```
\headline={intestazione}
```

```
\footline={fondo_pagina}
```

In condizioni normali, la variabile che contiene la riga di fondo pagina è già impostata con l'espressione '**\folio\pageno**', con la quale si estrae dal contatore '**\pageno**' il valore corrispondente al numero di pagina. Inoltre, per eliminare il fondo pagina predefinito in questo modo, si utilizza normalmente la parola di controllo '**\nopagenumbers**', come sinonimo di azzeramento, al posto di scrivere '**\footline={}**'.

Il testo che viene associato all'intestazione e al fondo pagina, risulta allineato a destra e a sinistra, simultaneamente. Tuttavia, attraverso l'uso della parola di controllo '**\hfil**', si riescono a ottenere degli effetti utili di allineamento. La parola di controllo '**\hfil**' ha la proprietà di inserire uno spazio allungabile, ma può essere utilizzata più volte, suddividendo equamente lo spazio della riga. Per esempio, scrivendo

```
\headline={sinistra \hfil centro \hfil destra}
```

si ottiene la suddivisione che si può vedere qui sotto:

sinistra centro destra

Quando si distingue tra pagine destre e sinistre, si può usare la struttura condizionale comandata dalla parola di controllo '**\ifodd**' per modificare l'aspetto delle intestazioni o dei fondo pagina. Nell'esempio seguente si vuole mettere il numero della pagina nell'intestazione, facendo in modo che appaia verso l'estremità esterna della pagina:

```
\headline={%
\ifodd \pageno
    {bla bla bla \hfil \the\pageno}
\else
    {\the\pageno \hfil bla bla bla}
\fi}
```

Nell'esempio vengono racchiuse le due alternative tra parentesi graffe, perché altrimenti '**\pageno**' creerebbe dei problemi con la prima ipotesi. Non sarebbe necessario il raggruppamento della seconda alternativa, ma qui lo si fa per motivi «estetici».

L'espressione dell'intestazione o del fondo pagina, viene espansa ogni volta, nel momento in cui serve. Tuttavia, in questo modo, le impostazioni riferite al carattere da stampa usato nel testo normale potrebbero influenzare l'aspetto di queste righe. Per sicurezza, conviene completare la definizione di intestazione e fondo pagina con l'indicazione del carattere da stampa, completo dell'informazione sulla sua dimensione, come nell'esempio seguente, in cui si richiede espressamente un carattere inclinato di 10 punti:

```
\headline={\tensl Questa \'e una bella intestazione}
```

49.11.3 Numerazione delle pagine

La numerazione delle pagine avviene in modo automatico, attraverso la gestione del contatore '**\pageno**'. In condizioni normali, la prima pagina ha il numero uno e a ogni salto pagina si ha un incremento di questo valore.

L'impostazione predefinita della riga che appare a fondo pagina, è tale per cui se il contatore contiene un valore positivo, si mostra un numero arabo normale, mentre se il valore è negativo, si mostra la pagina in numero romano. Per la precisione, ciò si ottiene attraverso la parola di controllo '**\folio**' che, senza bisogno di trasformazioni, restituisce il numero della pagina secondo questa convenzione.

In particolare, se il contatore '**\pageno**' contiene un valore positivo, ogni salto pagina continua a incrementarlo, mentre se ha un valore negativo, si ottiene un decremento, proprio in base alla logica della riga di fondo pagina predefinita.

Nel momento in cui si intende gestire il numero della pagina che appare nel fondo pagina, o nella riga di intestazione, non ha più importanza questa convenzione sul segno positivo o negativo; quello che conta è reinizializzare il contatore quando serve.

Il vero problema che si pone è quello di rappresentare le pagine attraverso numeri romani con lettere maiuscole. L'esempio seguente mostra la macro '**\Roman**' realizzata per ottenere la trasformazione in numero romano maiuscolo del suo argomento:

```
\def\Roman#1{\uppercase\expandafter{\romannumeral #1}}
```

La macro va usata ovviamente così:

```
\Roman{n}
```

49.11.4 Note a piè di pagina

Le note a piè di pagina sono gestite automaticamente attraverso la macro standard '**\footnote**', inserendole alla base della pagina, entro lo spazio stabilito da '**\vsize**'.

```
\footnote{simbolo}{testo_della_nota}
```

In pratica, la nota viene inserita nel testo, nel punto in cui si vuole che appaia il simbolo definito come primo parametro, poi la composizione colloca la nota dove più appropriato, preceduta sempre dallo stesso simbolo di riferimento.

Il simbolo usato per fare riferimento alla nota può essere scelto liberamente, tenendo conto che TeX non fa nulla per adattarlo. Pertanto, se l'intenzione è quella di usare il classico numero ad apice, occorre provvedere esplicitamente a sollevarlo, come nell'esempio seguente:

```
Bla bla\footnote{$^1$}{Questa è la mia prima nota a piè
pagina.} bla bla...
```

Generalmente, si numerano le note a piè pagina, ma TeX non offre un meccanismo automatico di numerazione. Tuttavia è facile rimediare attraverso la definizione di un contatore apposito:

```
\newcount\nota
\nota=0
\def\annotazione#1{\advance\nota by 1
                   \footnote{$^{\number\nota}$}{#1}}
```

Successivamente, invece di usare direttamente '**\footnote**', si può inserire la macro '**\annotazione**', la quale provvede da sola a incrementare il contatore '**\nota**', inserendo il numero corrispondente come apice. Eventualmente, quando si comincia con una sezione importante, che parte da una pagina nuova, può essere conveniente riazzerare il conteggio.

L'esempio mostrato però non è completo. Infatti, la macro '**\annotazione**' potrebbe essere inserita per qualche ragione in un gruppo isolato, per cui, l'incremento del contatore della pagina verrebbe perduto; inoltre, il tipo di carattere usato per le note viene ereditato dal punto in cui si trova l'inserimento. Per ovviare a questi inconvenienti, si può modificare la macro '**\annotazione**' nel modo seguente:

```
\def\annotazione#1{\global\advance\nota by 1
\footnote{$^{\number\nota}$}{\sevenrm #1}}
```

In questo modo, l'incremento del contatore diventa globale e si fissa il carattere da usare per le note in un tondo normale con un corpo da sette punti.

49.11.5 Oggetti fluttuanti

Nell'ambito del testo è possibile individuare delle zone che devono rimanere unite, nel senso che non possono apparire separate tra una pagina e la successiva, per cui si stabilisce una collocazione che non segue il flusso normale del testo. Questi ambienti sono introdotti da una parola di controllo particolare:

```
\midinsert
testo_e_altre_istruzioni
...
\endinsert
```

```
\topinsert
testo_e_altre_istruzioni
...
\endinsert
```

```
\pageinsert
testo_e_altre_istruzioni
...
\endinsert
```

Nel primo caso, quanto contenuto tra '**\midinsert**' e '**\endinsert**' viene collocato nella posizione in cui appare, se c'è lo spazio verticale sufficiente nella pagina, altrimenti viene spostato all'inizio della pagina successiva, lasciando che il testo che seguirebbe venga anticipato al suo posto nella pagina che termina.

Nel secondo caso, quanto contenuto tra '**\topinsert**' e '**\endinsert**' viene collocato all'inizio della pagina in cui si trova, anticipando eventualmente del testo che lo precede.

Nel terzo caso, quanto contenuto tra '**\pageinsert**' e '**\endinsert**' viene collocato in una pagina indipendente, che non può contenere altro.

Questi ambienti, eventualmente con l'uso di spazi verticali, permettono comunemente di riservare degli spazi per le figure.

49.12 Paragrafi

Il testo scritto con TeX viene reimpaginato in paragrafi, corrispondenti ai blocchi separati da almeno una riga bianca o vuota, oppure dalla parola di controllo '**\par**'.

Le caratteristiche comuni dei paragrafi vengono definite dai valori contenuti in alcune variabili, a cui si accede attraverso altrettante parole di controllo. Si tratta della giustezza, corrispondente a '**\hsize**', della dimensione del rientro della prima riga, corrispondente a '**\parindent**', della distanza tra le righe (tra la base delle righe), corrispondente a '**\baselineskip**', della distanza tra i paragrafi, corrispondente a '**\parskip**'.

Tabella 49.131. Parole di controllo che regolano o condizionano le caratteristiche dei paragrafi comuni.

Parola di controllo	Competenza	Condizione o valore predefinito
`\hoffset`	Posizione iniziale dei paragrafi nella pagina.	0
`\hsize`	Larghezza del paragrafo a partire da '**\hoffset**'.	6,5 pollici
`\parindent`	Rientro della prima riga.	20 punti
`\baselineskip`	Distanza tra la base di una riga e la base della riga successiva.	12 punti
`\parskip`	Distanza aggiuntiva tra i paragrafi.	0
`\raggedright`	Allinea il testo a sinistra.	allineato simultaneamente a sinistra e a destra
`\leftskip`	Rientro sinistro complessivo.	0
`\rightskip`	Rientro destro complessivo.	0

Si può provare a verificare i valori predefiniti con l'esempio seguente:

```
hoffset = \the\hoffset \par
hsize = \the\hsize \par
parindent = \the\parindent \par
baselineskip = \the\baselineskip \par
parskip = \the\parskip \par
leftskip = \the\leftskip \par
rightskip = \the\rightskip \par
```


hoffset = 0.0pt
hsize = 341.43306pt
parindent = 20.0pt
baselineskip = 12.0pt
parskip = 0.0pt plus 1.0pt
leftskip = 0.0pt
rightskip = 0.0pt

Sia '**\parskip**' che '**\baselineskip**' consentono l'attribuzione di lunghezze elastiche. Come si può osservare dall'esempio, il valore predefinito di '**\baselineskip**' non prevede elasticità, mentre '**\parskip**' consente un leggero allungamento. Si tenga presente che, anche se è stato indicato un allungamento di un solo punto, in pratica, nella composizione finale si possono ottenere allungamenti molto maggiori in caso di necessità.

Come si intende, in condizioni normali i paragrafi iniziano con un rientro, senza essere distanziati tra di loro. Se questo tipo di approccio è ciò che si desidera, quando eccezionalmente si vuole un paragrafo senza rientro, come si fa di solito dopo un titolo, basta usare la parola di controllo '**\noindent**' che ha effetto solo nel paragrafo in cui si trova.

Per modificare la posizione verticale e la larghezza dei paragrafi, invece di intervenire sul valore controllato dalla parola di controllo '**\hsize**' (su '**\hoffset**' non si può intervenire, perché riguarda la pagina intera), conviene usare invece '**\leftskip**' e '**\rightskip**', le quali si riferiscono rispettivamente a un rientro sinistro e a un rientro destro. In pratica, se invece si vuole estendere il paragrafo oltre il margine sinistro fissato da '**\hoffset**', oppure si vuole andare oltre il margine destro stabilito da '**\hsize**', basta usare valori negativi per i rientri. L'esempio seguente mostra il caso di un paragrafo rientrato a sinistra e a destra per scrivere un'annotazione che risalti agli occhi del lettore:

```
\leftskip=1cm
\rightskip=1cm
{\bf Attenzione}: bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla.

\leftskip=0cm
\rightskip=0cm
Questo \'e invece un paragrafo normale bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla.
```

Attenzione: bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Questo è invece un paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Si vede dall'esempio la necessità di intervenire successivamente per ripristinare i valori normali dei rientri.

I rientri definiti attraverso '**\leftskip**' e '**\rightskip**', possono essere gestiti anche attraverso la macroistruzione '**\narrower**', con cui si incrementa il loro valore di '**\parskip**'. In pratica, usando l'istruzione '**\narrower**' i margini del testo vengono aumentati di quanto contenuto in '**\parskip**'. Naturalmente, l'effetto di tale istruzione vale per tutti i paragrafi successivi, a meno che sia inserita in un raggruppamento; in alternativa si può intervenire nuovamente su '**\leftskip**' e '**\rightskip**' per ripristinare dei valori normali.

Un paragrafo risulta normalmente allineato simultaneamente a sinistra e a destra. Se si vuole eliminare l'allineamento destro, si utilizza la parola di controllo '**\raggedright**':

```
Questo \'e un paragrafo normale allineato a sinistra e a
destra, ma non tutti gli autori gradiscono questo sistema di
impaginazione orizzontale. Bla bla bla bla bla bla bla bla
```

```
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

{\raggedright In questo paragrafo, invece, l'allineamento
del testo al margine destro, non ha luogo. Purtroppo, non
c'\'e una parola di controllo opposta per ripristinare
l'allineamento normale, pertanto, occorre racchiudere
l'istruzione in un raggruppamento. Si osservi che il
paragrafo viene concluso prima della fine del
raggruppamento. Bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla. \par}

Questo \'e un altro paragrafo normale allineato a sinistra e
a destra. Bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla.
```

Questo è un paragrafo normale allineato a sinistra e a destra, ma non tutti gli autori gradiscono questo sistema di impaginazione orizzontale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

In questo paragrafo, invece, l'allineamento del testo al margine destro, non ha luogo. Purtroppo, non c'è una parola di controllo opposta per ripristinare l'allineamento normale, pertanto, occorre racchiudere l'istruzione in un raggruppamento. Si osservi che il paragrafo viene concluso prima della fine del raggruppamento. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Questo è un altro paragrafo normale allineato a sinistra e a destra. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

49.12.1 Spaziatura orizzontale automatica e separazione automatica dei paragrafi in righe

«

TeX utilizza una regola tipografica del passato, ormai condannata generalmente, secondo cui è necessario aumentare lo spazio che divide la fine di un periodo dall'inizio del successivo. Si pone rimedio a questa anomalia utilizzando l'istruzione '**`\frenchspacing`**'. Si osservi l'esempio seguente, il cui risultato di composizione viene ingrandito per permettere di scorgere meglio la differenza nella spaziatura orizzontale alla fine del primo periodo:

```
Prima frase. Seconda frase. \par
\frenchspacing
Prima frase. Seconda frase. \par
```

Prima frase. Seconda frase.

Prima frase. Seconda frase.

È bene sottolineare che '**`\frenchspacing`**' interviene a livello di paragrafi dal punto in cui viene utilizzato. Se successivamente si desidera ripristinare la spaziatura orizzontale «normale» per TeX, basta usare l'istruzione '**`\nonfrenchspacing`**'.

TeX impagina automaticamente i paragrafi, dividendo le righe dove ritiene più corretto, in base ai suoi calcoli, considerando le interruzioni di riga nel sorgente come equivalenti a uno spazio orizzontale, suddividendo eventualmente le parole in sillabe (come viene spiegato meglio in una sezione successiva). Tuttavia, ci sono spazi che fanno parte di un nome o che comunque si vuole evitare siano utilizzati per dividere in quel punto il testo. Per evitarlo si può inserire la tilde ('**`~`**') che rappresenta uno spazio non interrompibile. L'esempio seguente mostra il caso in cui si vuole evitare di separare il titolo da un nome:

```
Ing.~Bianchi
```

Quando non si usa la spaziatura orizzontale uniforme che si otterrebbe con '**`\frenchspacing`**', l'inserzione di questi spazi non interrompibili evita anche l'allargamento dello spazio quando, in situazioni come quella dell'esempio, sarebbe inopportuno. In alternativa, sempre allo scopo di ottenere uno spazio di ampiezza normale dopo un punto che non conclude una frase, si può usare il simbolo di controllo '**`\`**<*SP*>' (barra obliqua inversa, spazio).

49.12.2 Rientri particolari

«

Attraverso due variabili, a cui si accede per mezzo delle parole di controllo '**`\hangindent`**' e '**`\hangafter`**', è possibile organizzare un paragrafo in modo che una parte iniziale o finale delle righe che lo compongono sia rientrata rispetto al margine sinistro, oppure rispetto al margine destro.

Un valore positivo, esprimente una lunghezza, contenuto nella variabile controllata dalla parola di controllo '**`\hangindent`**', indica un rientro verso destra della stessa lunghezza rispetto al margine sinistro; al contrario, un valore negativo indica un rientro verso sinistra dal margine destro. Un valore positivo, esprimente una quantità di righe, contenuto nella variabile controllata dalla parola di controllo '**`\hangafter`**', indica dopo quante righe far intervenire il rientro dichiarato con '**`\hangafter`**'; invece, un valore negativo indica per quante righe iniziali deve valere il rientro. Si osservi l'esempio seguente:

```
\hangindent=5cm
\hangafter=2
Paragrafo in cui, dopo le prime due righe, si passa a un
rientro sinistro di cinque centimetri. Bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla.

\hangindent=-5cm
\hangafter=-2
Paragrafo in cui le prime due righe hanno un rientro destro
di cinque centimetri. Bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla.
```

La composizione che si ottiene potrebbe contenere degli errori estetici e degli errori nella separazione in sillabe delle parole. Questi problemi vengono trattati in seguito:

Paragrafo in cui, dopo le prime due righe, si passa a un rientro sinistro di cinque centimetri. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Paragrafo in cui le prime due righe hanno un rientro destro di cinque centimetri. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Dall'esempio non si vede, per cui è bene sottolineare che le parole di controllo '**`\hangindent`**' e '**`\hangafter`**' hanno effetto esclusivamente nel paragrafo in cui vengono utilizzate.

Un modo più complesso di definire un paragrafo è disponibile attraverso la parola di controllo '**`\parshape`**', da utilizzare secondo la sintassi seguente:

`\parshape` ***n_righe rientro_1 lunghezza_1*** [***rientro_2 lunghezza_2***]...

In pratica, '**`\parshape`**' permette di descrivere il rientro sinistro e la lunghezza delle prime ***n*** righe; in base alla quantità di righe indicate, altrettante devono essere le coppie delle indicazioni sulla lunghezza del rientro e sulla lunghezza della riga relativa. Si osservi l'esempio seguente:

```
\parshape 3 2cm 5cm 4cm 6cm 6cm 7cm
Paragrafo in cui, la prima riga \'e rientrata di due
centimetri dal margine sinistro ed \'e lunga cinque
centimetri, la seconda riga \'e rientrata di quattro
centimetri ed \'e lunga sei centimetri, la terza riga \'e
rientrata di sei centimetri ed \'e lunga sette centimetri.
Le righe successive sono impaginate come la terza riga.
```

```
Paragrafo normale. Bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
```

Paragrafo in cui, la prima riga
è rientrata di due centimetri dal margine
sinistro ed è lunga cinque centimetri, la seconda riga è rientrata di quattro centimetri ed è lunga sei centimetri, la terza riga è rientrata di sei centimetri ed è lunga sette centimetri. Le righe successive sono impaginate come la terza riga.

Paragrafo normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

È importante osservare che la prima riga risulta avere un rientro maggiore, pari al valore restituito da '**\parindent**'. Si osservi lo stesso esempio, con l'eliminazione del rientro normale dei paragrafi:

```
\noindent
\parshape 3 2cm 5cm 4cm 6cm 6cm 7cm
Paragrafo in cui, la prima riga \'e rientrata di due
centimetri dal margine sinistro ed \'e lunga cinque
centimetri, la seconda riga \'e rientrata di quattro
centimetri ed \'e lunga sei centimetri, la terza riga \'e
rientrata di sei centimetri ed \'e lunga sette centimetri.
Le righe successive sono impaginate come la terza riga.

Paragrafo normale. Bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
```

Paragrafo in cui, la prima riga è
rientrata di due centimetri dal margine
sinistro ed è lunga cinque centimetri, la seconda riga è rientrata di quattro centimetri ed è lunga sei centimetri, la terza riga è rientrata di sei centimetri ed è lunga sette centimetri. Le righe successive sono impaginate come la terza riga.

Paragrafo normale. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

49.12.3 Paragrafi etichettati

La macro standard '**\item**' consente di porre all'inizio di un paragrafo un'etichetta, cosa che si fa normalmente per realizzare degli elenchi:

```
\item{etichetta}
```

Il paragrafo che si ottiene ha una struttura particolare: L'etichetta viene inserita in modo che termini subito prima della posizione del rientro della prima riga, ovvero subito prima di '**\hoffset**'+'**\leftskip**'+'**\parindent**'; il testo successivo inizia dal rientro e nelle righe successive mantiene quel rientro iniziale. La parola di controllo '**\noindent**' non ha effetto.

Esiste anche un'altra parola di controllo simile, che ha lo scopo di generare un paragrafo dello stesso genere, rientrato del doppio del valore restituito da '**\parindent**'. Si tratta di '**\itemitem**', il cui utilizzo è quindi conveniente solo quando si usa normalmente il rientro della prima riga, altrimenti l'effetto generato è lo stesso di '**\item**'. Si osservino i due esempi seguenti che differiscono per la presenza o meno del rientro della prima riga.

```
Questo \'e un paragrafo normale che introduce un elenco, bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla:
\item{1)} questa \'e la prima voce dell'elenco;
\item{2)} questa \'e la seconda voce dell'elenco, che
contiene molte informazioni, bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla;
\item{3)} questa \'e la terza voce dell'elenco che si
articola in voci di livello inferiore:
\itemitem{a)} questa \'e la prima sotto-voce del terzo
punto;
\itemitem{b)} questa \'e la seconda sotto-voce del terzo
punto;
\item{4)} questo \'e la quarta e ultima voce dell'elenco
principale.
```

Questo è un paragrafo normale che introduce un elenco, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla:

1) questa è la prima voce dell'elenco;
2) questa è la seconda voce dell'elenco, che contiene molte informazioni, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla;
3) questa è la terza voce dell'elenco che si articola in voci di livello inferiore:
    a) questa è la prima sotto-voce del terzo punto;
    b) questa è la seconda sotto-voce del terzo punto;
4) questo è la quarta e ultima voce dell'elenco principale.

Come si può vedere, le macroistruzioni '**\item**' e '**\itemitem**' introducono un paragrafo nuovo, indipendentemente dal fatto che siano o meno separate dal testo precedente. Nell'esempio seguente si mostra cosa accade se si elimina il rientro della prima riga dei paragrafi:

```
\parindent=0pt
Questo \'e un paragrafo normale che introduce un elenco, bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla:
\item{1)} questa \'e la prima voce dell'elenco;
\item{2)} questa \'e la seconda voce dell'elenco, che
contiene molte informazioni, bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla;
\item{3)} questa \'e la terza voce dell'elenco che si
articola in voci di livello inferiore:
\itemitem{a)} questa \'e la prima sotto-voce del terzo
punto;
\itemitem{b)} questa \'e la seconda sotto-voce del terzo
punto;
\item{4)} questo \'e la quarta e ultima voce dell'elenco
principale.
```

Questo è un paragrafo normale che introduce un elenco, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla:

1) questa è la prima voce dell'elenco;
2) questa è la seconda voce dell'elenco, che contiene molte informazioni, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla;
3) questa è la terza voce dell'elenco che si articola in voci di livello inferiore:
a) questa è la prima sotto-voce del terzo punto;
b) questa è la seconda sotto-voce del terzo punto;
4) questo è la quarta e ultima voce dell'elenco principale.

Volendo approfondire un po' di più l'uso di queste macroistruzioni, si può osservare che '**\item**' assegna a '**\hangindent**' il valore di '**\parindent**'; inoltre assegna a '**\hangafter**' il valore uno. Al termine, tutto viene ripristinato, in modo da non coinvolgere il paragrafo successivo. Nello stesso modo interviene '**\itemitem**', assegnando però a '**\hangindent**' il doppio di '**\parindent**'.

49.12.4 Attuazione dei comandi riferiti ai paragrafi

Le istruzioni che definiscono l'aspetto del paragrafo, vengono prese in considerazione quando il paragrafo è stato analizzato completamente; pertanto, se in un paragrafo si modifica più volte un certo valore che ne controlla la forma, solo l'ultimo valore disponibile è quello che viene preso in considerazione. Si osservi l'esempio seguente:

```
\baselineskip=5cm
Questo paragrafo viene composto con una distanza tra la base
delle righe di 14 punti. Provare per credere. Bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla.
\baselineskip=14pt
```

Questo paragrafo viene composto con una distanza tra la base delle righe di 14 punti. Provare per credere. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

In pratica, all'inizio viene dichiarata una distanza di 5 cm, mentre alla fine, prima che sia concluso definitivamente il paragrafo, si di-

chiara nuovamente una distanza di soli 14 punti. L'ultima istruzione che modifica il valore, rimanendo nell'ambito del paragrafo, è quella che viene presa in considerazione.

Il problema può sembrare di poco conto; tuttavia, quando si usano i raggruppamenti per evitare di estendere l'effetto di un comando particolare anche ai paragrafi successivi, le cose si complicano un po'. Si osservi l'esempio seguente:

```
{\baselineskip=5cm Questo paragrafo che dovrebbe avere una
distanza tra la base delle righe di ben cinque centimetri,
ma qualcosa non funziona come si vorrebbe. Bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla.}
```

Questo paragrafo che dovrebbe avere una distanza tra la base delle righe di ben cinque centimetri, ma qualcosa non funziona come si vorrebbe. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Il raggruppamento limita l'effetto della modifica dello spazio tra le righe, controllato da '**`\baselineskip`**', all'interno del paragrafo. Tuttavia, il paragrafo finisce dopo la parentesi graffa di chiusura, ovvero quando viene ripristinato il valore precedente per la distanza tra le righe. Per risolvere il problema occorre concludere il paragrafo **prima** della parentesi graffa finale:

```
{\baselineskip=0.5cm Questo paragrafo ha effettivamente una
distanza tra la base delle righe di mezzo centimetro,
perch\'e prima della parentesi graffa finale si conclude
esplicitamente il paragrafo.\par}
```

Questo paragrafo ha effettivamente una distanza tra la base delle righe di mezzo centimetro, perché prima della parentesi graffa finale si conclude esplicitamente il paragrafo.

49.12.5 Tolleranza estetica

«

Quando la composizione che si genera non può rispettare gli spazi e le elasticità fissati, TeX calcola un indicatore, definibile come penalità estetica, per rappresentare la «bruttezza» del risultato che si ottiene. Per la precisione, si distinguono due casi: la necessità di introdurre una quantità di spazio eccessiva, tra le parole o tra le righe, oppure la necessità di andare leggermente oltre il margine.

Quando si superano le tolleranze orizzontali stabilite, si ottiene nella composizione finale un segno evidente sul margine destro, corrispondente a un rettangolino nero.

La parola di controllo '**`\hbadness`**' permette di definire il livello consentito per l'espansione delle righe. Il valore predefinito è 1000 e rappresenta un livello qualitativo molto buono. Elevando questo valore, si ottengono meno segnalazioni; si può arrivare fino a un massimo di 10000, con cui si esclude qualunque segnalazione di cattiva qualità estetica dovuta a un allargamento eccessivo delle righe.

Una riga può eccedere il limite del margine destro per una lunghezza massima pari al valore che si ottiene dalla parola di controllo '**`\hfuzz`**'. Se anche questo limite viene superato, si crea un altro tipo di problema estetico, controllato dalla parola di controllo '**`\tolerance`**'. Anche in questo caso, se si assegna alla tolleranza in eccesso un valore pari a 10000, si annullano tutte le segnalazioni di errore.

La qualità verticale è controllata dal valore contenuto in '**`\vbadness`**' e nello stesso modo esiste la possibilità di eccedere l'altezza del testo del valore di '**`\vfuzz`**'.

```
hbadness = \the\hbadness \par
hfuzz = \the\hfuzz \par
vbadness = \the\vbadness \par
vfuzz = \the\vfuzz \par
tolerance = \the\tolerance \par
```

hbadness = 1000
hfuzz = 0.1pt
tolerance = 200
vbadness = 1000
vfuzz = 0.1pt

L'esempio precedente consente di verificare i valori predefiniti abbinati alla tolleranza estetica del testo, nella sua distribuzione orizzontale e verticale. Le istruzioni che seguono, consentono di annullare tutte le segnalazioni di carenze estetiche, consentendo però di generare un risultato pessimo:

```
\hbadness=10000
\vbadness=10000
\tolerance=10000
```

In alternativa, si possono lasciare invariati i valori di tolleranza estetica, eliminando semplicemente l'indicazione del problema nella composizione finale. L'inserimento del rettangolo nero è controllato dalla parola di controllo '**`\overfullrule`**', a cui si assegna la grandezza della linea con cui rappresentare questo rettangolo. Indicando un valore nullo, si elimina tutto questo:

```
\overfullrule=0pt
```

È utile sapere interpretare i messaggi di avvertimento che vengono generati da TeX durante la composizione di testo con carenze estetiche. La tabella 49.161 riepiloga le varie situazioni.

Tabella 49.161. Messaggi di errore riferiti a carenze estetiche.

Tipo di messaggio	Significato
'**`Underfull \hbox`**'	La spaziatura orizzontale è eccessiva.
'**`Overfull \hbox`**'	Lo spazio orizzontale a disposizione è stato superato.
'**`Underfull \vbox`**'	La spaziatura verticale è eccessiva.
'**`Overfull \vbox`**'	Lo spazio verticale a disposizione è stato superato.
'**`badness`** ***n***'	Livello di carenza estetica: 10000 è il massimo a essere annotato.

49.12.6 Sillabazione

«

In condizioni normali, TeX gestisce la separazione in sillabe per la lingua inglese. Eventualmente, è possibile indicare delle eccezioni attraverso l'uso della parola di controllo '**`\hyphenation`**':

`\hyphenation{`***parola_suddivisa***`...}`

Per esempio, per annotare la suddivisione di «cartellone» e «manuale», si può indicare l'istruzione seguente:

```
\hyphenation{car-tel-lo-ne ma-nu-a-le}
```

È possibile anche usare una separazione facoltativa attraverso il simbolo di controllo '**`\-`**' all'interno di una parola in una posizione precisa. Tuttavia, l'uso di queste istruzioni, annulla la possibilità di formarsi un legato. Per esempio, scrivendo '**`dif\-fi\-ci\-le`**' si impedisce la formazione del legato tra la coppia di lettere «f».

La frequenza della separazione delle parole alla fine delle righe può essere regolata tramite il valore attribuito a una variabile, cui si accede attraverso la parola di controllo '**`\pretolerance`**': Valori bassi producono una separazione frequente, mentre valori elevati riducono l'uso della suddivisione delle parole. In particolare, se si assegna il valore 10000, non avviene alcuna separazione in sillabe.

`\pretolerance=`***n***

Il valore predefinito di '**`\pretolerance`**' è 100.

È importante sottolineare che '**`\pretolerance`**' interviene a livello di paragrafo, per cui non è possibile isolare un gruppo all'interno di un paragrafo, perché queste modifiche non avrebbero effetto.

Nella sezione 49.1 è descritto brevemente come abilitare la separazione delle parole in sillabe per una lingua differente. Supponendo che la sillabazione italiana sia abbinata al numero 1, è sufficiente iniziare il proprio documento con l'istruzione seguente:

```
\language 1
```

49.12.7 Spaziature verticali

È già stato descritto il significato delle parole di controllo '**\baselineskip**' e '**\parskip**'. Entrambe le variabili a cui fanno riferimento queste parole di controllo possono contenere una lunghezza elastica, anche se normalmente '**\baselineskip**' non prevede tolleranze.

Oltre allo spazio verticale tra i paragrafi, controllato dal valore restituito dalla parola di controllo '**\parskip**', si possono inserire degli spazi aggiuntivi[8] attraverso due parole di controllo alternative, '**\vskip**' e '**\vglue**':

```
\vskip lunghezza_elastica
```

```
\vglue lunghezza_elastica
```

Il secondo tra i due tipi di inserimento, corrispondente alla parola di controllo '**\vglue**', è il più semplice, perché inserisce lo spazio verticale in modo incondizionato: se non c'è lo spazio sufficiente nella pagina in cui appare, viene messo integralmente all'inizio della pagina successiva. Invece, nel caso di '**\vskip**', lo spazio viene inserito effettivamente solo se c'è lo spazio sufficiente nella pagina in cui appare, provocando un salto pagina se questo spazio manca, ma in tal caso non viene più inserito all'inizio della pagina successiva.

È importante osservare che in entrambe le situazioni, se lo spazio alla fine della pagina non basta, si provoca un salto pagina. Se è attiva la modalità di allineamento verticale del testo, corrispondente alla parola di controllo '**\normalbottom**', ciò significa che il testo precedente viene allungato per completare la pagina, così come avviene quando si inserisce la parola di controllo '**\eject**'.

A titolo informativo, vale la pena di vedere in che modo si possa ottenere l'inserimento di uno spazio incondizionato, come fa '**\vglue**', ma utilizzando solo '**\vskip**':

```
\null\par
\vskip …
```

In pratica, l'istruzione '**\null**' serve a iniziare un paragrafo (passando così in modalità orizzontale). Il paragrafo deve essere terminato, quindi si può inserire '**\vskip**' con l'indicazione dello spazio da saltare. L'istruzione '**\null**' corrisponde a un carattere nullo, come un gruppo vuoto, che però permette di iniziare un paragrafo (viene descritta in seguito).

È già stato fatto osservare che il valore predefinito che si ottiene dall'espansione di '**\parskip**' è '**0pt plus 1pt**'. Volendo mantenere la coerenza con la spaziatura tra i paragrafi, quando si usano le parole di controllo '**\vskip**' e '**\vglue**', conviene mantenere la stessa elasticità in espansione, con l'aggiunta eventuale di un'elasticità equivalente in contrazione:

```
\vskip lunghezza plus 1pt minus 1pt
```

```
\vglue lunghezza plus 1pt minus 1pt
```

In effetti, TeX offre già delle parole di controllo che si traducono in salti standardizzati. Si tratta di '**\smallskipamount**', '**\medskipamount**' e '**\bigskipamount**'. Si può sperimentare facilmente a cosa corrispondono:

```
smallskipamount = \the\smallskipamount \par
medskipamount = \the\medskipamount \par
bigskipamount = \the\bigskipamount \par
```

smallskipamount = 3.0pt plus 1.0pt minus 1.0pt
medskipamount = 6.0pt plus 2.0pt minus 2.0pt
bigskipamount = 12.0pt plus 4.0pt minus 4.0pt

Queste parole di controllo sono fatte per essere sostituite agli argomenti delle istruzioni che provocano un avanzamento verticale. Per esempio,


```
\vskip\smallskipamount
```

provoca quello che viene considerato convenzionalmente un piccolo avanzamento verticale.

Per semplificare ulteriormente la gestione dei salti verticali, con la parola di controllo '**\vskip**', sono disponibili delle macro che richiamano direttamente l'entità dello spazio da inserire. Si tratta di '**\smallskip**', '**\medskip**' e '**\bigskip**', corrispondenti in pratica a '**\vskip\smallskipamount**', '**\vskip\medskipamount**' e '**\vskip\bigskipamount**'.

Quando si vuole indicare uno salto verticale di lunghezza indefinita, si utilizzano le parole di controllo '**\vfil**' e '**\vfill**'. Entrambe generano un salto dal punto in cui si trovano, fino alla fine della pagina, qualsiasi sia la distanza, ma se la parola di controllo appare più volte, lo spazio necessario viene distribuito equamente. Si osservi l'esempio seguente in cui si simula una pagina di piccole dimensioni:

```
\hsize=4cm
\vsize=7cm
Questa \'e una pagina fittizia in cui il testo ha a
disposizione solo quattro per sette centimetri. \par
\vfil
I paragrafi sono spaziati verticalmente in modo da riempire
la pagina. \par
\vfil
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla. \par
\vfil
Fine della pagina. \par
\eject
```

Questa è una pagina fittizia in cui il testo ha a disposizione solo quattro per sette centimetri.

I paragrafi sono spaziati verticalmente in modo da riempire la pagina.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Fine della pagina.

La stessa cosa si potrebbe ottenere sostituendo le parole di controllo '**\vfil**' con '**\vfill**':

```
\hsize=4cm
\vsize=7cm
Questa \'e una pagina fittizia in cui il testo ha a
disposizione solo quattro per sette centimetri. \par
\vfill
I paragrafi sono spaziati verticalmente in modo da riempire
la pagina. \par
\vfill
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla. \par
\vfill
Fine della pagina. \par
\eject
```

Tuttavia, '**\vfill**' è più elastico rispetto a '**\vfil**'. Si osservi cosa accade se si inserisce un solo '**\vfill**' tra altri '**\vfil**':

```
\hsize=4cm
\vsize=7cm
Questa \'e una pagina fittizia in cui il testo ha a
disposizione solo quattro per sette centimetri. \par
\vfil
I paragrafi sono spaziati verticalmente in modo da riempire
la pagina. \par
\vfill
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla. \par
\vfil
Fine della pagina. \par
\eject
```

Questa è una pagina fittizia in cui il testo ha a disposizione solo quattro per sette centimetri.

I paragrafi sono spaziati verticalmente in modo da riempire la pagina.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Fine della pagina.

Esiste un altro tipo di spazio verticale elastico, controllato dalla parola di controllo '**\vss**'. Si ottiene un risultato molto simile a '**\vfil**', con la differenza che in questo caso viene intesa esplicitamente anche un'elasticità in contrazione di livello equivalente.

Le parole di controllo '**\vfil[l]**' e '**\vss**' corrispondono in pratica alle definizioni seguenti:

```
\vfil  | \vskip 0pt plus 1fil
\vfill | \vskip 0pt plus 1fill
\vss   | \vskip 0pt plus 1fil minus 1fil
```

49.13 Righe

«

Il testo di una riga viene generato posando le lettere su una linea invisibile che corrisponde alla base della stessa. Nella sezione 47.2 è affrontato in generale l'argomento relativo alle dimensioni di un carattere. Per quanto riguarda TeX le cose si semplificano come si vede nella figura 49.173.

Figura 49.173. Le dimensioni del carattere secondo TeX.

Le righe di un paragrafo sono distanziate, da una basa alla successiva, della distanza fissata con '**\baselineskip**'. Ma questo non può impedire che si formino degli accavallamenti, soprattutto se si ingrandisce il corpo del carattere. Pertanto, entra in gioco il valore contenuto nella variabile '**\lineskiplimit**' che rappresenta l'interlinea minima, ovvero lo spazio minimo tra la profondità massima della riga precedente e l'altezza massima della riga successiva. Se si supera questo valore, viene inserita un'interlinea pari alla lunghezza contenuta nella variabile '**\lineskip**'. Si osservi l'esempio seguente in cui a un certo punto viene cambiato il corpo del carattere, per dimostrare l'adattamento che si ottiene nella distanza tra le righe:

```
\font\prova=cmss17 scaled 2000

Questo \'e un paragrafo di prova in cui, a un certo punto,
si usa un carattere pi\'u grande del solito, provocando un
aumento automatico della distanza tra le righe, {\prova per
evi\-ta\-re accavallamenti}. Bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla. \par
```

Questo è un paragrafo di prova in cui, a un certo punto, si usa un carattere più grande del solito, provocando un aumento automatico della distanza tra le righe, per evitare accavalla-menti. Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

È possibile sollevare o abbassare una parte del testo, attraverso le istruzioni '**\raise**' e '**\lower**'. Queste devono essere seguite da una «scatola», contenente ciò che si vuole sollevare o abbassare rispetto alla base della riga in cui ci si trova. Il concetto di scatola viene spiegato nella sezione 49.14.

\raise ***altezza scatola***

\lower ***altezza scatola***

L'esempio seguente mostra due situazioni, in cui si sposta una parte del testo in alto e in basso rispetto alla base normale delle righe:

```
Nel testo c'\'e una parte leggermente sollevata di
\raise 5pt \hbox{cinque punti} e un'altra parte
leggermente abbassata di \lower 3pt \hbox{tre punti}
tipografici.
```

Nel testo c'è una parte leggermente sollevata di cinque punti e un'altra parte leggermente abbassata di tre punti tipografici.

Quando una riga singola ha una sua propria autonomia nel testo che si vuole scrivere, si possono utilizzare una serie di parole di controllo per gestirne la collocazione e l'aspetto.

La prima cosa da analizzare è la possibilità di interrompere il flusso normale di un paragrafo, imponendo di iniziare una riga nuova. Ciò si ottiene con la parola di controllo '**\break**':

```
Questo \'e un paragrafo normale, contenente alcune
informazioni, bla bla bla bla bla bla bla bla bla bla
bla,\break ma questa riga deve apparire sempre da sola, pur
senza dover iniziare un paragrafo nuovo.
```

Questo è un paragrafo normale, contenente alcune informazioni, bla bla bla bla bla bla bla bla bla bla bla bla,
ma questa riga deve apparire sempre da sola, pur senza dover iniziare un paragrafo nuovo.

L'esempio mostra proprio una situazione di questo tipo, dove si può osservare che il testo della riga incompleta viene allungato per raggiungere il margine destro. Per evitare che succeda questo, si utilizza uno spazio orizzontale allungabile, che può essere ottenuto a scelta da tre parole di controllo: '**\hfil**', '**\hfill**' e '**\hfilll**'. Così come avviene negli spazi verticali, la prima di queste parole di controllo ha un'elasticità minore rispetto alle altre, così come la seconda ha un'elasticità maggiore della prima e minore della terza. Generalmente, si usa '**\hfill**' prima di '**\break**', come si vede dall'esempio seguente:

```
Questo \'e un paragrafo normale, contenente alcune
informazioni, bla bla bla bla bla bla bla bla bla bla
bla,\hfill\break ma questa riga deve apparire sempre da
sola, pur senza dover iniziare un paragrafo nuovo.
```

Questo è un paragrafo normale, contenente alcune informazioni, bla bla bla bla bla bla bla bla bla bla bla,
ma questa riga deve apparire sempre da sola, pur senza dover iniziare un paragrafo nuovo.

Quando si vuole controllare una riga completamente autonoma, si possono usare le macroistruzioni '**\leftline**', '**\rightline**', '**\centerline**' e '**\line**'. La prima allinea a sinistra il testo, la seconda lo allinea a destra, la terza lo allinea al centro, l'ultima lo stira in modo da allinearlo simultaneamente a sinistra e a destra. Si osservi l'esempio:

```
\leftline{Questa riga \'e allineata a sinistra.}
\centerline{Questa riga \'e allineata al centro.}
\rightline{Questa riga \'e allineata a destra.}
\line{Questa riga \'e allineata a sinistra e a destra.}
```

Questa riga è allineata a sinistra.
Questa riga è allineata al centro.
Questa riga è allineata a destra.
Questa riga è allineata a sinistra e a destra.

Si potrebbe essere portati a pensare che la macroistruzione '**\line**' serva a poco, perché l'effetto che si ottiene è abbastanza sgradevole. In realtà, la possibilità di inserire degli spazi orizzontali allungabili consente di creare degli effetti molto utili. Si osservi l'esempio seguente in cui si fa uso di '**\hfil**' per separare degli allineamenti indipendenti:

```
\line{Sinistra \hfil Centro \hfil Destra}
```

Sinistra Centro Destra

In particolare, '**\headline**' e '**\footline**' funzionano nello stesso modo, perché sono controllate in pratica dalla parola di controllo '**\line**'.

Si osservi che la definizione delle macroistruzioni '**\[left|center|right]line**' isola il testo in un raggruppamento, come se fosse racchiuso tra parentesi graffe.

49.13.1 Spazi orizzontali

È possibile inserire degli spazi orizzontali ben definiti attraverso la parola di controllo '**\hskip**', a cui si assegna l'indicazione di una lunghezza elastica:

```
\hskip lunghezza_elastica
```

Come nel caso di '**\vskip**', l'indicazione di un'elasticità in estensione, consente in pratica di allargare lo spazio in modo indefinito.

L'esempio seguente mostra diversi casi di utilizzo di '**\hskip**', in una situazione in cui la riga che lo contiene viene allineata simultaneamente a sinistra e a destra:

```
\line{Riga allineata a \hskip 2cm sinistra e a destra.}
\line{Riga allineata a \hskip 2cm plus 1pt sinistra e a destra.}
\line{Riga allineata \hskip 2cm plus 1pt a sinistra \hfil e a destra.}
\line{Riga allineata \hskip 0pt plus 1fil a sinistra \hfil e a destra.}
\line{Riga allineata \hskip 0pt plus 2fil a sinistra \hfil e a destra.}
```

Riga allineata a sinistra e a destra.
Riga allineata a sinistra e a destra.
Riga allineata a sinistra e a destra.
Riga allineata a sinistra e a destra.
Riga allineata a sinistra e a destra.

Esiste un altro tipo di spazio orizzontale molto elastico, controllato dalla parola di controllo '**\hss**'. Si ottiene un risultato molto simile a '**\hfil**', con la differenza che in questo caso viene intesa esplicitamente anche un'elasticità in contrazione di livello equivalente.

Le parole di controllo '**\hfil[l[l]]**' e '**\hss**' corrispondono in pratica alle definizioni seguenti:

```
\hfil  | \hskip 0pt plus 1fil
\hfill | \hskip 0pt plus 1fill
\hfilll | \hskip 0pt plus 1filll
\hss | \hskip 0pt plus 1fil minus 1fil
```

Esiste infine un gruppo di parole di controllo che consentono di inserire spazi orizzontali rigidi o poco elastici, corrispondenti all'elenco seguente:

- `\enspace`
  rappresenta uno spazio orizzontale rigido di mezzo quadratone;
- `\enskip`
  rappresenta uno spazio orizzontale leggermente elastico di mezzo quadratone;
- `\quad`
  rappresenta uno spazio orizzontale leggermente elastico di un quadratone;
- `\qquad`
  rappresenta uno spazio orizzontale leggermente elastico di due quadratoni;

- `\thinspace`
  in modalità orizzontale, introduce uno spazio rigido molto sottile, pari a un sesto di quadratone;
- `\negthinspace`
  in modalità orizzontale, introduce uno spazio negativo rigido molto sottile, pari a un sesto di quadratone.

Se si intende iniziare un paragrafo con un rientro prodotto da un'istruzione '**\hskip**', o altre che comunque producono uno spazio orizzontale determinato, occorre tenere presente la presenza del rientro definito da '**\parindent**'. In pratica, se si vuole iniziare un paragrafo con un rientro iniziale di due centimetri usando l'istruzione '**\hskip 2cm**', occorre fare in questo modo:

```
\noindent \hskip 2cm Bla bla bla...
```

Diversamente, il rientro finale risulterebbe aumentato del valore di '**\parindent**'.

Un'ultima cosa da considerare sono gli spazi orizzontali della stessa ampiezza di un certo carattere. Quando per qualche ragione si ha la necessità di incolonnare delle informazioni, soprattutto dati numerici, può essere utile la macroistruzione '**\phantom**', la quale genera effettivamente lo spazio orizzontale equivalente al carattere indicato:

```
\phantom{x}
```

Anche se l'utilizzo più importante riguarda i numeri, viene proposto qualcosa di diverso, per consentire di verificare visivamente che lo spazio viene calcolato in modo corretto:

```
\font\prova=cmss12
\prova
Ciao ciao ciao ciao \par
\hskip 0pt\phantom{C}iao ciao c\phantom{i}ao ciao \par
```

Ciao ciao ciao ciao
iao ciao c ao ciao

Come si vede, si utilizza un carattere senza grazie, in modo da poter verificare che l'inserzione dello spazio corrispondente a una lettera «i» viene fatta correttamente, così come è corretto lo spazio corrispondente alla lettera «C» maiuscola. L'istruzione iniziale '**\hskip 0pt**' serve a imporre che si tenga conto dello spazio iniziale, perché altrimenti verrebbe perduto nell'inizio del paragrafo.

49.13.2 Spazio rigido

Lo spazio rigido per definizione viene richiesto attraverso l'istruzione '**\kern**' che richiede semplicemente l'indicazione della distanza da compiere:

```
\kern lunghezza
```

Lo spazio in questione può essere orizzontale o verticale, in base al contesto. Pertanto, se ci si trova in modalità orizzontale, si ottiene uno spazio orizzontale; diversamente si ottiene uno spazio verticale. Per questo, quando si vuole scrivere una macroistruzione con cui iniziare sicuramente la modalità orizzontale e inserire quindi uno spazio rigido, si usa spesso '**\leavevmode**' per questo scopo.

Lo spazio rigido viene usato a volte in modo negativo, per avvicinare delle lettere o altri simboli, garantendo che non avvenga la separazione della riga in quel punto. Infatti, gli spazi rigidi che si ottengono con '**\kern**' non sono separabili. L'esempio tipico di utilizzo di '**\kern**' per avvicinare le lettere è quello della macroistruzione '**\TeX**', dichiarata nel modo seguente:

```
\def\TeX{T\kern -.1667em\lower .5ex\hbox {E}\kern -.125emX}
```

Come si vede, non c'è alcun bisogno di richiedere espressamente il passaggio alla modalità orizzontale, perché le istruzioni '**\kern**' appaiono dentro un testo che ha già fatto questo in ogni caso.

49.13.3 Linee guida

Per realizzare delle linee guida orizzontali, allo scopo di accompagnare la lettura verso qualcosa, si possono usare delle parole di controllo che contengono implicitamente un'elasticità in estensione molto elevata:

```
\hrulefill
```

```
\dotfill
```

```
\leftarrowfill | \rightarrowfill
```

Nel primo caso si ottiene una linea continua che unisce due parti di un testo separato; nel secondo si ottiene invece una guida fatta di puntini; nel terzo caso si genera una linea continua con una piccola freccia all'estremità sinistra o all'estremità destra.

Tuttavia, oltre a questi casi di guide abbastanza comuni, è possibile definire le proprie con una tra le istruzioni seguenti:

```
\leaders\hbox [to distanza] {espressione_guida} \hfill
```

```
\cleaders\hbox [to distanza] {espressione_guida} \hfill
```

```
\xleaders\hbox [to distanza] {espressione_guida} \hfill
```

La differenza tra le parole di controllo '**\leaders**', '**\cleaders**' e '**\xleaders**', sta nel fatto che la seconda dovrebbe centrare la linea guida, mentre l'ultima dovrebbe estenderla in modo da usare tutto lo spazio a disposizione. In pratica è difficile comprendere bene la differenza da un punto di vista visivo, per cui si utilizza in genere solo il primo tipo, salvo provare gli altri per verificare l'effetto finale che si può ottenere in alternativa.

Generalmente, nell'espressione che si utilizza per creare la guida si inseriscono parole di controllo per elasticizzare la distanza dei simboli utilizzati.

È importante sottolineare che TeX, dopo la parola di controllo '**\[c|x]leader**' vuole necessariamente una scatola '**\hbox**' e alla fine vuole uno spazio elastico di tipo '**\hfil[l[l]]**'. Nel modello sintattico non è stata mostrata la possibilità di usare diversi spazi orizzontali elastici, perché in pratica ci si limita sempre solo a usare '**\hfill**'.

Nel caso si utilizzi '**\hbox**' con l'indicazione della larghezza, l'effetto che si ottiene è quello di distanziare precisamente gli elementi simbolici che compongono la linea guida. In questo senso, l'elasticità che gli si attribuisce all'interno del raggruppamento, serve a collocarli correttamente nella scatola.

L'esempio seguente mostra diversi casi di utilizzo di queste linee guida messi a confronto. In particolare, si creano delle linee guida speciali, in cui si usa l'asterisco come simbolo ripetuto:

```
linea semplice: \hrulefill fine della linea \par
puntini: \dotfill fine dei puntini \par
linea con freccia a destra: \rightarrowfill fine della linea \par
linea con freccia a sinistra: \leftarrowfill fine della linea \par
asterischi normali: \leaders\hbox {*} \hfill fine degli asterischi \par
asterischi distanziati: \leaders\hbox to 1cm {\hss * \hss} \hfill
 fine degli asterischi \par
asterischi normali: \cleaders\hbox {*} \hfill fine degli asterischi \par
asterischi distanziati: \cleaders\hbox to 1cm {\hss * \hss} \hfill
 fine degli asterischi \par
asterischi normali: \xleaders\hbox {*} \hfill fine degli asterischi \par
asterischi distanziati: \xleaders\hbox to 1cm {\hss * \hss} \hfill
 fine degli asterischi \par
```


linea semplice: _____ fine della linea
puntini: ..... fine dei puntini
linea con freccia a destra: ——→ fine della linea
linea con freccia a sinistra: ←—— fine della linea
asterischi normali: ********** fine degli asterischi
asterischi distanziati: * * * * fine degli asterischi
asterischi normali: ********** fine degli asterischi
asterischi distanziati: * * * * fine degli asterischi
asterischi normali: ********** fine degli asterischi
asterischi distanziati: * * * * fine degli asterischi

Volendo creare una macroistruzione per una linea guida particolare, si può procedere come nell'esempio seguente, in cui si riprende uno dei casi già presentati sopra:

```
\def\asterischi{\leaders\hbox to 1cm {\hss * \hss} \hfill}
```

49.13.4 Linee

TeX consente di tracciare delle linee orizzontali o verticali, attraverso le parole di controllo '**\hrule**' e '**\vrule**'. La sintassi per queste istruzioni può essere abbastanza articolata:

```
\hrule [width larghezza] [height altezza] [depth profondità]
```

```
\vrule [width larghezza] [height altezza] [depth profondità]
```

La differenza più importante tra le due parole di controllo sta nel fatto che '**\hrule**' introduce implicitamente una separazione di paragrafi, mentre '**\vrule**' no.

Quando '**\hrule**' viene usato senza argomenti, si ottiene una linea orizzontale, appoggiata sulla base della riga, spessa 0,4 punti e larga quanto lo spazio a disposizione per il testo, ovvero quanto definito da '**\hsize**'; nello stesso modo, '**\vrule**' senza argomenti genera una linea verticale, larga 0,4 punti, che parte dalla base della riga e si innalza per l'altezza massima del contesto in cui si trova. Si osservi l'esempio:

```
Prima di una linea orizzontale; \hrule{} dopo una linea
orizzontale. \par
Prima di una linea verticale; \vrule{} dopo una linea
verticale. \par
```

Prima di una linea orizzontale;
dopo una linea orizzontale.
Prima di una linea verticale; | dopo una linea verticale.

Quando si usano gli argomenti con cui si possono controllare le caratteristiche di queste linee, è meglio pensare a dei rettangoli, dove l'unica differenza che conta è il fatto che '**\hrule**' conclude un paragrafo. Seguono altri esempi a questo proposito:

```
Linea orizzontale lunga 10 cm e alta 1 punto: \hrule width
10cm height 1pt \par
Linea verticale alta 1 cm e larga 2 punti: \vrule height 1cm
width 2pt \par
Rettangolino alto 5 mm e largo 3 mm: \vrule height 5mm width
3mm \par
Linea orizzontale nel testo: \vrule height 0pt depth 1pt
width 10cm \par
```

Linea orizzontale lunga 10 cm e alta 1 punto:
Linea verticale alta 1 cm e larga 2 punti:
Rettangolino alto 5 mm e largo 3 mm:
Linea orizzontale nel testo:

49.14 Scatole

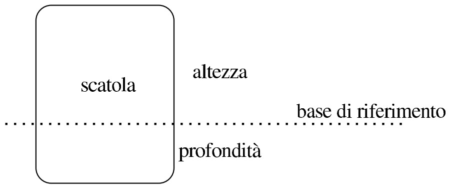
TeX tratta tutti gli oggetti da collocare nella composizione come tanti rettangoli, o scatole, di dimensione appropriata, distanziati in base a delle regole determinate. In questo senso, gli spazi che si inseriscono nel file sorgente, corrispondenti al carattere <*SP*>, comprese le tabulazioni orizzontali (il carattere <*HT*>), assieme alle righe vuote e a quelle bianche, sono solo la separazione logica delle istruzioni (intendendo anche il testo come istruzioni), pertanto non conta la loro quantità, essendo sufficiente che siano presenti dove serve.

Attraverso alcune parole di controllo è possibile creare della scatole di tipo orizzontale o verticale, per ottenere dei comportamenti par-

ticolari che vanno al di fuori della gestione normale dei paragrafi e delle righe.

Una scatola ha tre misure che ne definiscono la forma: una larghezza, un'altezza che parte dalla base della riga di riferimento e una profondità, che è rappresentata dall'altezza della scatola che si espande sotto la base di riferimento.

Figura 49.199. Dimensioni di una scatola.



Una scatola orizzontale è un rettangolo in cui si inserisce una sola riga di testo, dove l'altezza e la profondità dipendono dalle dimensioni del testo stesso. Una scatola verticale è invece un'area in cui il testo può scomporsi in più righe; inoltre, se la scatola viene inserita in una riga di testo, questa conclude il paragrafo.

Si utilizzano anche delle scatole vuote. Una scatola orizzontale vuota può servire per inserire un testo sporgente, a sinistra o a destra rispetto alla stessa; inoltre, può servire per imporre una certa altezza e profondità al testo della riga in cui viene inserita.

Una scatola orizzontale si definisce generalmente con la parola di controllo '**`\hbox`**' che può essere usata in due modi differenti:

`\hbox {`***espressione_stringa***`}`

`\hbox to` ***larghezza*** `{`***espressione_stringa***`}`

Nel primo caso si definisce una scatola orizzontale, contenente il testo che si ottiene dall'espansione dell'espressione racchiusa nel raggruppamento, larga quanto la giustezza del testo stesso; nel secondo caso, si impone la larghezza indicata.

> È importante osservare che '**`\hbox`**' non è propriamente una macro e il suo argomento è precisamente un gruppo, per cui l'espressione contenuta è isolata dal testo successivo. Lo stesso dicasi per le parole di controllo che definiscono delle scatole verticale, descritte più avanti.

Esistono delle parole di controllo per la gestione delle scatole orizzontali vuote a vario titolo:

`\llap {`***espressione***`}`

`\rlap {`***espressione***`}`

`\null`

`\strut`

`\mathstrut`

Le parole di controllo '**`\llap`**' e '**`\rlap`**' definiscono un gruppo vuoto, in cui l'espressione del gruppo che viene indicato risulta sporgere rispettivamente a sinistra e a destra. I nomi usati per queste parole di controllo ricordano mnemonicamente i termini *left overlap* e *right overlap*. Tra le altre possibilità, queste istruzioni si prestano per consentire la sovrapposizione del testo; per esempio, si può sbarrare un lettera «o» con una barra obliqua inversa semplicemente così: '**`o\llap{$\backslash$}`**'.

La parola di controllo '**`\null`**' inserisce una scatola vuota equivalente all'istruzione '**`\hbox{}`**'; invece, '**`\strut`**' crea una scatola vuota alta e profonda quanto una parentesi tonda; nello stesso modo funziona la parola di controllo '**`\mathstrut`**' che invece si usa in modalità matematica.

Sempre in modalità matematica, si può usare '**`\underline`**' che sottolinea il testo contenuto nell'espressione:

`\underline {`***espressione***`}`

In questo caso, per fare in modo che la linea di sottolineatura sia sempre alla stessa altezza, si inserisce generalmente una scatola vuota generata da '**`\mathstrut`**'. L'esempio seguente riepiloga l'uso delle scatole orizzontali:

```
Paragrafo normale bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla \par
inizio paragrafo; \hbox {testo in scatola}; fine paragrafo.
\par
inizio paragrafo; \hbox to 5cm {testo in scatola}; fine
paragrafo. \par
inizio paragrafo; \hbox to 5cm {testo \hss in scatola}; fine
paragrafo. \par
\line{Riga di testo normale allineata a sinistra e a
destra.}
\line{\llap {a)} Bla bla bla \rlap {(b}}
$\underline {testo sottolineato}, g j p q y,
\underline {\mathstrut testo sottolineato con una linea
pi\grave u bassa}$ \par
```

Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
inizio paragrafo; testo in scatola; fine paragrafo.
inizio paragrafo; testo in scatola; fine paragrafo.
inizio paragrafo; testo in scatola; fine paragrafo.
Riga di testo normale allineata a sinistra e a destra.
a) Bla bla bla (b
$\underline{testosottolineato}, gjpqy, \underline{testosottolineatoconunalineapi\grave{u}bassa}$

L'esempio dell'utilizzo più importante delle scatole orizzontali è dato dalla macro '**`\line`**' che è definita semplicemente così:

```
\def\line{\hbox to\hsize}
```

In pratica, quando si scrive '**`\line{`*testo*`}`**', non si sta indicando un argomento alla macroistruzione, ma si ottiene solo la sostituzione di '**`\line`**' con '**`\hbox to\hsize`**', per cui alla fine si ottiene '**`\hbox to\hsize {`*testo*`}`**' complessivamente. In altri termini, la macroistruzione '**`\line`**' è seguita da un gruppo che, come tale, isola il testo che racchiude.

Per la realizzazione di scatole verticali sono disponibili tre parole di controllo: '**`\vbox`**', '**`\vtop`**' e '**`\vcenter`**', dove l'ultima interviene solo in ambienti matematici.

`\vbox {`***espressione***`}` | `\vbox to` ***altezza*** `{`***espressione***`}`

`\vtop {`***espressione***`}` | `\vtop to` ***altezza*** `{`***espressione***`}`

`\vcenter {`***espressione***`}` | `\vcenter to` ***altezza*** `{`***espressione***`}`

I due modelli sintattici alternativi, '**`\`*nome gruppo***' e '**`\`*nome* `to` *altezza gruppo***', consentono di creare scatole alte quanto serve, in base al contenuto, oppure alte esattamente quanto richiesto.

La scatola che si ottiene con '**`\vbox`**', colloca la posizione inferiore al livello della riga da cui si parte; estendendosi verso l'alto; la scatola che si ottiene con '**`\vtop`**' si estende invece verso il basso; infine, la scatola generata con '**`\vcenter`**' risulta centrata verticalmente rispetto alla posizione di partenza.

La larghezza di queste scatole è controllata dal valore restituito da '**\hsize**'; valore che può essere cambiato opportunamente al loro interno.

```
Paragrafo normale bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla \par
A: \vbox {\hsize=10cm Scatola verticale, larga 10 cm, che si
espande verso l'altro. bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla} \par
B: \vtop {\hsize=10cm Scatola verticale, larga 10 cm, che si
espande verso il basso. bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla} \par
C: $\vcenter {\hsize=10cm Scatola verticale, larga 10 cm,
che si espande verso l'altro e verso il basso. bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla}$ \par
D: \vbox to 3cm {\hsize=7cm Questa \'e una scatola
rettangolare, larga sette centimetri e alta tre centimetri.
bla bla bla bla bla bla bla bla bla bla bla bla bla bla}
\par
Paragrafo normale bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla \par
```

Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

A: Scatola verticale, larga 10 cm, che si espande verso l'altro. bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

B: Scatola verticale, larga 10 cm, che si espande verso il basso. bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

C: Scatola verticale, larga 10 cm, che si espande verso l'altro e verso il basso. bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

D: Questa è una scatola rettangolare, larga sette centimetri e alta tre centimetri. bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Paragrafo normale bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

L'esempio mostra quattro casi di scatole inserite in paragrafi di testo normale, dove ogni paragrafo inizia con una lettera alfabetica per dare un riferimento visivo alla collocazione della stessa. Si può osservare l'ultimo caso in cui si fissa anche la larghezza della scatola intervenendo nel valore a cui si accede con la parola di controllo '**\hsize**'.

Benché la parola di controllo '**\vcenter**' debba essere inserita in un ambiente matematico, il gruppo che controlla si trova in un ambiente normale.

49.14.1 Spostamento orizzontale delle scatole

Le due macro '**\moveleft**' e '**\moveright**', consentono di spostare orizzontalmente una scatola verticale od orizzontale che sia:

\moveleft *lunghezza* \hbox{*contenuto*} | \vbox{*contenuto*}
\moveright *lunghezza* \hbox{*contenuto*} | \vbox{*contenuto*}

Come si vede dal modello sintattico, si ottiene lo spostamento, rispettivamente, verso sinistra o verso destra, della lunghezza indicata. Si osservi l'esempio seguente:

```
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla.\par
\moveright 3 cm \vbox{
\hsize=7cm
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla.\par
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla.\par
}\par
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla.\par
```


In pratica, dopo un paragrafo normale, viene richiesto un rientro di 3 cm per una scatola verticale contenente del testo, nella quale la giustezza è di soli 7 cm. Infine, un altro paragrafo normale permette di verificare che al di fuori della scatola, la giustezza torna quella di prima.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Eventualmente, un piccolo spazio verticale può evitare che il testo che precede la scatola verticale si trovi a toccare il contenuto della stessa.

49.14.2 Scatole più complesse

Le scatole definite con '**\hbox**', '**\vbox**' e altre macroistruzioni simili, possono essere inserite una dentro l'altra per formare degli effetti speciali, soprattutto se si abbina questo fatto all'uso delle linee orizzontali e verticali. Di solito ci si affida a macroistruzioni già pronte, oppure a esempi documentati; comunque è utile cercare di comprendere cosa sta sotto. Viene proposto un esempio, attraverso il quale si crea una macroistruzione che racchiude del testo dentro una scatola orizzontale bordata:

```
\def\scatola#1{%
    \leavevmode                                  % 5
    \hbox{%                                      % 4
        \vrule
        \vtop{%                                  % 3
            \vbox{%                              % 2
                \hrule\kern 1pt
                \hbox{%
                    \kern 1pt \strut#1\kern 1pt  % 1
                }%
            }%
            \kern 1pt
            \hrule
        }%
        \vrule
    }
}
```

Questo esempio così articolato è tratto da *TeX made easy* di Daniel M. Zirin. Per riuscire a comprenderlo, occorre partire dal centro.

1. L'istruzione '**\kern 1pt \strut#1\kern 1pt**' inserisce il parametro della macro aggiungendo un punto di spazio a sinistra e a destra, garantendo di avere a disposizione lo spazio verticale necessario a scrivere qualunque lettera (che si espanda in alto e in basso, rispetto alla base).

2. Tutto questo è racchiuso in una scatola verticale che, prima disegna una linea orizzontale, quindi lascia un punto di spazio in verticale, infine inserisce l'istruzione precedente.

3. Il tutto è contenuto in un'altra scatola verticale che si espande verso il basso, dove alla fine si aggiunge un altro spazio verticale di un punto e un'altra linea orizzontale.

4. Ancora una volta, tutto è contenuto dentro un'altra scatola, questa volta orizzontale, che prima e dopo aggiunge due linee verticali.

5. Per sicurezza, la macroistruzione inizia chiamando '**\leavevmode**' per garantire il passaggio alla modalità orizzontale.

Ecco come si comporta la macroistruzione:

```
\scatola{Attenzione!} con le scatole ci si confonde
facilmente.
```

Attenzione! con le scatole ci si confonde facilmente.

Si osservi che nella definizione della macroistruzione '**\scatola**' sono stati inseriti molti commenti per consentire la suddivisione del codice su più righe, senza interferire con la composizione finale.

49.15 Tabelle

La gestione delle tabelle con TeX è piuttosto complessa. Nelle sezioni successive vengono mostrati soltanto i due tipi fondamentali di tabella di TeX; Si tenga presente che esistono altri modi di crearne, per esempio inserendo scatole dentro altre scatole, ma anche l'ambiente matematico dispone di un tipo di tabella che facilita la realizzazione di matrici.

49.15.1 Tabulazione

Le macchine da scrivere tradizionali disponevano di un sistema di stop di tabulazione, programmato di volta in volta, con il quale si potevano allineare i dati in colonne, formando così delle tabelle. Il sistema più semplice per creare tabelle con TeX è proprio quello che si rifà all'uso della macchina da scrivere, attraverso la definizione di stop di tabulazione che poi vengono richiamati quando desiderato. Si osservi l'esempio seguente:

```
\settabs 3 \columns

Testo normale, bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla

\+ nero & 0 & zero \cr
\+ marrone & 1 & uno \cr
\+ rosso & 2 & due \cr
\+ arancio & 3 & tre \cr
\+ giallo & 4 & quattro \cr
\+ verde & 5 & cinque \cr
\+ blu & 6 & sei \cr
\+ viola & 7 & sette \cr
\+ grigio & 8 & otto \cr
\+ bianco & 9 & nove \cr

Testo normale, bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla

\+ argento & 10 \% \cr
\+ oro & 5 \% \cr
\+ & 2 \% \cr
\+ & 1 \% \cr
```

L'istruzione '**\settabs 3 \columns**' definisce, da quel punto in poi, l'inserimento di due stop di tabulazione, dividendo la giustezza del testo in tre parti uguali. Dopo tale definizione, si può scrivere normalmente, come se nulla fosse cambiato, oppure si può fare riferimento esplicito alla tabulazione, iniziando e concludendo le righe con '**\+**' e con '**\cr**'. Nello spazio tra queste due istruzioni, la e-commerciale ('**&**') ha il significato di carattere di tabulazione, con il quale si raggiunge lo stop successivo. Si può vedere il risultato della composizione dell'esempio nella figura successiva.

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre
giallo	4	quattro
verde	5	cinque
blu	6	sei
viola	7	sette
grigio	8	otto
bianco	9	nove

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

argento	10 %
oro	5 %
	2 %
	1 %

Come si dovrebbe notare leggendo l'esempio, non è necessario usare tutte le colonne previste, così come è possibile saltare le colonne che non interessano. Quello che non si intuisce, invece, è la possibilità di fare riferimento a stop di tabulazione oltre alla quantità dichiarata inizialmente: in tal caso si raggiungono colonne successive che, ovviamente, vanno oltre la giustezza del testo.

L'istruzione del tipo '**\settabs *n* \columns**' ha effetto a partire dal punto in cui viene utilizzata. Se si vuole cambiare tabulazione, basta inserire una nuova istruzione di questo tipo. Inoltre, è possibile usare anche i raggruppamenti per cambiare temporaneamente la tabulazione di una tabella; tutto funziona in modo molto intuitivo.

```
\settabs 3 \columns
\+ nero & 0 & zero \cr
\+ marrone & 1 & uno \cr
\+ rosso & 2 & due \cr
{\settabs 4 \columns
\+ arancio & 3 & tre \cr
\+ giallo & 4 & quattro \cr
\+ verde & 5 & cinque \cr
\+ blu & 6 & sei \cr
}
\+ viola & 7 & sette \cr
\+ grigio & 8 & otto \cr
\+ bianco & 9 & nove \cr
```

In questa variante semplificata dell'esempio già presentato, le righe dall'arancio al blu hanno una tabulazione più stretta, perché si divide la giustezza in quattro porzioni.

nero	0		zero
marrone	1		uno
rosso	2		due
arancio	3	tre	
giallo	4	quattro	
verde	5	cinque	
blu	6	sei	
viola	7		sette
grigio	8		otto
bianco	9		nove

Gli stop di tabulazione possono essere collocati in modo diverso, indicando una riga di esempio al posto della definizione pura e semplice della quantità di colonne che si vogliono ottenere. In pratica, si hanno a disposizione due istruzioni alternative:

```
\settabs n \columns
```

```
\settabs \+ colonna_1 [& colonna_n]… \cr
```

Viene riproposto l'esempio già visto, in forma ancora più breve, in cui viene inserito lo stesso modello usato per definire gli stop di tabulazione alla fine della tabella, per poter vedere effettivamente in che modo vengono misurati gli spazi:

```
\settabs \+ MMMMMMMMMMM & MMMMM & MMMMMMM \cr
\+ nero & 0 & zero \cr
\+ marrone & 1 & uno \cr
\+ rosso & 2 & due \cr
\+ arancio & 3 & tre \cr
Il modello usato per gli stop \‘e il seguente:
```

```
\+ MMMMMMMMMMMM & MMMMM & MMMMMMM \cr
```

nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Il modello usato per gli stop è il seguente:
MMMMMMMMMMMM MMMMM MMMMMMM

Naturalmente, il modello delle colonne può essere anche più simile al testo usato realmente nella tabella. In questo caso, con la lettera «M» maiuscola, si fa riferimento a un quadratone in modo semplice e chiaro.

Il testo che si inserisce all'interno di uno spazio che rappresenta una cella, esclusa l'ultima di ogni riga, è racchiuso implicitamente in un gruppo che si comporta come la macro '**`\leftline`**'. Questo fatto consente quindi di gestire gli allineamenti nell'ambito delle celle; inoltre, garantisce che il testo di ognuna di queste celle sia isolato in un gruppo autonomo.

```
\settabs
\+ MMMMMMMMMMMM & MMMMM        & MMMMMMM      & \cr
\+ \hrulefill   & \dotfill     & \hrulefill   & \cr
\+ \hfil Colore & \hfil Cifra  & \hfil Cifra  & \cr
\+ \hrulefill   & \dotfill     & \hrulefill   & \cr
\+ nero         & \hfill 0     & \hfil zero   & \cr
\+ marrone      & \hfill 1     & \hfil uno    & \cr
\+ \bf rosso    & \hfill 2     & \hfil due    & \cr
\+ arancio      & \hfill 3     & \hfil tre    & \cr
\+ \hrulefill   & \dotfill     & \hrulefill   & \cr
```

L'esempio mostra una tabella, in cui il sorgente è organizzato in modo da facilitarne la lettura. In particolare, il modello per gli stop di tabulazione è stato messo nella riga successiva a quella in cui appare '**`\settabs`**'; inoltre, c'è una cella finale in più che non viene usata. Infatti, dalle celle in cui il testo si comporta come se fosse usata la macro '**`\leftline`**' è esclusa l'ultima che precede l'istruzione '**`\cr`**'.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Bisogna ricordare che la macro '**`\leftline`**' include uno spazio leggermente elastico alla fine, corrispondente a '**`\hfil`**'. In tal modo, per centrare il testo in una cella, basta farlo precedere da un altro '**`\hfil`**', se invece si vuole ottenere l'allineamento a destra, si deve usare inizialmente uno spazio più elastico come '**`\hfill`**'.

Nell'esempio mostrato si riesce ad arrivare a una tabella abbastanza «dignitosa», con tanto di intestazione, delimitata da linee orizzontali. Naturalmente, si potrebbero usare linee dello stesso tipo, ma in questo caso si vuole mostrare bene lo spazio di competenza di ogni colonna. Al posto di '**`\hrulefill`**' o di '**`\dotfill`**' si può anche usare '**`\hrule`**', al di fuori delle celle, nel modo seguente:

```
\settabs
\+ MMMMMMMMMMMM & MMMMM        & MMMMMMM      & \cr
\hrule
\+ \hfil Colore & \hfil Cifra  & \hfil Cifra  & \cr
\hrule
\+ nero         & \hfill 0     & \hfil zero   & \cr
\+ marrone      & \hfill 1     & \hfil uno    & \cr
\+ \bf rosso    & \hfill 2     & \hfil due    & \cr
\+ arancio      & \hfill 3     & \hfil tre    & \cr
\hrule
```

Tuttavia, in tal caso si vede che le linee orizzontali che si ottengono toccano il testo delle righe della tabella:

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Per risolvere il problema, può essere conveniente l'uso della macro '**`\strut`**', con cui si dà alle righe l'altezza e la profondità corrispondente a quella di una parentesi tonda, senza che questa appaia effettivamente nel testo:

```
\settabs
\+ MMMMMMMMMMMM         & MMMMM        & MMMMMMM      & \cr
\hrule
\+ \strut \hfil Colore  & \hfil Cifra  & \hfil Cifra  & \cr
\hrule
\+ \strut nero          & \hfill 0     & \hfil zero   & \cr
\+ \strut marrone       & \hfill 1     & \hfil uno    & \cr
\+ \strut \bf rosso     & \hfill 2     & \hfil due    & \cr
\+ \strut arancio       & \hfill 3     & \hfil tre    & \cr
\hrule
```

In tal modo si ottiene anche il risultato di avere tutte le righe della tabella della stessa altezza, quella massima, anche se il testo che contengono non lo richiederebbe.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Se si inserisce una tabella del genere in una scatola verticale, questa scatola può essere spostata facilmente in modo orizzontale, con l'istruzione '**`\moveright`**', come nell'esempio seguente, in cui la si sposta di 3 cm:

```
Paragrafo di esempio, bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

\moveright 3 cm \vbox{
\settabs
\+ MMMMMMMMMMMM         & MMMMM        & MMMMMMM      & \cr
\hrule
\+ \strut \hfil Colore  & \hfil Cifra  & \hfil Cifra  & \cr
\hrule
\+ \strut nero          & \hfill 0     & \hfil zero   & \cr
\+ \strut marrone       & \hfill 1     & \hfil uno    & \cr
\+ \strut \bf rosso     & \hfill 2     & \hfil due    & \cr
\+ \strut arancio       & \hfill 3     & \hfil tre    & \cr
\hrule
}
```

Tuttavia, come si vede sotto, in questo modo il testo che precede la tabella si trova a toccare la tabella stessa:

Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

Per rimediare basta aggiungere un piccolo spazio verticale, per esempio con l'istruzione '**`\vskip`**', prima della tabella.

Volendo arrivare a centrare la tabella, si può inserire la scatola verticale nella macro '**`\line`**', come nell'esempio seguente:

```
Paragrafo di esempio, bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla.\par
\vskip 1 em
\line{\hfil\vbox{
\settabs
\+ MMMMMMMMMMMM         & MMMMM        & MMMMMMM      & \cr
\hrule
\+ \strut \hfil Colore  & \hfil Cifra  & \hfil Cifra  & \cr
\hrule
\+ \strut nero          & \hfill 0     & \hfil zero   & \cr
\+ \strut marrone       & \hfill 1     & \hfil uno    & \cr
\+ \strut \bf rosso     & \hfill 2     & \hfil due    & \cr
\+ \strut arancio       & \hfill 3     & \hfil tre    & \cr
\hrule
}\hfil}
```

Si osservi l'uso dell'istruzione '**`\vskip`**' per ottenere una piccola

distanza tra il testo e la tabella, come già descritto in precedenza:

Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

49.15.2 Tabelle più complesse

Attraverso l'istruzione '**`\halign`**' è possibile definire delle tabelle un po' più complesse rispetto al sistema di stop di tabulazione già presentato nella sezione precedente. L'istruzione in questione viene utilizzata sinteticamente nel modo seguente:

```
\halign{modello_delle_righe
riga
...
}
```

In particolare, il modello delle righe utilizza il simbolo '**`#`**' per fare riferimento al contenuto delle celle, mentre le righe hanno la forma consueta, con la differenza che manca il simbolo di controllo '**`\+`**' iniziale. L'esempio seguente è molto simile all'ultima tabella realizzata con l'uso degli stop di tabulazione, con tanto di centratura orizzontale:

```
Paragrafo di esempio, bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla.\par
\vskip 1 em
\line{\hfil\vbox{
\halign{
\strut #        & \hfil # & \hfil # \hfil \cr
\noalign{\hrule}
Colore          & Cifra & Cifra \cr
\noalign{\hrule}
nero            & 0     & zero  \cr
marrone         & 1     & uno   \cr
\bf rosso       & 2     & due   \cr
arancio         & 3     & tre   \cr
\noalign{\hrule}
}
}\hfil}
```

La prima cosa che si deve osservare è il fatto che '**`\strut`**' viene usato solo nella dichiarazione del modello delle righe; ciò è sufficiente perché sia inserito automaticamente all'inizio di tutte le righe della tabella. Si può osservare anche che non è necessario definire la larghezza delle colonne nel modello, perché questa viene determinata automaticamente in base al contenuto delle righe stesse. Infine, l'ultima colonna della tabella non si comporta diversamente dalle altre, pertanto non è più necessario il trucco della colonna finale vuota.

Paragrafo di esempio, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

Colore	Cifra	Cifra
nero	0	zero
marrone	1	uno
rosso	2	due
arancio	3	tre

A differenza delle tabelle realizzate per mezzo della tabulazione, quando si vuole inserire una linea orizzontale con l'istruzione '**`\hrule`**', occorre inserirla nell'istruzione '**`\noalign`**'. In pratica, con '**`\noalign`**' si può inserire qualcosa che riguarda tutta la riga, senza suddivisione in colonne; tuttavia, è meglio evitare di inserire testo normale, perché in tal caso verrebbe perso l'effetto dell'allineamento orizzontale introdotto con '**`\line`**' e '**`\vbox`**'.

È importante comprendere che ciò che si inserisce nel modello iniziale delle righe, viene inserito nello stesso modo all'interno delle righe. Si osservi la variante seguente:


```
\line{\hfil\vbox{
\halign{
\strut \quad \bf # & \hfil #          & \hfil (#) \hfil \cr
\noalign{\hrule}
Colore             & \bf Cifra \hfil  & \bf Cifra     \cr
\noalign{\hrule}
nero               & 0                & zero          \cr
marrone            & 1                & uno           \cr
rosso              & 2                & due           \cr
arancio            & 3                & tre           \cr
\noalign{\hrule}
}
}\hfil}
```

Per prima cosa, è stato aggiunto un piccolo spazio orizzontale, di un quadratone, nella prima cella, attraverso la macro '**`\quad`**'; in tal modo si evita che la prima colonna inizi esattamente sul bordo sinistro, per motivi estetici. Inoltre, si fa in modo che tutta la prima colonna appaia in neretto, senza dover intervenire in ogni cella. Nel modello dell'ultima colonna, si racchiude il riferimento alle celle tra parentesi tonde, solo a scopo dimostrativo (infatti il risultato che si ottiene non è perfetto esteticamente). Dal momento che si vuole la riga di intestazione tutta in neretto, si interviene singolarmente nella seconda e nella terza cella di questa riga; in particolare, la parola «Cifra» della seconda colonna, viene centrata, aggiungendo un '**`\hfil`**' finale, che contrasta e bilancia quanto dichiarato nel modello corrispondente.

Colore	**Cifra**	(**Cifra**)
nero	0	(zero)
marrone	1	(uno)
rosso	2	(due)
arancio	3	(tre)

Dal momento che tutto ciò che si inserisce nella riga del modello viene messo anche nelle righe della tabella, per ottenere delle linee verticali si possono usare delle istruzioni '**`\vrule`**':

```
\line{\hfil\vbox{
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                   & \bf Cifra \hfil & \bf Cifra                    \cr
\noalign{\hrule}
nero                     & 0               & zero                         \cr
marrone                  & 1               & uno                          \cr
rosso                    & 2               & due                          \cr
arancio                  & 3               & tre                          \cr
\noalign{\hrule}
}
}\hfil}
```

In questo caso il risultato estetico non è ancora perfetto, ma ormai dovrebbe essere chiaro come si può intervenire nel modello delle righe.

Colore	**Cifra**	(**Cifra**)
nero	0	(zero)
marrone	1	(uno)
rosso	2	(due)
arancio	3	(tre)

Esiste la possibilità di unire assieme più celle nella stessa riga, attraverso l'uso della macroistruzione '**`\multispan{n}`**', dove ***n*** rappresenta la quantità di colonne da unire. Tuttavia, in tal caso non viene preso in considerazione il formato stabilito nella riga di intestazione. Si osservi l'esempio seguente:

```
\line{\hfil\vbox{
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                   & \bf Cifra \hfil & \bf Cifra                    \cr
\noalign{\hrule}
nero                     & 0               & zero                         \cr
marrone                  & 1               & uno                          \cr
\multispan{2} rosso                        & due                          \cr
arancio                  & 3               & tre                          \cr
\noalign{\hrule}
}
}\hfil}
```

Come si vede sotto, il contenuto della cella allargata appare centrato nello spazio che si ritrova ad avere a disposizione, ma ciò è ottenuto con uno spazio leggermente elastico, contrastabile facilmente.

Colore	Cifra	(Cifra)
nero	0	(zero)
marrone	1	(uno)
rosso		(due)
arancio	3	(tre)

Per sistemare l'allineamento, ripristinare il neretto e la linea verticale alla sinistra, basta intervenire nella cella; in particolare è sufficiente uno spazio con una piccola elasticità per contrastare l'allineamento normale al centro:

```
\line{\hfil\vbox{
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                   & \bf Cifra \hfil & \bf Cifra                    \cr
\noalign{\hrule}
nero                     & 0               & zero                         \cr
marrone                  & 1               & uno                          \cr
\multispan{2}\vrule\quad\bf rosso scuro \hfil    & due                     \cr
arancio                  & 3               & tre                          \cr
\noalign{\hrule}
}
}\hfil}
```

Colore	Cifra	(Cifra)
nero	0	(zero)
marrone	1	(uno)
rosso scuro		(due)
arancio	3	(tre)

Per evitare che in una cella venga inserito tutto ciò che appare nel modello delle righe, basta usare l'istruzione '**\omit**'. Si osservi l'esempio, in cui si risolve il problema delle parentesi tonde attorno al titolo dell'ultima colonna:

```
\line{\hfil\vbox{
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                &\bf Cifra\hfil &\omit\vrule\hfil\bf Cifra\hfil\vrule\cr
\noalign{\hrule}
nero                     & 0               & zero                         \cr
marrone                  & 1               & uno                          \cr
rosso                    & 2               & due                          \cr
arancio                  & 3               & tre                          \cr
\noalign{\hrule}
}
}\hfil}
```

Colore	Cifra	Cifra
nero	0	(zero)
marrone	1	(uno)
rosso	2	(due)
arancio	3	(tre)

Prima di concludere l'argomento, occorre fare presente la possibilità che TeX allunghi il testo verticalmente in fase di composizione. Se ciò accade nell'ambito di una tabella nella quale si utilizza '**\vrule**' per ottenere delle linee verticali, si rischia di vedere queste linee spezzettate. Per evitare che nell'ambito della tabella TeX possa fare degli allungamenti, si può usare la macro '**\offinterlineskip**' dentro la scatola verticale:

```
\line{\hfil\vbox{\offinterlineskip
\halign{
\strut\vrule \quad \bf # & \vrule \hfil #  & \vrule \hfil (#) \hfil \vrule \cr
\noalign{\hrule}
Colore                &\bf Cifra\hfil &\omit\vrule\hfil\bf Cifra\hfil\vrule\cr
\noalign{\hrule}
nero                     & 0               & zero                         \cr
marrone                  & 1               & uno                          \cr
rosso                    & 2               & due                          \cr
arancio                  & 3               & tre                          \cr
\noalign{\hrule}
}
}\hfil}
```

49.16 Ambienti matematici

Come già accennato, TeX distingue tra due modalità di funzionamento: un contesto normale e un contesto matematico. L'ambiente matematico si introduce e si conclude con il simbolo '**$**' e in tale situazione diventano disponibili delle istruzioni che non si possono utilizzare al di fuori di questo ambito, mentre alcune istruzioni dell'ambiente normale non lo sono più.

Anche se nelle sezioni successive si fa riferimento soltanto all'ambiente matematico, nelle tabelle riassuntive mostrate, le sequenze di controllo che possono essere usate solo nell'ambiente matematico appaiono delimitate sempre tra una coppia di '**$**', in modo da evidenziare il contesto del loro utilizzo ed evitare confusione.

49.16.1 Due situazioni differenti

Esistono due modi di mostrare un'informazione matematica: all'interno di testo normale, oppure in un blocco a parte. Quando si inserisce l'informazione dentro del testo normale, si delimita l'ambiente matematico tra due simboli '**$**', per esempio nel modo seguente:

```
Se $R$ \'e il ritardo di ogni singola consegna, $T$
\'e il tempo a disposizione per la consegna, $I$ \'e il
valore della merce, il ritardo medio si esprime come
${\sum {R \over T} \cdot I}\over{\sum I}$.
```

In tal caso, come si vede dal risultato della composizione, il testo ne risente, in particolare per lo spazio tra le righe che può essere aumentato.

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime come $\frac{\sum \frac{R}{T} \cdot I}{\sum I}$.

In alternativa, le formule possono essere messe in un blocco separato, come nell'esempio seguente:

```
Se $R$ \'e il ritardo di ogni singola consegna, $T$ \'e il
tempo a disposizione per la consegna, $I$ \'e il valore
della merce, il ritardo medio si esprime secondo la formula
seguente:\par
$${\sum {R \over T} \cdot I}\over{\sum I}$$
```

L'effetto è quello di ottenerle al centro della giustezza, con una spaziatura verticale adeguata rispetto al testo che precede e che segue.

Se R è il ritardo di ogni singola consegna, T è il tempo a disposizione per la consegna, I è il valore della merce, il ritardo medio si esprime secondo la formula seguente:

$$\frac{\sum \frac{R}{T} \cdot I}{\sum I}$$

Negli esempi è stato mostrato l'uso dell'ambiente matematico anche per delimitare i nomi delle variabili. Ciò permette di mantenere coerenza con la forma mostrata nelle formule.

Ciò che viene delimitato in una coppia di '**$**' o di '**$$**', non può essere suddiviso in «righe» differenti (tranne il caso di forme tabellari particolari, come le matrici o i sistemi di equazioni), al contrario di quanto avviene per la composizione del testo normale; inoltre, il sorgente non può contenere righe vuote o bianche. In pratica si tratta di un blocco orizzontale compatto e indivisibile.

49.17 Spazi orizzontali

Si osservi subito l'esempio seguente, in cui si scrive la parola «affittare» all'interno di un ambiente matematico, prima in modo normale, quindi spaziando le lettere:

```
$affittare$\par
$a    f    f    i    t  t  a  r  e$\par
```

Come si può osservare dalla composizione che si ottiene, il risultato è esattamente lo stesso. In pratica non viene più preso in considerazione il legato tipografico e gli spazi sono ignorati completamente:

$$affittare$$
$$a\ f\ f\ i\ t\ t\ a\ r\ e$$

Per inserire degli spazi in un ambiente matematico, l'unico modo è quello di usare sequenze di controllo specifiche, che possano essere accettate anche in questo ambiente. La tabella 49.245 riepiloga le sequenze di controllo che possono essere usate in un ambiente matematico per ottenere degli spazi orizzontali. Alcune di queste sequenze sono valide solo in un ambito matematico, pertanto appaiono circoscritte da una coppia di '**$**', in modo da sottolineare questa loro caratteristica.

Tabella 49.245. Sequenze di controllo per ottenere delle spaziature orizzontali in ambito matematico.

Sequenza	Significato
`\qquad`	Due quadratoni.
`\quad`	Un quadratone.
`\<SP>`	Uno spazio «normale».
`$\;$`	Uno spazio spesso (5/8 di quadratone).
`$\>$`	Uno spazio medio (2/9 di quadratone).
`$\.$`	Uno spazio molto sottile (1/6 di quadratone).
`$\!$`	Uno spazio negativo (-1/6 di quadratone).

Per inserire del testo descrittivo all'interno di un ambiente matematico, lo si può delimitare in una scatola orizzontale, con l'istruzione '**`\hbox`**'. Si osservi l'esempio:

```
$prima durante e dopo$\par
$prima \hbox{ durante } e dopo$\par
```

In questo caso, si vuole isolare e spaziare la parola «durante» rispetto al resto, mostrandola con caratteri normali.

$$primaduranteedopo$$
$$prima\ \text{durante}\ edopo$$

49.17.1 Caratteri e simboli

Nell'ambiente matematico, il testo normale viene composto utilizzando un insieme di caratteri differente rispetto a quello normale, in cui anche la forma è particolare, essendo un tipo speciale di corsivo (si ottiene eventualmente con la macro '**`\mit`**'). Inoltre, sono disponibili dei simboli aggiuntivi, in particolare le lettere greche e altri simboli utili in matematica.

Tra le lettere greche, alcune si ottengono come caratteri dell'insieme normale. Nelle tabelle, quando si vuole sottolineare il fatto che si tratta di lettere scritte utilizzando il carattere tondo normale, si mostra un'istruzione del tipo '**`\hbox{\rm x}`**', che rappresenta il codice necessario all'inserimento in un ambiente matematico.

Tabella 49.248. Accenti nell'ambiente matematico.

Simbolo	Codice	Annotazioni
$\acute{o}$	$\acute o$	Accento acuto matematico.
$\grave{o}$	$\grave o$	Accento grave matematico.
$\hat{o}$	$\hat o$	Accento circonflesso matematico.
$\ddot{o}$	$\ddot o$	
$\tilde{o}$	$\tilde o$	
$\bar{o}$	$\bar o$	
$\dot{o}$	$\dot o$	
$\breve{o}$	$\breve o$	
$\check{o}$	$\check o$	
$\vec{o}$	$\vec o$	
$\imath$	$\imath$	Da usare per aggiungere un accento.
$\jmath$	$\jmath$	Da usare per aggiungere un accento.
$\widehat{abc}$	$\widehat{abc}$	
$\widetilde{abc}$	$\widetilde{abc}$	

Tabella 49.249. Lettere greche.

Simbolo	Codice	Simbolo	Codice	Simbolo	Codice	Nome
α	α			A	\hbox{\rm A}	Alfa
β	β			B	\hbox{\rm B}	Beta
γ	γ			Γ	Γ	Gamma
δ	δ			Δ	Δ	Delta
ϵ	ϵ	ε	ε	E	\hbox{\rm E}	Epsilon
ζ	ζ			Z	\hbox{\rm Z}	Zeta
η	η			H	\hbox{\rm H}	Eta
θ	θ	ϑ	ϑ	Θ	Θ	Theta
ι	ι			I	\hbox{\rm I}	Iota
κ	κ			K	\hbox{\rm K}	Kappa
λ	λ			Λ	Λ	Lambda
μ	μ			M	\hbox{\rm M}	Mu
ν	ν			N	\hbox{\rm N}	Nu
ξ	ξ			Ξ	Ξ	Xi
o	\hbox{\rm o}	o	o	O	\hbox{\rm O}	Omicron
π	π	ϖ	ϖ	Π	Π	Pi
ρ	ρ	ϱ	ϱ	P	\hbox{\rm P}	Rho
σ	σ	ς	ς	Σ	Σ	Sigma
τ	τ			T	\hbox{\rm T}	Tau
υ	υ			Υ	Υ	Upsilon
ϕ	ϕ	φ	φ	Φ	Φ	Phi
χ	χ			X	\hbox{\rm X}	Chi
ψ	ψ			Ψ	Ψ	Psi
ω	ω			Ω	Ω	Omega

Tabella 49.250. Simboli matematici comuni.

Simbolo	Codice	Annotazioni
$\backslash$	$\backslash$	Barra obliqua inversa.
$\aleph$	$\aleph$	
$\hbar$	$\hbar$	
ℓ	ℓ	
$\wp$	$\wp$	
$\Re$	$\Re$	
$\Im$	$\Im$	
∂	∂	Derivata parziale.
∞	∞	Infinito.
$\prime$	$\prime$	Primo.
$'$	$'
$\emptyset$	$\emptyset$	
∇	∇	
$\top$	$\top$	
$\bot$	$\bot$	
$\vert$	$\vert$	
$\vert$	$\vert$	Equivalente a $|$.
$\Vert$	$\Vert$	
$\Vert$	$\Vert$	Equivalente a $\|$.
$\angle$	$\angle$	
$\triangle$	$\triangle$	
$\forall$	$\forall$	
$\exists$	$\exists$	
$\neg$	$\neg$	
$\flat$	$\flat$	
$\natural$	$\natural$	
$\sharp$	$\sharp$	
$\surd$	$\surd$	Radice senza la linea superiore.
$\clubsuit$	$\clubsuit$	Fiori.
$\diamondsuit$	$\diamondsuit$	Quadri.
$\heartsuit$	$\heartsuit$	Cuori.
$\spadesuit$	$\spadesuit$	Picche.

Tabella 49.251. Operatori binari comuni.

Simbolo	Codice	Annotazioni
$\pm$	$\pm$	Più o meno.
$\mp$	$\mp$	Meno o più.
$\cdot$	$\cdot$	
$\setminus$	$\setminus$	
$+$	$+$	Somma.
$-$	$-$	Sottrazione.
$\times$	$\times$	Moltiplicazione.
$\div$	$\div$	Divisione.
	$*$	Moltiplicazione discrezionale.
$\ast$	$\ast$	Asterisco.
$\star$	$\star$	Stella.
$\diamond$	$\diamond$	
$\circ$	$\circ$	
$\bullet$	$\bullet$	
$\cap$	$\cap$	Intersezione.
$\cup$	$\cup$	Unione.
$\uplus$	$\uplus$	
$\sqcap$	$\sqcap$	
$\sqcup$	$\sqcup$	
$\triangleleft$	$\triangleleft$	
$\triangleright$	$\triangleright$	
$\wr$	$\wr$	
$\bigcirc$	$\bigcirc$	
$\bigtriangleup$	$\bigtriangleup$	
$\bigtriangledown$	$\bigtriangledown$	
$\vee$	$\vee$	
$\vee$	$\lor$	OR logico (uguale a $\vee$).
$\wedge$	$\wedge$	
$\wedge$	$\land$	AND logico (uguale a $\wedge$).
$\oplus$	$\oplus$	
$\ominus$	$\ominus$	
$\otimes$	$\otimes$	
$\oslash$	$\oslash$	
$\odot$	$\odot$	
$\dagger$	$\dagger$	
$\ddagger$	$\ddagger$	
II	$\amalg$	

Tabella 49.252. Operatori di relazione.

Simbolo	Codice	Simbolo	Codice
$=$	$=$	$\neq$	$\not =$
		$\neq$	$\neq$
		$\neq$	$\ne$
$\equiv$	$\equiv$	$\not\equiv$	$\not \equiv$
$\sim$	$\sim$	$\not\sim$	$\not \sim$
$\simeq$	$\simeq$	$\not\simeq$	$\not \simeq$
$\approx$	$\approx$	$\not\approx$	$\not \approx$
$\cong$	$\cong$	$\not\cong$	$\not \cong$
$\asymp$	$\asymp$	$\not\asymp$	$\not \asymp$
$\bowtie$	$\bowtie$		
$\doteq$	$\doteq$		
$\parallel$	$\parallel$		
$\perp$	$\perp$	$\not\perp$	$\not \perp$
$<$	$<$	$\not<$	$\not <$
$\leq$	$\leq$	$\not\leq$	$\not \leq$
$\le$	$\le$	$\not\le$	$\not \le$
$>$	$>$	$\not>$	$\not >$
$\geq$	$\geq$	$\not\geq$	$\not \geq$
$\ge$	$\ge$	$\not\ge$	$\not \ge$
$\prec$	$\prec$	$\not\prec$	$\not \prec$
$\preceq$	$\preceq$	$\not\preceq$	$\not \preceq$
$\succ$	$\succ$	$\not\succ$	$\not \succ$
$\succeq$	$\succeq$	$\not\succeq$	$\not \succeq$
$\ll$	$\ll$		
$\gg$	$\gg$		
$\subset$	$\subset$	$\not\subset$	$\not \subset$
$\subseteq$	$\subseteq$	$\not\subseteq$	$\not \subseteq$
$\supset$	$\supset$	$\not\supset$	$\not \supset$
$\supseteq$	$\supseteq$	$\not\supseteq$	$\not \supseteq$
$\sqsubseteq$	$\sqsubseteq$	$\not\sqsubseteq$	$\not \sqsubseteq$
$\sqsupseteq$	$\sqsupseteq$	$\not\sqsupseteq$	$\not \sqsupseteq$
$\mid$	$\mid$		
$\smile$	$\smile$		
$\frown$	$\frown$		
$\in$	$\in$	$\notin$	$\not \in$
$\ni$	$\ni$	$\not\ni$	$\not \ni$
$\owns$	$\owns$	$\not\owns$	$\not \owns$
$\vdash$	$\vdash$		
$\dashv$	$\dashv$		
$\models$	$\models$		
$\propto$	$\propto$		

Tabella 49.253. Frecce.

Simbolo	Codice	Simbolo	Codice
$\leftarrow$	$\leftarrow$	$\rightarrow$	$\rightarrow$
$\gets$	$\gets$	$\to$	$\to$
$\Leftarrow$	$\Leftarrow$	$\Rightarrow$	$\Rightarrow$
$\longleftarrow$	$\longleftarrow$	$\longrightarrow$	$\longrightarrow$
$\Longleftarrow$	$\Longleftarrow$	$\Longrightarrow$	$\Longrightarrow$
$\hookleftarrow$	$\hookleftarrow$	$\hookrightarrow$	$\hookrightarrow$
$\leftharpoonup$	$\leftharpoonup$	$\rightharpoonup$	$\rightharpoonup$
$\leftharpoondown$	$\leftharpoondown$	$\rightharpoondown$	$\rightharpoondown$
$\rightleftharpoons$	$\rightleftharpoons$		
$\iff$	$\iff$		
$\mapsto$	$\mapsto$		
$\longmapsto$	$\longmapsto$		
$\leftrightarrow$	$\leftrightarrow$	$\Leftrightarrow$	$\Leftrightarrow$
$\longleftrightarrow$	$\longleftrightarrow$	$\Longleftrightarrow$	$\Longleftrightarrow$
$\uparrow$	$\uparrow$	$\downarrow$	$\downarrow$
$\Uparrow$	$\Uparrow$	$\Downarrow$	$\Downarrow$
$\updownarrow$	$\updownarrow$	$\Updownarrow$	$\Updownarrow$
$\nwarrow$	$\nwarrow$	$\nearrow$	$\nearrow$
$\swarrow$	$\swarrow$	$\searrow$	$\searrow$
$\overleftarrow{ab}$	$\overleftarrow {ab}$	$\overrightarrow{ab}$	$\overrightarrow {ab}$
$\vec{ab}$	$\vec {ab}$		

Tabella 49.254. Ellissi.

Simbolo	Codice	Annotazioni
$\cdots$	$\cdots$	Ellissi orizzontale al centro.
$\ldots$	$\ldots$	Ellissi orizzontale in basso.
$\vdots$	$\vdots$	Ellissi verticale.
$\ddots$	$\ddots$	Ellissi diagonale.

Tabella 49.255. Altri simboli matematici.

Simbolo	Codice	Annotazioni
$\sum$	$\sum$	Sommatoria.
$\prod$	$\prod$	
$\coprod$	$\coprod$	
$\int$	$\int$	Integrale.
$\oint$	$\oint$	
$\bigcap$	$\bigcap$	Intersezione.
$\bigcup$	$\bigcup$	Unione.
$\bigsqcup$	$\bigsqcup$	
$\bigvee$	$\bigvee$	
$\bigwedge$	$\bigwedge$	
$\bigoplus$	$\bigoplus$	
$\biguplus$	$\biguplus$	
$\bigotimes$	$\bigotimes$	
$\bigodot$	$\bigodot$	

Eventualmente, è possibile modificare in modo sistematico il tipo di carattere da usare negli ambienti matematici. Si usano per questo due istruzioni alternative:

```
\everymath={macro_alternativa}
```

```
\everydisplay={macro_alternativa}
```

La prima si riferisce agli ambienti matematici inclusi nel testo normale, mentre la seconda riguarda gli ambienti che creano un blocco separato dal testo.

La macro alternativa a cui si fa riferimento può essere quella che definisce il tipo di carattere da usare. Si osservi l'esempio seguente:

```
\everymath={\bf}
\everydisplay={\rm}
la funzione $f(x)$ bla bla bla:
$$y = {{x^2}\over 2}$$
```

Si può vedere nel risultato seguente che nel testo normale si ottiene, mentre nel blocco separato si ha un tondo normale:

la funzione $\mathbf{f(x)}$ bla bla bla:

$$y = \frac{x^2}{2}$$

49.17.2 Dimensione del testo matematico

Il testo matematico, ovvero i simboli che si usano per le espressioni matematiche, viene dimensionato in modo automatico, salvo l'uso diretto di alcune istruzioni apposite. Le istruzioni sono nell'ordine: '**`\displaystyle`**', '**`\textstyle`**', '**`\scriptstyle`**' e '**`\scriptscriptstyle`**'. La prima rappresenta la dimensione più grande, mentre l'ultima genera il carattere più piccolo.

Anche se in condizioni normali non è necessario il controllo diretto della dimensione del testo matematico, è bene conoscere questa possibilità che può rivelarsi utile in presenza di formule particolarmente complesse in cui alcune parti rischiano di diventare troppo piccole per la lettura.

A titolo di esempio viene mostrata una frazione piuttosto articolata, in due modi differenti, anche se le frazioni vengono descritte in un'altra sezione:

```
$$x+{y\over x+{y\over x+{y\over x+{y\over x+{y\over
z}}}}}$$\par
$$x+{y\over \displaystyle x+{y\over \displaystyle x+{y\over
\displaystyle x+{y\over \displaystyle x+{y\over z}}}}}$$\par
```

$$x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{z}}}}}$$

$$x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{z}}}}}$$

Esiste la possibilità di usare '**\strut**', anche nell'ambito matematico, ma questo non previene la riduzione della dimensione dei caratteri. Viene mostrato lo stesso esempio in cui si controlla lo spazio verticale disponibile con l'aiuto di '**\strut**':

```
\hsize=14cm
$$x+{y\over \strut x+{y\over \strut x+{y\over \strut
x+{y\over \strut x+{y\over z}}}}}$$\par
```

$$x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{x+\cfrac{y}{z}}}}}$$

49.17.3 Punteggiatura

Dal momento che nell'ambiente matematico non si tiene conto degli spazi inseriti nel testo, lo spazio attorno ai segni di punteggiatura è gestito automaticamente. Si può osservare l'esempio seguente:

```
$123456.7890$\par
$123456,7890$\par
$123456:7890$\par
$123456;7890$\par
$123456{,}7890$\par
```

Come si può intuire, se si vuole usare la virgola come separatore della parte intera da quella rimanente, occorre delimitarla in un gruppo a sé stante, per evitare che gli sia aggiunto uno spazio superfluo alla sua destra.

$$123456.7890$$
$$123456,7890$$
$$123456:7890$$
$$123456;7890$$
$$123456{,}7890$$

Un discorso particolare va fatto per i due punti, che possono servire per indicare una divisione, oppure un assegnamento. Si osservi l'esempio seguente in cui si riprende un pezzo di quanto già mostrato:

```
$123456:7890$\par
$123456:=7890$\par
$123456\colon 7890$\par
```

In pratica, per ottenere i due punti «normali», a cui segue un po' di spazio, si deve usare l'istruzione '**\colon**'.

$$123456:7890$$
$$123456:=7890$$
$$123456\colon 7890$$

49.17.4 Frazioni e simili

Le frazioni si indicano solitamente in uno dei due modi seguenti:

numeratore* / *denominatore

numeratore `\over` ***denominatore***

Il secondo modo è quello più gradevole esteticamente. È importante osservare che l'istruzione '**\over**' interviene su tutto il gruppo in cui è contenuta. Si osservino gli esempi seguenti:

- `$${x+y \over z}$$`

$$\frac{x+y}{z}$$

- `$${x \over y+z}$$`

$$\frac{x}{y+z}$$

- `$${a \over x} + {b \over x} = {a+b \over x}$$`

$$\frac{a}{x}+\frac{b}{x}=\frac{a+b}{x}$$

Oltre alle frazioni vere e proprie, ci sono altre situazioni simili:

x `\atop` ***y***

x `\choose` ***y***

x `\above` ***spessore y***

Nel primo caso, si ottiene la sovrapposizione di ***x*** sopra ***y***, senza la linea orizzontale tipica di una frazione; nel secondo si ottiene la stessa cosa, ma il tutto viene anche racchiuso tra parentesi; nell'ultimo caso, si ottiene una cosa simile alla frazione, dove si specifica lo spessore della linea di separazione. Seguono alcuni esempi:

- `$${x+y \atop z}$$`

$$\begin{matrix}x+y\\z\end{matrix}$$

- `$${x+y \choose z}$$`

$$\binom{x+y}{z}$$

- `$${x+y \above 1mm z}$$`

$$\begin{matrix}x+y\\ \hline z\end{matrix}$$

Si osservi che anche in questo caso, le istruzioni mostrate spezzano il gruppo in cui sono inserite, pertanto è necessario racchiudere l'espressione tra parentesi graffe.

La dimensione del testo che si trova a essere spezzata con queste istruzioni viene ridotta automaticamente. Se si vuole evitare questo comportamento, si possono usare le istruzioni già descritte per il controllo esplicito della dimensione.

49.17.5 Apici e pedici

I pedici e gli apici si ottengono con i simboli '_' e '**^**' rispettivamente, che intervengono sul carattere oppure sul raggruppamento successivo:

x*_*pedice

x*^*apice

Si osservino gli esempi successivi, in particolare per quanto riguarda la necessità o meno di raggruppare ciò che va messo ad apice o a pedice:

- `$$10^2$$`

$$10^2$$

- `$$10^{20}$$`

$$10^{20}$$

- `$$10^{-2}$$`

$$10^{-2}$$

- `$$x^a_b$$`

$$x^a_b$$

- `$$x^{y^z}$$`

$$x^{y^z}$$

- `$\sum_{x=0}^n x^3$`

$\sum_{x=0}^n x^3$

- `$$\sum_{x=0}^n x^3$$`

$$\sum_{x=0}^n x^3$$

- `$$\int_0^n f(x)$$`

$$\int_0^n f(x)$$

- `$\lim_{n\to 0} {1 \over n} = \infty$`

$\lim_{n\to 0} \frac{1}{n} = \infty$

- `$$\lim_{n\to 0} {1 \over n} = \infty$$`

$$\lim_{n\to 0} \frac{1}{n} = \infty$$

49.17.6 Radici

Le radici si possono ottenere attraverso due modi differenti, a seconda che si tratti delle radici quadrate comuni, oppure di radici di altro genere:

```
\sqrt{contenuto}
```

```
\root n \of {contenuto}
```

Come si intuisce, nel primo caso si ottiene una radice quadrata pura e semplice, mentre nel secondo si dichiara il tipo di radice e successivamente il contenuto. Seguono alcuni esempi:

- `$$\sqrt{x^2+y^2}$$`

$$\sqrt{x^2+y^2}$$

- `$$\root 3 \of {8}$$`

$$\root 3 \of {8}$$

- `$$\sqrt{x+y \over z}$$`

$$\sqrt{x+y \over z}$$

- `$$x^{\sqrt{y}}$$`

$$x^{\sqrt{y}}$$

49.17.7 Sottolineature e sopralineature

Per ottenere delle linee orizzontali sopra o sotto un gruppo di simboli, si usano rispettivamente '**`\overline`**' e '**`\underline`**':

```
\overline{contenuto_posto_sotto_la_linea}
```



```
\underline{contenuto_posto_sopra_la_linea}
```

Seguono alcuni esempi:

- `$$\overline{x+y}$$`

$$\overline{x+y}$$

- `$$\underline{\overline{x+y}}$$`

$$\underline{\overline{x+y}}$$

49.17.8 Funzioni

Quando si vuole fare riferimento al nome standard di una certa funzione matematica, è necessario fare in modo che questo nome appaia con un carattere diverso dal corsivo matematico, per evitare che si possa confondere con il prodotto di una serie di variabili. Per questa ragione e anche per facilitare la lettura del sorgente TeX, esistono delle macro specifiche, con il solo scopo di scrivere il nome della funzione corrispondente usando un carattere tondo normale. La tabella 49.318 elenca brevemente queste macro. Se per qualche ragione si preferiscono dei nomi differenti, si può sempre usare la tecnica del testo nella scatola orizzontale, attraverso l'istruzione '**`\hbox{`*testo*`}`**'. Viene mostrato solo qualche esempio di utilizzo:

- `$$\sin(2x)$$`

$$\sin(2x)$$

- `$$\cos x$$`

$$\cos x$$

- `$$\hbox{sen}(2x)$$`

$$\hbox{sen}(2x)$$

Benché si possa scrivere il nome di una funzione usando la macro '**`\hbox{`*testo*`}`**', TeX offre l'istruzione primitiva '**`\mathop`**', con cui è possibile definire delle macro aggiuntive da affiancare a quelle standard per la scrittura dei nomi di funzione. Si usa normalmente così:

```
\def\nome{\mathop{\rm nome}}
```

Successivamente, è sufficiente fare riferimento alla macro '**`\`*nome*'' per ottenere il nome della funzione relativa in un contesto matematico. Segue l'esempio della dichiarazione e dell'uso della funzione seno, con il nome «sen», assieme al risultato che si ottiene:

```
\def\sen{\mathop{\rm sen}}
$$\sen(2x)$$
```

$$\sen(2x)$$

Tabella 49.318. Funzioni standard.

Simbolo	Codice	Annotazioni
arccos	$\arccos$	arco-coseno
arcsin	$\arcsin$	arco-seno
arctan	$\arctan$	arco-tangente
arg	$\arg$	argomento
cos	$\cos$	coseno
cosh	$\cosh$	coseno iperbolico
cot	$\cot$	cotangente
coth	$\coth$	cotangente iperbolica
csc	$\csc$	cosecante
deg	$\deg$	gradi
det	$\det$	determinante
dim	$\dim$	dimensione
exp	$\exp$	esponenziale
gcd	$\gcd$	massimo comune divisore
hom	$\hom$	
inf	$\inf$	
ker	$\ker$	
lg	$\lg$	logaritmo base 2
lim	$\lim$	limite
lim inf	$\liminf$	
lim sup	$\limsup$	
ln	$\ln$	logaritmo naturale
log	$\log$	logaritmo base 10
max	$\max$	massimo
min	$\min$	minimo
Pr	$\Pr$	probabilit
sec	$\sec$	secante
sup	$\sup$	
tan	$\tan$	tangente
tanh	$\tanh$	tangente iperbolica

49.17.9 Delimitatori

In matematica si usano tre tipi di parentesi per delimitare delle sottoespressioni; si tratta notoriamente di parentesi tonde, quadre e graffe. Le stesse parentesi sono disponibili nella modalità matematica di TeX, con la particolarità delle parentesi graffe che devono essere usate attraverso due macro speciali: '**\{**' e '**\}**'. In pratica, questa differenza dipende dall'uso speciale che TeX stesso fa di queste parentesi nella sua sintassi.

In condizioni normali, la grandezza delle parentesi usate non è diversa da quella del carattere standard dell'ambiente matematico. Per ottenere parentesi più grandi si possono usare alcune macroistruzioni come '**\bigl**' e '**\bigr**', le quali devono essere seguite immediatamente da una parentesi o da un altro simbolo di delimitazione, allo scopo di ottenerlo più grande del normale. Si osservi l'esempio seguente:

```
$$\Biggl(\biggl\{\Bigl[\bigl(a+b\bigr)\times
c\Bigr]+d\biggr\}\times e\Biggr)$$
```

$$\left(\left\{\left[\left(a+b\right)\times c\right]+d\right\}\times e\right)$$

In modo più semplice si possono usare le istruzioni '**\left**' e '**\right**', le quali adattano automaticamente le dimensioni delle parentesi o di altri delimitatori, in funzione di ciò che devono contenere. Si osservi l'esempio seguente:

```
$$\left(\left\{\left[\left(a+b\right)\times
c\right]+d\right\}\over e\right)$$
```

$$\left(\frac{\left\{\left[\left(a+b\right)\times c\right]+d\right\}}{e}\right)$$

Si osservi che l'altezza delle parentesi non aumenta in presenza di annidamenti successivi, ma soltanto quando la dimensione del contenuto lo richiede.

Tabella 49.323. Delimitatori in sintesi.

Simbolo	Codice
( ) [ ] { }	$($ $)$ $[$ $]$ $\{$ $\}$
( ) [ ] { }	$\bigl($ $\bigr)$ $\bigl[$ $\bigr]$ $\bigl\{$ $\bigr\}$
( ) [ ] { }	$\Bigl($ $\Bigr)$ $\Bigl[$ $\Bigr]$ $\Bigl\{$ $\Bigr\}$
( ) [ ] { }	$\biggl($ $\biggr)$ $\biggl[$ $\biggr]$ $\biggl\{$ $\biggr\}$
( ) [ ] { }	$\Biggl($ $\Biggr)$ $\Biggl[$ $\Biggr]$ $\Biggl\{$ $\Biggr\}$
(x) $\left(\frac{x}{y}\right)$	$\left(x\right)$ $\left(x\over y\right)$
⌊ ⌋ ⌈ ⌉	$\lfloor$ $\rfloor$ $\lceil$ $\rceil$
⟨ ⟩	$\langle$ $\rangle$
/ \	$/$ $\backslash$
∣ ∥	$\vert$ $\Vert$
↑ ↓	$\uparrow$ $\downarrow$
⇑ ⇓	$\Uparrow$ $\Downarrow$
↕ ⇕	$\updownarrow$ $\Updownarrow$

La tabella 49.323 mostra un elenco sintetico dei delimitatori e dell'uso delle istruzioni che consentono di ingrandirli. Naturalmente, le istruzioni '**\left**' e '**\right**', le macro '**\bigl**', '**\bigr**', fino a '**\Biggl**' e '**\Biggr**', possono essere usate con tutti i tipi di delimitatori matematici disponibili, non solo le parentesi comuni.

49.17.10 Matrici e sistemi di equazioni

Le matrici si rappresentano in modo molto semplice attraverso le macroistruzioni '**\pmatrix**' e '**\matrix**'. La differenza tra le due sta solo nel fatto che nel primo caso si ottiene la delimitazione del contenuto con parentesi tonde di altezza adeguata, mentre nel secondo caso queste devono essere inserite espressamente:

```
\[p]matrix{
elemento_1_1 [& elemento_1_2]... elemento_1_m \cr
[elemento_2_1 [& elemento_2_2]... elemento_2_m \cr]
...
elemento_n_1 [& elemento_n_2]... elemento_n_m \cr
}
```

In pratica, il raggruppamento che segue la macro contiene una serie di elementi organizzati in tabella, separati orizzontalmente dal simbolo '**&**' e conclusi alla fine di ogni riga da '**\cr**'. L'esempio seguente mette a confronto due matrici, realizzate con '**\pmatrix**' e '**\matrix**':

```
$$\pmatrix{
a & b & c \cr
d & e+f & g \cr
h & i & j \cr
}$$

$$\left |
\matrix{
a & b & c \cr
d & e+f & g \cr
h & i & j \cr
}
\right |
$$
```

$$\begin{pmatrix} a & b & c \\ d & e+f & g \\ h & i & j \end{pmatrix}$$

$$\begin{vmatrix} a & b & c \\ d & e+f & g \\ h & i & j \end{vmatrix}$$

Gli elementi della matrice sono spaziati in pratica da uno spazio leggermente elastico che può essere forzato facilmente. Per esempio, volendo allineare le celle a sinistra, o a destra, basta usare nel lato opposto uno spazio un po' più elastico:

```
$$\left |
\matrix{
a & b \hfill & c \cr
d & e+f & g \cr
h & \hfill i & j \cr
}
```

```
\right |
$$
```

$$\left|\matrix{a & b & c \cr d & e+f & g \cr h & i & j}\right|$$

Naturalmente, queste macroistruzioni si possono utilizzare anche per qualcosa di diverso dalle matrici, che però è bene appaia incasellato in forma tabellare:

```
$$|x| =
\bigg \{
\matrix{
x & x \geq 0 \cr
-x & x \leq 0 \cr
}
$$
```

$$|x| = \begin{cases} x & x \geq 0 \\ -x & x \leq 0 \end{cases}$$

Per rappresentare un insieme di «casi», come nella definizione di una funzione, si può usare la macro '**`\cases`**':

```
\cases{ espressione & condizione \cr
    espressione & condizione \cr
    …
    espressione & condizione \cr }
```

Come si vede dal modello sintattico, si distinguono espressioni e condizioni relative: le espressioni sono inserite in un ambiente matematico, mentre le condizioni si trovano in un ambiente di testo normale.

```
$$f(x) = \cases{
    2x & se $x > 0$ \cr
    {3x \over 2}+x^2 & se $x < 0$ \cr
}$$
```

$$f(x) = \begin{cases} 2x & \text{se } x > 0 \\ \frac{3x}{2} + x^2 & \text{se } x < 0 \end{cases}$$

In modo analogo, si usa la macro '**`\eqalign`**' per allineare delle equazioni; in tal caso però, tutti i dati si trovano nell'ambiente matematico:

```
\eqalign{ espressione & espressione \cr
    espressione & espressione \cr
    …
    espressione & espressione \cr
```

Dato il nome (*equation align*) si intende che l'allineamento debba avere luogo, preferibilmente, in corrispondenza del segno di uguaglianza (o disuguaglianza) usato per abbinare le espressioni. Si osservi l'esempio seguente:

$$\begin{aligned} x + y &= y + x \times 1 \\ (x \times y)^1 &= y \times x \\ \frac{x}{y} &\neq \frac{y}{x} \end{aligned}$$

Infine, un gruppo di espressioni può essere rappresentato su più righe attraverso la macro '**`\displaylines`**', in modo molto simile a '**`\matrix`**':

```
\displaylines{ espressione_1 \cr
    espressione_2 \cr
    espressione_3 \cr
    …
    espressione_n \cr }
```

Viene mostrato un esempio con il risultato che si ottiene:

```
$$\displaylines{ x+y = y+x \cr
    x \times y = y \times x \cr
    {x\over y} \not = {y \over x} \cr }
$$
```


$$\begin{gathered} x + y = y + x \\ x \times y = y \times x \\ \frac{x}{y} \neq \frac{y}{x} \end{gathered}$$

49.17.11 Dichiarazione di teoremi e corollari

È possibile definire facilmente un teorema o un corollario attraverso la macro '**`\proclaim`**', secondo lo schema seguente:

```
\proclaim titolo. definizione
```

In pratica, si distingue una parte iniziale del testo che segue la macro, fino al primo punto, che viene evidenziato in modo più nero rispetto al resto. Evidentemente, la macroistruzione '**`\proclaim`**' viene usata in un ambiente normale (non matematico) e può contenere ambienti matematici nell'ambito della definizione. Si osservi l'esempio seguente:

```
\proclaim Teorema 1. In teoria, non c'\'e differenza tra
pratica e teoria.\par
\proclaim Teorema 2. In pratica, la teoria \'e diversa dalla
pratica.\par
```

Teorema 1. *In teoria, non c'è differenza tra pratica e teoria.*

Teorema 2. *In pratica, la teoria è diversa dalla pratica.*

È importante tenere presente che la dichiarazione non può essere più lunga di una riga.

49.17.12 Equazioni in evidenza

La gestione di equazioni poste nell'ambiente matematico esterno al testo normale, ovvero quello che viene posto in evidenza al centro della giustezza del testo, è particolare.

In modo simile alle matrici, è possibile allineare le equazioni, ma per questo si usa preferibilmente la macroistruzione '**`\eqalign`**', che prevede l'uso di un solo simbolo '**`&`**' per ogni riga. Si osservi l'esempio:

```
$$\eqalign{
a+b &= c        \cr
a+c &= b+c-1    \cr
b+c &= 5        \cr
}$$
```

$$\begin{aligned} a + b &= c \\ a + c &= b + c - 1 \\ b + c &= 5 \end{aligned}$$

Eventualmente, la macroistruzione '**`\equalignno`**', consente di mostrare qualche informazione in più in prossimità del margine destro. Si ottiene ciò individuando una colonna aggiuntiva nella tabella immaginaria, come si può vedere dall'esempio:

```
Testo normale, bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla.
$$\eqalignno{
a+b &= c        & (1)   \cr
a+c &= b+c-1    & (2)   \cr
b+c &= 5        \cr
}$$
```

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

$$\begin{aligned} a + b &= c && (1) \\ a + c &= b + c - 1 && (2) \\ b + c &= 5 \end{aligned}$$

Nello stesso modo, la macroistruzione '**`\lequalignno`**' mostra le annotazioni a sinistra:

```
Testo normale, bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
bla bla bla bla.
$$\leqalignno{
a+b &= c        & (1)   \cr
a+c &= b+c-1    & (2)   \cr
b+c &= 5        \cr
```

```
}$$
```

Testo normale, bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

(1) $$a+b=c$$

(2) $$a+c=b+c-1$$

$$b+c=5$$

Eventualmente, se c'è la necessità di annotare un'espressione che non ha bisogno di essere rappresentata in forma tabellare, si possono usare le istruzioni '**\eqno**' e '**\leqno**':

```
$$a+b = c       \eqno (1)$$
$$b+c = b+c-1   \leqno (2)$$
```

$$a+b=c \tag{1}$$

(2) $$b+c=b+c-1$$

49.18 Riferimenti

«

- Documentazione interna della distribuzione TeX: '`texmf/doc/*`'
- Documentazione interna Info: *info web2c* e *info latex*
- Marco Pratesi, Marco Latini, Michele Antonecchia *LATEX-PDF-Howto*, 2003, *http://www.telug.it/marco/LaTeX-PDF-HOWTO/LaTeX-PDF-HOWTO.html*
- *General documentation about (La)TeX*, *http://tex.loria.fr/english/general.html*
- Michael Doob, *A gentle introduction to TeX*, *http://www.ctan.org/tex-archive/info/gentle/gentle.pdf*
- Daniel M. Zirin, *TeX made easy, Using TeX with plain macro package*, 1993, Zar Limited, *http://tex.loria.fr/general/all.dvi*
- D. R. Wilkins, *Getting started with plain TeX*, 1993, *http://tex.loria.fr/general/pllong.dvi*
- D. R. Wilkins, *Summary of commonly-used features of plain TeX*, 1993, *http://tex.loria.fr/general/plshort.dvi*
- *TeX cookbook*, 1989, MathPro Press, *http://tex.loria.fr/general/cookbook.dvi*
- Sorgenti della definizione del formato '**tex**', contenuti nella directory '`texmf/tex/plain/base/`' di una distribuzione teTeX
- Sorgenti della definizione del formato '**latex**', contenuti nel file '`texmf/tex/latex/base/latex.ltx`' di una distribuzione teTeX
- Claudio Beccari, *LaTeX, Guida a un sistema di editoria elettronica*, Hoepli, 1991, ISBN 88-203-1931-4

[1] È consentito anche l'uso della virgola per separare la parte intera di un numero. Tuttavia, ci sono situazioni in cui ciò non va bene, per cui è meglio usare sempre solo il punto.

[2] TeX ammette che tra il numero e l'unità di misura ci sia dello spazio. Qui si preferisce attaccare l'unità di misura al numero, per evitare confusione con il testo successivo.

[3] Si distingue tra «parametro formale» e «argomento attuale», per indicare, nel primo caso, ciò che si usa nella dichiarazione di una macroistruzione a rappresentare i dati variabili da fornire, mentre nel secondo caso si fa riferimento ai dati che vengono passati effettivamente, in corrispondenza di un certo parametro della dichiarazione formale.

[4] Si possono creare delle alternative anche a simboli di controllo, ma in generale ciò non è conveniente.

[5] Leggendo la tabella, è il caso di osservare che in condizioni normali, la sequenza di controllo '**\$**' genera il simbolo del dollaro, mentre se il testo è in corsivo, si ottiene il simbolo della sterlina inglese.

[6] Si osservi che in questo caso non si può usare un raggruppamento vuoto al posto dello spazio, perché l'accento risulterebbe spostato rispetto al centro della lettera.

[7] Nella lingua italiana sono sufficienti due tipi di trattini.

[8] Questi spazi verticali aggiuntivi possono essere inseriti solo se ci si trova in modalità verticale, in pratica nello spazio tra i paragrafi e non al loro interno.