

Capitolul 2

INTRODUCERE ÎN LIMBAJUL LaTeX

2.1 INTRODUCERE

O lucrare este predată editorului care, după verificarea semantică de către un grup de recenzori, va accepta lucrarea. O lucrare acceptată pentru publicare va trebui aranjată în pagină, operație făcută de către tehnoredactor, care hotărăște dimensiunile paginii, fonturile, aspectul general etc. Aceste informații formează *layout*-ul lucrării ce va trebui publicată. Zețarul culege textul, urmând acest model și scoate o primă copie, ce va fi supusă corecturilor gramaticale. După corectură, lucrarea se tipărește în numărul dorit de exemplare.

Dacă recenzia este făcută de oameni, celelalte operații pot fi efectuate cu ajutorul calculatorului, pe baza unor reguli simple și fixe. Pentru aceasta, este necesară utilizarea unui procesor de texte, iar din motive legate de răspândire și disponibilitate, am ales LaTeX.

În scenariul de mai sus, LaTeX este tehnoredactorul și TeX este zețarul. TeX este un program elaborat de Donald E. Knuth [Knuth 1984], care se folosește la culegerea și tipărirea de texte și formule matematice. LaTeX (se pronunță *la-tehi*) este construit de Leslie Lamport [Lamport 1994], are la bază TeX și permite definirea de *layout*-uri pentru cărți, articole, rapoarte și scrisori.

LaTeX este un pachet de macrodefiniții pentru TeX, iar TeX este un sistem de prelucrare a textelor. Nici LaTeX și nici TeX *nu* sunt editoare de texte.

În general, *layout*-ul paginilor realizate în MS Word (i.e. poziția figurilor, a tabelelor, etc.) poate să varieze dramatic, atunci când textul este portat la un calculator ce are instalat alt driver de imprimantă. Dependența de

imprimantă este neplăcută și este necesar un procesor independent. Acest fenomen nu apare în cazul procesării textelor în LaTeX și constituie, printre altele, unul din principalele motive ale utilizării sale, pe scară din ce în ce mai largă, de către editorii și autorii de texte. Pe de altă parte, existența LaTeX, pe diferite platforme, face din el un procesor de texte cu largă utilizare în mediile științifice.

2.1.1 Un exemplu simplu

Când tehnoredactorul lucrează la *layout*, el recunoaște, în text, intențiile autorului privind titlul de capitol, titlul de paragraf, citat, listă etc. Pentru ca LaTeX să recunoască aceste detalii, este necesar ca autorul să-i furnizeze informații suplimentare despre textul scris și forma de prezentare a acestuia.

Un program sursă în LaTeX se poate edita cu orice editor ASCII, de exemplu în Notepad. Comenzile LaTeX încep cu caracterul `\` și nu vor apărea în documentul final. De la caracterul `%` până la finalul unei linii LaTeX, toată informația este ignorată, fiind similară cu un comentariu în scrierea unui program.

Structura standard a unui document este, pe scurt, următoarea:

Preambulul - începe cu o comandă `\documentclass`; el poate conține comenzi `\usepackage`, declarații globale și definiții de cadre și comenzi;

```
\begin{document}
textul documentului
\end{document}
```

LaTeX nu este un sistem *What You See Is What You Get*, adică autorul nu vede pe monitor efectul comenzilor asupra documentului. În mod normal, autorul poate întrerupe editarea în orice moment, pentru a vizualiza fișierul rezultat după procesarea în LaTeX.

O caracteristică a LaTeX este aceea că autorul este lăsat să se concentreze, în primul rând, asupra organizării logice a lucrării, fără a consuma efort pentru formatarea textului.

În organizarea materialului unei cărți este necesară respectarea câtorva principii, pentru a asigura o bună lizibilitate a textului, comoditate la citirea unei pagini precum și organizarea logică a documentului. De multe ori, se dorește ca anumite părți ale textului să iasă în evidență, deși nu este indiferent modul în care acest lucru este realizat. LaTeX permite specificarea acelor părți

și el va alege forma potrivită pentru tipărire. Mulți autori utilizează fonturi multiple, fapt ce duce la o îngreunare a citirii textului. Aceste aspecte sunt importante atunci când se proiectează un *layout*. În general, LaTeX permite modificarea unui *layout* de o manieră destul de greoaie, însă integrează o serie de concepte ce au rezultat în urma experienței privind tehnoredactarea. Aceste concepte respectă convențiile, valabile în S.U.A., de realizare a cărților, articolelor și revistelor. Adaptarea la convențiile europene de tehnoredactare se poate realiza relativ ușor, prin comenzi speciale.

Trebuie subliniat faptul că orice text (sub formă de carte, articol, raport etc.) este scris pentru a fi citit și pentru ca impactul conținutului asupra cititorului să fie maxim. Din acest motiv, pe baza unei analize riguroase a situațiilor specifice privind conținutul unui text, în LaTeX documentele sunt împărțite în clase sau categorii, iar lizibilitatea este obiectivul principal.

În fig. 2.1 este descris un exemplu simplu de fișier LaTeX minimal, în care se evidențiază principalele secțiuni și câteva dintre comenzile LaTeX de bază.

Exemplul 2.1

```

\documentclass{article} % clasa articol
%Titlul cuprinsului este implicit "Contents" dar
%poate fi schimbat dupa dorinta
\renewcommand{\contentsname}{Cuprins}
%Se specifica titlul, autorul si data
\title{Gestiunea documentelor}
\author{Ion Popescu}
\date{17.03.2000}
\begin{document} %inceputul documentului
\maketitle %tiparire titlu, autor, data
\tableofcontents %tiparire cuprins
\section{Introducere} %sectiune paragraf
Paragraf specificand necesitatea procesorului LaTeX. Textul este
formatat automat.

Trecerea la un nou alineat se face prin inserarea unei linii
goale.
\section{Titlul capitolului 1} %inceputul altui paragraf
Acesta este alt paragraf. Toate paragrafele se numeroteaza
automat.
\end{document} %sfarsitul articolului

```

Fig. 2.1 – Un exemplu simplu de articol scris în LaTeX. Rezultatul procesării este redat în fig. 2.2.

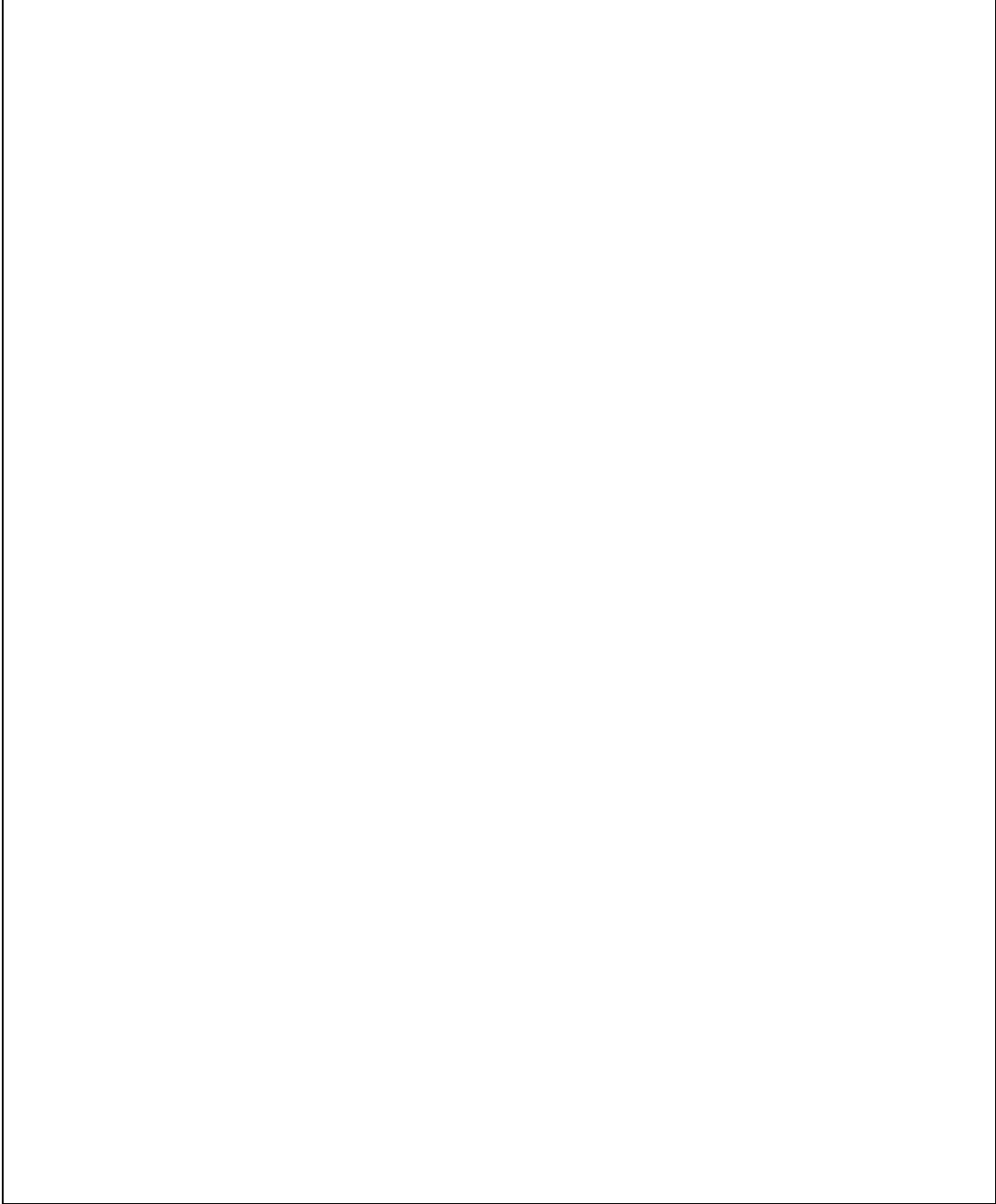


Fig. 2.2 – Rezultatul procesării fișierului sursă din fig. 2.1 (*setup*-ul paginii în LaTeX este diferit).



2.1.2 Realizarea documentelor în LaTeX

TeX și LaTeX sunt produse *shareware*, adică au prețuri scăzute sau sunt gratuite, dacă nu se folosesc în scopuri comerciale. Ele au o largă răspândire și au implementări sub toate sistemele de operare (Unix, OS/2, Windows etc.), asigurând portabilitatea documentelor realizate. Un produs de acest tip este însoțit de o bogată colecție de fonturi, utilitare și fișiere ce conțin documentația aferentă.

Folosind un editor de texte obișnuit, de exemplu Norton Editor sub DOS sau Notepad sub Windows, se construiește un fișier cu extensia `.tex`. Această extensie nu este obligatorie, dar prezența sa permite un mod de lucru mai comod și organizat. De exemplu, fișierul din fig.2.1 este salvat cu numele `ex1.tex`. Pentru procesare se va lansa LaTeX, având ca parametru numele fișierului cu extensia `.tex`. Dacă se folosește versiunea LaTeX pentru Win95 (a se vedea URL-ul www.stat.uiowa.edu/resources/help/win95/latex.html), atunci comanda este:

```
>c:/texmf/miktex/bin/latex ex1<ENTER>
```

Ca urmare, pe ecran apar mesajele:

```
This is TeX, Version 3.14159 (MiKTeX 1.20)
(ex1.tex
LaTeX2e <1998/12/01> patch level 1
Babel <v3.6x>...loaded.
(C:\texmf\tex\latex\base\article.cls
Document Class: article 1999/01/07 v1.4a Standard LaTeX
document class
Output written on ex1.dvi (1 page, 944 bytes)
Transcript written on ex1.log.
```

Zona marcată cu caracterele “...” semnifică informații dependente de versiunea de instalare. În funcție de conținut, un fișier `.tex` parcurge mai multe etape. După procesarea fișierului `ex1.tex`, informația se păstrează în mai multe fișiere, care au următoarele denumiri și semnificații:

`ex1.aux` - conține informații auxiliare despre entitățile logice din fișierul sursă și despre etichete (`\label` și `\bibitem`). Fișierul este utilizat de LaTeX la a doua procesare, pentru a rezolva referințe încrucișate (`\cite`, `\ref`).

`ex1.toc` - conține informații necesare alcătuirii cuprinsului și este generat doar dacă în fișierul sursă apare comanda `\tableofcontents`. Cuprinsul apare doar la a doua procesare a fișierului.

`ex1.log` - conține raportul despre procesarea fișierului cu extensia `tex`, eventualele erori de procesare și informații statistice despre resursele hardware utilizate.

`ex1.dvi` - fișier de ieșire ce conține rezultatele procesării într-un cod independent de perifericul de tipărire (DeVice Independent).

Observație. În funcție de clasa sau categoria din care face parte documentul, mai apar și alte fișiere auxiliare:

- * `.lof` – creat dacă apare comanda `\listoffigures` (List of Figures), pentru generarea automată a listei figurilor;
- * `.lot` – creat dacă apare comanda `\listoftable` (List of Tables), pentru generarea automată a listei tabelor;
- * `.idx` – creat dacă apare comanda `\makeindex` în combinație cu `\index` și va conține comenzile `\indexentry` aferente;
- * `.gb` – creat dacă apare comanda `\makeofglossary`.

□

Imaginea documentului poate fi vizualizată sub DOS cu ajutorul comenzii `v numefișier`. Sub X WINDOWS, vizualizarea se poate realiza utilizând comanda `xdvi numefișier`. Tipărirea se va realiza cu ajutorul unui utilitar ce poate accesa imprimanta. Pentru DOS, tipărirea la o imprimantă cu ace implică utilizarea utilitarului `dvidot`, cu o serie de parametri. De aceea, în general, pentru fiecare tip de imprimantă există o comandă distinctă. De exemplu pentru EPSON FX80 comanda este `prtfx numefișier`, iar pentru o imprimantă HEWLETT PACKARD LJ comanda este `dvihplj numefișier`. Pentru Windows95, vizualizarea și tipărirea se realizează cu utilitarul Yap. Sub sistemul de operare Unix se utilizează o procedură asemănătoare. Când se dispune de o imprimantă *PostScript*, utilitarul `dvi2ps` transformă fișierul `*.dvi` într-un fișier `*.ps` (i.e. *PostScript*).

Observație. La referințe încrucișate sau la generarea cuprinsului, a listei de figuri, a indexului, este necesară procesarea de mai multe ori a sursei `*.tex`, pentru că este nevoie de fișiere care la prima procesare nu au fost create încă.

□

2.2 NOȚIUNI FUNDAMENTALE PRIVIND LaTeX

Putem privi LaTeX ca un limbaj de programare în care instrucțiunile au anumite efecte.

O serie de caractere au semnificație specială:

\$ & % # _ { } ~ ^ " \ | < >

Cu unele excepții, aceste caractere nu pot fi folosite în textul sursă. Unele dintre ele se pot utiliza în text cu ajutorul secvențelor de control listate mai jos, pe coloana din dreapta:

\$ & % # _ { } |\$ |& \% \# _ \{ \}

De multe ori, caracterul *blank* (spațiu) are același efect dacă apare o singură dată sau ca un șir de *blank*-uri.

Caracterul *newline* (linie nouă) este echivalent cu *blank*, atunci când apare o singură dată. Mai mult de două astfel de caractere succesive semnifică terminarea unui paragraf și începutul unui nou.

2.2.1 Preliminarii privind secvențe de control și fonturi

Secvențele de control sunt șiruri de caractere ce reprezintă comenzi LaTeX. De exemplu, trecerea la un alt paragraf se face cu secvența de control `\par`.

O secvență de control, i.e. o comandă în LaTeX, este compusă fie dintr-un caracter *backslash* (adică `\`) urmat de o succesiune de litere, fie dintr-un caracter *backslash* urmat de un caracter diferit de literă. De exemplu, `\today` are ca efect scrierea datei mașinii, iar `\%` produce caracterul `%`. Comenzile din prima categorie sunt valabile doar pentru primul șir până la caracterul *blank*.

Orice comandă poate avea argumente obligatorii și/sau opționale. Un argument obligatoriu este cuprins între acolade `{ și }`, iar unul opțional între paranteze drepte `[și]`.

O comandă `\LaTeX` ignoră spațiile ce o urmează.

Exemplul 2.2

Execuția fișierului sursă următor

```
\documentclass{report}
\begin{document}
\LaTeX este un program de tehnoredactare
\end{document}
```

are drept rezultat textul

După cum se observă, era necesar un *blank* după cuvântul “LaTeX”, iar un caracter *backslash* plasat la sfârșitul comenzii rezolvă problema. □

Exemplul 2.3

```
\LaTeX\ este un program de tehnoredactare
```

are ca rezultat textul

Observație. În cele ce urmează, în cadrul exemplelor prezentate, dacă acestea apar pe două coloane, fișierul sursă **.tex* apare în dreapta, iar textul rezultat este inserat pe coloana din stânga. Dacă ele sunt prezentate grupate orizontal, atunci fișierul **.tex* apare primul. În primele exemple a fost folosită mai ales clasa *report* dar, dacă nu se specifică altfel, comenzile din fișierul **.tex* respectiv sunt valabile pentru toate clasele de documente. □

Caracterul *rând nou* este echivalent cu *spațiu* și, ca urmare, paragrafele se vor separa printr-un rând gol sau utilizând comanda `\par`. □

Exemplul 2.4

	<pre>\documentclass{report} \begin{document} Paragraf1. Tot el. Paragraf2. \par Paragraf3. \end{document}</pre>
--	--

LaTeX dispune de un set bogat de fonturi. Scrierea textului se face implicit cu un font numit *Roman*, însă se pot selecta alte fonturi cu o secvență □

de control `\tt` pentru fontul *typewriter* sau `\bf` pentru *bold face* (subsecțiunea 2.6.1).

Dacă numai câteva cuvinte se tipăresc cu alt font, atunci nu este necesară schimbarea fontului activ, ci se poate folosi comanda `\bf`.

Exemplul 2.5

```
\documentclass{report}
\begin{document}
...font curent oarecare {\bf doresc bold} revin la fontul curent.
\end{document}
```

□

O pereche de paranteze acolade formează un grup, deci o comandă are efect doar asupra aceluși grup. Pentru transmiterea parametrilor la secvențele de control cu parametri, se vor utiliza tot paranteze acolade. Prin parametru se înțelege orice șir de caractere între acolade sau un caracter, dacă acoladele lipsesc. De exemplu, comanda

```
\hspace {4cm}
```

indică inserarea unui spațiu orizontal de 4 cm.

Observație. Acoladele ce delimitează un parametru nu pot forma un grup. Ca urmare, dacă se dorește refacerea valorii unei variabile LaTeX, după modificarea sa prin executarea unei comenzi corespunzătoare, atunci chiar comanda trebuie inclusă în grup.

□

2.2.2 Facilități matematice

Formulele matematice sunt culese diferit față de textele obișnuite. În LaTeX se pot identifica două moduri de lucru:

- modul text – în care se culege textul;
- modul matematic – în care se culeg formulele.

Există comenzi ce specifică trecerea de la un mod la altul și comenzi ce sunt valabile doar într-un singur mod.

Intrarea în modul matematic se poate realiza în mai multe variante. În cadrul paragrafului, formulele se delimitează prin două caractere $\$$.

Exemplul 2.6

	<pre>\documentclass{report} \begin{document} Pentru \$x=y\$ rezulta ... \end{document}</pre>
--	--

În mod matematic, caracterele $_$ și $^$ indică indici inferiori, respectiv superiori, iar o formulă pusă pe rânduri separate se delimitează cu câte o pereche de secvențe de caractere $\$$.

Exemplul 2.7

	<pre>\documentclass{report} \begin{document} Pentru \$x_1, x_2, \dots, x_{n+1}\$ rezulta ca \$\$x_i = x_{i-1} + x_{i-2}\$\$ \end{document}</pre>
--	--

Pentru caractere speciale se vor utiliza secvențele de control $\backslash\alpha$, $\backslash\beta$ sau $\backslash\frac$ cu parametrii aferenți.

Exemplul 2.8

	<pre>\documentclass{report} \begin{document} Sciem: \$\$\frac{1}{2+\alpha} = \frac{1}{2\alpha+1}\$\$ \end{document}</pre>
--	---

În formule pot să apară operatori, cum ar fi: $\backslash\sum$ pentru Σ , $\backslash\prod$ pentru Π sau $\backslash\int$ pentru \int . Indicii operațiilor sunt poziționați diferit la formulele plasate în text, față de cele separate.

Exemplul 2.9

	<pre>\documentclass{report} \begin{document} ... \$\sum_{i=1}^{ny_i=0}\$ \$\$\sum_{i=1}^{ny_i=0}\$\$... \end{document}</pre>
--	---

Există operatori la care nu se face această diferențiere (de exemplu \int), însă limitele pot să apară sau nu, prin plasarea comenzilor `\limits`, respectiv `\nolimits`.

Secvențele de control `\left` și `\right` se pot aplica asupra parantezelor, pentru ajustarea automată a dimensiunii acestora în funcție de conținut. Ele apar în perechi.

Exemplul 2.10

	<pre> \documentclass{report} \begin{document} \$x+2\left(\frac{ \sin x}{x+ \cos\left(x+1\right)\right)}=0\$ \end{document} </pre>
--	---

□

2.2.3 Adaptarea la diacritice și despărțirea în silabe

Diacriticele se pot obține, în LaTeX, prin folosirea *accentelor*, ca opțiune oferită de o serie de secvențe de control ce acceptă orice tip de caracter, având la bază alfabetul latin.

Exemplul 2.11

	<pre> \documentclass{report} \begin{document} \^A, \^a, \u A, \u a, \^I, \^i, \c S, \c s, \c T, \c t. \par Aten\c tie: NU \^i. Accentul se aplic\u a semnului\par \i (i f\u ar\u a punct). \end{document} </pre>
--	--

□

O altă variantă ce ușurează editarea este cea de creare a unui fișier cu extensia `.sty`, cu rolul de a defini diacriticele [Pusztai 1994]. Acest fișier va fi creat în funcție de limba în care se editează și va fi înglobat în sursa LaTeX.

Pentru despărțirea în silabe, se poate utiliza un set de șabloane ce sunt alese la inițializarea LaTeX. Pentru a specifica o anumită limbă, se va utiliza comanda `\language` cu un parametru numeric. Noi reguli pot fi introduse

prin comenzile `\-`, care are efect local, sau `\hyphenation`, cu efect global. De exemplu, comanda `\hyphenation{ma-ma}` are ca efect despărțirea în silabe a cuvântului *mama*, oriunde este întâlnit în text. Scrierea cuvântului *concesie* în forma `con\ -ce\ -si\ -e` are ca efect despărțirea în silabe a sa doar în locul indicat.

Detalii suplimentare privind comenzile de generare a accentelor sunt prezentate în subsecțiunea **2.6.2**.

2.2.4 Titluri redactate în diverse limbi

Anumite comenzi LaTeX produc un text predefinit, reprezentând, în general, titlul unei unități secționale. De exemplu, comanda `\chaptername` are ca efect scrierea titlului capitolului, precedat de “Chapter”. Această scriere trebuie modificată, dacă textul este redactat în alte limbi. Pentru aceasta, în preambulul fișierului sursă `*.tex` se va folosi comanda

```
\renewcommand{\chaptername}{nume_capitol}
```

unde *nume_capitol* este numele capitolului în limba respectivă (a se vedea și secțiunea **2.5**). În mod obișnuit, secvența de control `\chaptername` este o comandă LaTeX care este apelată de `\chapter` și are ca efect *implicit* tipărirea șirului de caractere “Chapter”.

Comenzile ale căror definiții trebuie modificate pentru adaptarea la limba în care este redactat textul lucrării sunt următoarele (cu precizarea, în paranteze și între apostroafe, a textului generat implicit):

```
\partname - numele dat unei părți, în clasele de documente book și
report;
\chaptername - numele dat capitolului, în clasele de documente book și
report (“Chapter”);
\bibname - numele dat bibliografiei, în clasele de documente book și
report (“Bibliography”);
\refname - numele dat bibliografiei în clasa de documente article
(“References”);
\appendixname - numele dat anexelor (“Appendix”);
\abstractname - numele dat rezumatului (“Abstract”);
\indexname - numele dat indexului (“Index”);
\contentsname - numele dat cuprinsului (“Contents”);
```

```

\figurename - numele dat figurilor (“Figure”);
\listfigurename - numele dat listei figurilor (“List of Figures”);
\tablename - numele dat tabelelor (“Table”);
\listtablename - numele dat listei tabelelor (“List of Tables”).

```

În continuare se prezintă sintaxa câtorva comenzi `\renewcommand`, utile pentru redactarea unui text în limba română.

```

\renewcommand{\contentsname}{Cuprins}
\renewcommand{\listfigurename}{Lista figurilor}
\renewcommand{\listtablename}{Lista tabelelor}
\renewcommand{\figurename}{Figura}
\renewcommand{\tablename}{Tabelul}
\renewcommand{\chaptername}{Capitolul}
\renewcommand{\bibname}{Bibliografie}
\renewcommand{\refname}{Bibliografie}
\renewcommand{\appendixname}{Anexa}

```

2.2.5 Organizarea fișierului de intrare

2.2.5.1 Categorii de documente

Un fișier LaTeX începe cu o comandă `\documentclass`, având sintaxa:

```
\documentclass [options] {class}
```

în care *options* este lista de una sau mai multe opțiuni, separate prin virgulă și fără spații suplimentare, iar *class* este categoria sau clasa documentului.

Printre categoriile de documente valabile în LaTeX se numără:

- book
- report
- article
- letter
- slides.

Clasa `book` specifică organizarea documentului sub formă de carte, având următoarele caracteristici:

- culegerea diferențiată a textului pe pagini pare și impare;
- organizarea pe părți, secțiuni și subsecțiuni;
- dacă nu se specifică prin comanda `\pagestyle`, antetul (*header*-ul) paginii pare conține titlul capitolului curent, iar cel al paginii impare conține titlul secțiunii curente și fiecare capitol începe pe o pagină impară;

Clasa `report` permite editarea documentului sub formă de raport, asemănătoare cu `book`, dar culegerea se face nediferențiat pentru pagini pare și impare, iar capitolele pot începe pe pagini atât pare cât și impare.

Clasa `article` înseamnă forma de articol, fără părți și capitole. Antetele sunt implicit goale și culegerea este nediferențiată de paritatea paginii.

Clasa `letter` permite redactarea unei scrisori oficiale.

Clasa `slides` este recomandată pentru pregătirea documentelor ce vor fi prezentate pe folii expuse la retroproiector.

Toate clasele standard de documente, cu excepția `slides`, acceptă următoarele opțiuni pentru selectarea mărimii literei tipărite (10 pt implicit):

10pt, 11pt, 12pt.

Toate clasele de documente acceptă următoarele opțiuni pentru setarea mărimii paginii (implicit `letter`):

`a4paper`, `a5paper`, `b5paper`, `letterpaper`, `legalpaper`, `executivepaper`.

Alte opțiuni mixte sunt:

- `landscape` – selectează formatul *landscape*, dar implicit este *portrait*;
- `titlepage`, `notitlepage` –specifică dacă trebuie să apară o pagină separată de titlu;
- `leqno` – numărul ecuației este plasat *la stânga*, dar implicit el este *la dreapta*;
- `fleqn` – înseamnă “FlushLeft EquatioN” și determină alinierea *la stânga* a ecuației, care implicit este *centrată*;
- `german` – pune la dispoziție comenzi utile pentru realizarea textelor în limba germană;
- `makeidx` – este utilă la documentele în care se dorește obținerea (semi)automată a unui index.

Următoarele opțiuni *nu* sunt valabile pentru clasa `slides`:

- `oneside`, `twoside` – selectează afișarea pe o față sau pe ambele ale unei coli. Valoarea implicită este `oneside`, cu excepția clasei `book`.
- `openright`, `openany` – determină dacă un capitol începe pe pagina din dreapta. Valoarea implicită este `openright`, pentru clasa `book`.
- `onecolumn`, `twocolumn` – permite tipărirea pe una sau, respectiv, două coloane. Valoarea implicită este `onecolumn`.

2.2.5.2 Principalele comenzi utile la crearea unui document

În continuare, sunt descrise principalele comenzi utile la crearea unui document.

`\begin{document}` este comanda de început de document. Între `\documentclass` și `\begin{document}` nu se poate insera text, dar pot apărea alte comenzi LaTeX, care nu au ca efect generarea de text. Această zonă se numește *preamble* (secțiunea 2.1). Aici poate apărea comanda `\title`, ce stabilește titlul, dar nu și comanda `\maketitle`, ce pune titlul în pagină.

`\end{document}` marchează sfârșitul documentului și tot ce urmează după această comandă este ignorat la procesare.

`\title` specifică titlul documentului. Dacă acesta nu încapă într-un singur rând, atunci LaTeX nu împarte automat titlul pe rânduri, lăsând aceasta la latitudinea autorului, prin comanda `\\` acolo unde se trece la rând nou. Acest lucru preîntâmpină despărțirea unui titlu în rânduri, ceea ce ar putea să altereze semnificația dată de autor. Sintaxa comenzii este

```
\title{text}
```

și, pentru a fi operațională, ea trebuie să apară înainte de comanda `\maketitle`.

`\author` permite specificarea numelui autorului. Ca și la `\title`, trecerea pe mai multe rânduri trebuie marcată cu aceeași comandă `\\`. Dacă sunt mai mulți autori, separarea numelor lor pe rânduri se face prin comanda `\and`.

`\date` setează data producerii documentului, iar dacă nu există, atunci la `\maketitle` se va lua automat data sistemului.

`\maketitle` pune în pagină titlul, autorul și data realizării documentului. Comanda apare prima după `\begin{document}` și efectul său este diferit în funcție de clasa documentului:

- pentru `book` și `report`, titlul, autorul și data apar pe o pagină separată, numerotată cu 0;
- pentru `article`, informațiile apar la început de pagină, urmate de text (a se vedea și exemplul 2.1, subsecțiunea 2.1.1).

Notele de subsol se pot insera într-unul din următoarele moduri:

- 1) se folosește doar comanda `\footnote`;
- 2) se utilizează comenzile `\footnotemark` și `\footnotetext`.

Comanda `\footnote` având sintaxa:

```
\footnote [number] {text}
```

plasează textul *text*, numerotat, la subsolul paginii. Argumentul opțional *number* este necesar dacă se dorește schimbarea numărului implicit al notei. Această comandă poate fi folosită doar *în afara* modului `paragraph`, i.e. nu poate fi folosită în secționarea unor comenzi precum `\chapter`, în figuri, în tabele sau într-un mediu tabular.

Comanda `\footnotemark` pune numărul notei de subsol în text. Ea poate fi utilizată *în interiorul* modului `paragraph`, iar textul notei de subsol este specificat prin comanda `\footnotetext`, având sintaxa

```
\footnotetext [number] {text}
```

`\footnotetext` poate să apară, oriunde, după comanda `\footnotemark`, *în afara* modului `paragraph`. Argumentul opțional *number* este utilizat când se dorește schimbarea numărului implicit al notei. Se pot produce mai multe marcaje consecutive ale unei note prin comanda

```
\footnotemark [\value{footnote}]
```

după prima comandă `\footnote`.

Comanda

```
\thanks {text}
```

are ca efect o notă de subsol atașată titlului, cu un simbol distinct de cele generate prin comanda `\footnote`.

Exemplul 2.12

	<pre> \documentclass[11pt]{article} \title{Un titlu\\lung} \author{Ionescu\footnote {adresa Ionescu}\\Popescu} \date{1.11.1999} \begin{document} \maketitle Textul articolului..... \end{document} </pre>
--	---

□

2.2.5.3 Secționarea unui document în unități

O lucrare se poate împărți în *părți*, *capitole*, *secțiuni*, *subsecțiuni* și *subsubsecțiuni*, iar apoi în *paragrafe* și *subparagrafe*. Pentru manevrarea și aranjarea în pagină a acestora, LaTeX permite utilizarea unor comenzi de secționare, a căror listă este prezentată în continuare:

- `\part` (doar pentru clasele report și book)
- `\chapter` (doar pentru clasele report și book)
- `\section`
- `\subsection`
- `\subsubsection`
- `\paragraph`
- `\subparagraph`

Toate comenzile de secționare au aceeași formă generală. De exemplu, pentru capitol, forma comenzii este:

```
\chapter[titlu_scurt]{title}
```

În afară de locul unde se secționează textul documentului, argumentul *title* al comenzii poate să apară în două alte locuri, și anume: (1) în cuprins și (2) în antetul (*header*-ul) paginii curente. Dacă *nu* se dorește ca *title* să apară în text la fel ca în pozițiile specificate la (1) și (2), atunci se setează argumentul *titlu_scurt*. Ca urmare, parametrul *title* va fi afișat în text, iar parametrul *titlu_scurt* este utilizat pentru cuprins și antet, unde poate avea semnificația unui titlu scurt.

Comenzile de secționare suportă și forma cu “*” la sfârșit. De exemplu:

```
\chapter*{title}
```

care are ca rezultat tipărirea argumentului *title*, nenumerotat și neinclus în cuprins. La clasele *book* și *report*, secțiunea se numerotează cu două cifre i.e. capitol.secțiune, iar la clasa *article* numerotarea conține doar o cifră. De exemplu, în clasa *article*, 4.7 înseamnă a 7-a subsecțiune a Secțiunii 4, iar în clasa *report* aceeași subsecțiune se poate numerota 5.4.1, unde “5” este numărul capitolului. În felul acesta, un document *article* poate fi inclus, ca un capitol, într-un document *report*.

Comanda

```
\appendix
```

permite schimbarea modului de numerotare a anexelor (de exemplu A.1, A.1.1) și folosește aceleași comenzi de secționare ca și documentul principal. Comanda `\appendix` nu generează text. În fișierul **.tex*, o posibilă secvență de instrucțiuni este următoarea:

```
\chapter{titlul_primului_capitol}
...
\appendix
\chapter{titlul_primului_appendix}
```

2.2.5.4 Etichete și referiri

LaTeX numerotează singur entitățile din text, deși aceste numere nu se cunosc în momentul editării fișierului **.tex*. Când este necesară referirea unor capitole, secțiuni, figuri, ecuații sau tabele, se folosesc comenzile

```
\label{key}
\ref{key}
```

unde *key* este un nume simbolic pentru entitatea respectivă, ce constă dintr-o secvență oarecare de litere, cifre sau semne de punctuație. Literele mari și mici sunt interpretate distinct. Comanda `\label`, plasată în textul uzual, asignează numărul unității curente prin parametrul *key*; dacă ea apare în interiorul unui mediu numerotat, atunci acel număr este asignat argumentului *key*.

Pentru a evita crearea accidentală a două etichete cu același nume, se utilizează etichete constând dintr-un *prefix* și un *suffix*, separate prin caracterul “:”. Prefixele convenționale utilizate sunt:

- `cha` pentru `chapters`
- `sec` pentru `sections`
- `fig` pentru `figures`
- `tab` pentru `tables`
- `eq` pentru `equations`

Astfel, o etichetă pentru o figură are forma `fig:nome`.

Comanda `\ref` produce numărul unității secționale, numărul ecuației etc., din comanda `\label` corespunzătoare.

Deosebit de utilă este și comanda

```
\pageref{key}
```

care permite referirea la pagina unde este plasată entitatea etichetată de `\label`, ca în exemplul următor.

Exemplul 2.13

```
\documentclass{report}
\renewcommand{\chaptername}{Capitolul}
\begin{document}
\chapter{Introducere}
\section{\^Inceputul}\label{inceput}
text 1
\section{Dezvoltare}
\^In sec\c tiunea \ref{inceput} de la pagina
\pageref{inceput}este vorba despre...
\end{document}
```



□

Observație. Reamintim că, dacă se folosesc referiri etichetate, atunci fișierul `*.tex` trebuie *procesat de două ori*, pentru rezolvarea referințelor încrucișate, deoarece la prima procesare se scriu, în fișierul `*.aux`, doar informații despre entitățile etichetate. Abia la a doua procesare aceste informații sunt preluate pentru a fi folosite de comenzile `\ref` și `\pageref`.

□

2.2.5.5 Generarea cuprinsului, a listei figurilor și a listei tabelor

Generarea automată a cuprinsului, a listei figurilor și a listei tabelor se realizează, respectiv, utilizând comenzile:

```

\tableofcontents
\listoffigures
\listoftables

```

Fiecare din aceste comenzi se pune în locul în care se dorește să apară lista respectivă. Ca rezultat, este generat un fișier (*header*), dar nu se începe automat o nouă pagină. Dacă se dorește o pagină nouă după cuprins, respectiv după lista de figuri sau de tabele, atunci se include comanda `\newpage` după comanda `\tableofcontents`, respectiv după `\listoffigures` sau după `\listoftables`. Fișierul are același nume cu fișierul rădăcină și, respectiv, extensia

```

.toc pentru \tableofcontents,
.lof pentru \listoffigures sau
.lot pentru \listoftables.

```

Scrierea altor fișiere, în afară de cele cu extensia `.dvi` și respectiv `.log`, poate fi inhibată prin comanda `\nofiles`.

Adăugarea de noi linii, în oricare din cele trei fișiere, se poate realiza prin comenzile

```

\addcontentsline{file}{sec_unit}{entry}
\addtocontents{file}{text}

```

unde:

- *file* este o extensie a fișierul în care trebuie scrisă informația și poate fi toc, lof, sau lot;
- *sec_unit* poate fi:
 - nume de unitate de document (part, section, chapter, etc.), dacă fișierul are extensia `.toc`,
 - figure, dacă fișierul are extensia `.lof` sau
 - table, dacă fișierul are extensia `.lot`;
- *entry* reprezintă textul inserat.

2.3 CADRE

LaTeX pune la dispoziție mai multe tipuri de cadre (în engleză *environments*), cu scopul delimitării unor entități, cum ar fi ecuații, tabele, figuri, șiruri, liste sau, în general, paragrafe cu diverse caracteristici distincte de ale textului ordinar.

Orice cadru începe și se sfârșește în același mod:

```
\begin{environment_name}
text
\end{environment_name}
```

unde *environment_name* este numele cadrului, iar *text* este conținutul cadrului.

Numele cadrului poate fi unul din următoarele:

- `array` (șiruri și tabele cu simboluri matematice)
- `center` (linii centrate)
- `description` (liste etichetate)
- `enumerate` (liste numerotate)
- `eqnarray` (secvențe de ecuații aliniate)
- `equation` (ecuație)
- `figure` (figuri)
- `flushleft` (paragraf aliniat la stânga)
- `flushright` (paragraf aliniat la dreapta)
- `itemize` (listă marcată cu *bullets*)
- `letter` (scrisoare)
- `list` (listă generică)
- `minipage` (pagină miniaturală)
- `picture` (figură cu text, săgeți, linii și cercuri)
- `quotation` (paragraf indentat cu prima linie indentată)
- `quote` (paragraf indentat cu prima linie neindentată)
- `tabbing` (alinieră oarecare a textului)
- `table` (tabele flotante)
- `tabular` (alinieră textului în coloane)
- `thebibliography` (bibliografie)
- `titlepage` (pagină specială de titlu, fără numerotare)
- `verbatim` (lucrurile în regim “mașină de scris”)

- `verse` (pentru versuri etc.)

În plus față de cadrele listate mai sus, se numără cadrele `document` și `abstract`.

Cadrele sunt similare cu alte structuri de control din limbajele de programare, delimitate de deschideri și închideri corespunzătoare.

Cadrul obligatoriu pentru orice fișier LaTeX este `document`.

Cadrul `abstract` permite așezarea în pagină a rezumatului și este opțional pentru clasele `article` și `report` și nedefinit pentru clasele `book`, `letter` și `slides`.

Iată un exemplu simplu.

Exemplul 2.14

	<pre> \documentclass{article} \title{A Title} \author{Ionescu} \date{1.11.1999} \begin{document} \maketitle \begin{abstract} This is the abstract... \end{abstract} This is the text..... \end{document} </pre>
--	---

□

În continuare sunt prezentate succint, în ordine alfabetică, cadrele listate, cu accent pe acele facilități ce permit crearea în LaTeX a textelor științifice.

2.3.1 Cadrul array

```
\begin{array} [pos] {cols}
col1 & col2 & \dots & coln \\
.
.
\end{array}
```

Șirurile și tabelele matematice de tip matriceal se construiesc utilizând cadrul `array`. Argumentul `cols` (similar cu cel al cadrului `tabular`, a se vedea subsecțiunea 2.3.19) descrie numărul de coloane și alinierea elementelor în interiorul lor și este compus, de regulă, din secvențe de caractere (câte unul pentru fiecare coloană), prin care se indică poziția elementelor în coloană:

- `c` (*centred*) – centrat;
- `l` (*left*) - aliniat la stânga;
- `r` (*right*) - aliniat la dreapta.

`pos` este un argument opțional, care specifică poziționarea verticală în cadru; poziționarea implicită este centrat, iar opțiunile sunt:

- `t` (*top*) - aliniere la marginea de sus a liniei;
- `b` (*bottom*) - aliniere la marginea de jos a liniei

Conținutul unei coloane este separat, de următoarea coloană, prin simbolul `&`. Acest conținut poate include alte comenzi LaTeX. Fiecare linie a tabelului matriceal trebuie terminată cu secvența de simboluri `\\`.

Cadrul `array` poate fi utilizat doar în modul matematic (secțiunea 2.7), astfel încât, în general, el este deschis și închis în interiorul unui cadru `equation` sau apare între două simboluri `$`.

Exemplul 2.15

	<pre>\documentclass{report} \begin{document} \$A(x)=\left(\begin{array}{ccc} x & 0 & 0 \\ 1 & 0 & x \end{array}\right)\$ \end{document}</pre>
--	--

□

2.3.2 Cadrul `center` și comanda `\centering`

```
\begin{center}
Text linia 1 \\
Text linia 2 \\
.
.
.
\end{center}
```

Cadrul `center` permite crearea unui paragraf, care constă din linii centrate între marginea din stânga și marginea din dreapta, pe pagina curentă. Fiecare linie trebuie terminată cu șirul `\\`, ce delimitează conținutul unui rând.

Comanda

```
\centering
```

corespunde cadrului `center` și poate fi utilizată în interiorul unor cadre precum `quote` sau într-un *parbox*. Figurile sau tabelele pot fi centrate în pagină, dacă se dă comanda `\centering` la începutul cadrelor ce definesc entitățile respective.

Spre deosebire de cadrul `center`, comanda `\centering` nu începe un nou paragraf, ci schimbă doar modul în care LaTeX formatează unitatea de paragraf respectivă.

Exemplul 2.16

	<pre>\documentclass{report} \begin{document} \begin{center} un paragraf centrat\\ se pune ^\i ntr-un cadru\\ center \end{center} \end{document}</pre>
--	---

□

2.3.3 Cadrul `description`

```
\begin{description}
\item [label] Primul element
\item [label] Al doilea element
.
.
.
\end{description}
```

Cadrul `description` este utilizat pentru crearea listelor etichetate. Eticheta este un șir de caractere și apare aliniată *la dreapta* și tipărită cu caractere *bold*. În descrierea de mai sus, comenzile `\item` definesc lista, iar argumentul *label* definește eticheta.

Exemplul 2.17

	<pre>\documentclass{report} \begin{document} \begin{description} \item[List 1] este lista 1 \item[List 2] este lista 2 \end{description} \end{document}</pre>
--	---

□

2.3.4 Cadrul `enumerate`

```
\begin{enumerate}
\item Primul element
\item Al doilea element
.
.
.
\end{enumerate}
```

Cadrul `enumerate` generează o listă numerotată. Numerotările pot fi încuibate, până la o adâncime de patru niveluri. De asemenea, ele pot fi încuibate într-un alt cadru.

Fiecare entitate a unei liste numerotate începe de la comanda `\item`.

În interiorul cadrului trebuie să existe cel puțin o comandă `\item`.

Cadrul `enumerate` folosește contorii `enumi` până la `enumiv` (subsecțiunea 2.4.1), iar tipul de numerotare se poate modifica, redefinind `\theenumi`.

Exemplul 2.18

	<pre> \documentclass{report} \begin{document} The steps are: \begin{enumerate} \item for i=1:n do \item i:=i+1 \begin{enumerate} \item j:=i \end{enumerate} \end{enumerate} \item end \end{enumerate} \end{document} </pre>
--	---

□

2.3.5 Cadrul `eqnarray` și cadrul `eqnarray*`

```

\begin{eqnarray}
formula matematică 1 \\
formula matematică 2 \\
.
.
.
\end{eqnarray}

```

Cadrul `eqnarray` este folosit pentru a scrie o secvență de ecuații sau de inegalități. Seamănă, oarecum, cu un cadru `array` cu trei coloane, având coloanele consecutive separate prin șirul `\\`, iar elementele consecutive dintr-o linie separate prin `&`. Formulele se pot alinia după simbolul cuprins, pe fiecare linie, între semnele `&` (de exemplu după semnul `=`, scriind `&=&`). Fiecare linie este numerotată, cu excepția cazului în care se întâlnește comanda `\nonumber`.

Comanda `\leftteqn` este folosită pentru a scrie o formulă lungă pe mai multe linii. Argumentul său este tipărit aliniat la stânga față de restul

liniilor ce cuprind formula.

Cadrul `eqnarray*` acționează la fel ca `eqnarray`, cu excepția faptului că nu generează numerotarea liniilor.

Exemplul 2.19

	<pre> \documentclass{report} \begin{document} \begin{eqnarray} x_n&=&x_{n-1}+x_{n-2}\\\ f(x)&=&x^2\\\ \lefteqn{g(x)=x^2+y^2} \&&+z^2 \end{eqnarray} \end{document} </pre>
--	---

□

2.3.6 Cadrul equation

```

\begin{equation}
formula matematica
\end{equation}

```

Cadrul `equation` centrează ecuația pe pagină și plasează numărul ecuației în marginea din dreapta. Clasa documentului determină modul cum sunt numerotate ecuațiile (subsubsecțiunile **2.2.5.3** și **2.2.5.4**).

Exemplul 2.20.

	<pre> \documentclass{report} \begin{document} \begin{equation} \lim_{n\rightarrow \infty} \left(1+\frac{1}{n} \right)^n=e \end{equation} \begin{equation} F(s)=\int_0^{\infty} f(t) e^{-st}dt \end{equation} \end{document} </pre>
--	--

□

2.3.7 Cadrul figure

```
\begin{figure} [placement]
corpul figurii
\caption{titlul figurii}
\end{figure}
```

Figurile sunt obiecte care nu fac parte din textul propriu-zis și sunt “deplasate”, în mod normal, în locuri convenabile, de exemplu la începutul sau la sfârșitul paginii. O figură nu este tipărită pe pagini separate.

Argumentul opțional *placement* determină locul în care LaTeX va încerca să plaseze figura, și anume:

- `h` (*here*) – în locul din text în care apare cadrul `figure`;
- `t` (*top*) – la începutul paginii;
- `b` (*bottom*) – la sfârșitul paginii;
- `p` (*page of floats*) – pe o pagină separată, care nu conține text, ci numai obiecte din figura plasată de LaTeX.

Clasele standard `report` și `article` utilizează, ca amplasament *default*, `tbp`, astfel încât figura poate fi așezată la începutul paginii, la sfârșitul paginii sau pe o pagină separată.

Corpul figurii este compus, în general, dintr-un cadru `picture` centrat (subsecțiunea 2.3.14). Comanda `\caption` permite scrierea legendei figurii. Un exemplu de creare a unui desen va fi dat după discutarea cadrului `picture`.

2.3.8 Cadrul flushleft și comanda \raggedright

```
\begin{flushleft}
Text linia 1 \\
Text linia 2 \\
.
.
\end{flushleft}
```

Cadrul `flushleft` permite crearea de paragrafe aliniate la stânga. Fiecare linie trebuie să se termine cu secvența `\\`.

Comanda

`\raggedright`

corespunde cadrului `flushleft` și poate fi folosită în interiorul unui cadru cum ar fi `quote` sau într-un *parbox*. Spre deosebire de cadrul `flushleft`, comanda `\raggedright` nu începe un nou paragraf, ci doar schimbă formatul paragrafului curent, cuprins între `\begin{}` și `\end{}`.

Exemplu 2.21

	<pre> \documentclass{report} \begin{document} \begin{flushleft} Ne aliniem la st^anga, ^i ntr-un \cadru flushleft. Acesta este\ cunoscut drept un\ text "ragged- right". \end{flushleft} Sf^ar^c situl altui paragraf. \begin{quote} \raggedright Aici este un cadru quote,\ ale c^u arui linii sunt\ aliniate la st^anga. \end{quote} \end{document} </pre>
--	--

□

2.3.9 Cadrul `flushright` și comanda `\raggedleft`

```

\begin{flushright}
Text linia 1 \
Text linia 2 \
.
.
\end{flushright}

```

Cadrul `flushright` permite crearea de paragrafe aliniate la dreapta. Fiecare linie trebuie să se termine cu secvența `\`.

Comanda

```
\raggedleft
```

corespunde cadrului `flushright` și poate fi folosită în interiorul unui cadru cum ar fi `quote` sau `parbox`. Spre deosebire de cadrul `flushleft`, comanda `\raggedright` nu începe un nou paragraf, ci doar schimbă formatul paragrafului curent, cuprins între `\begin{}` și `\end{}`.

2.3.10 Cadrul `itemize`

```
\begin{itemize}
\item Primul element
\item Al doilea element
.
.
\end{itemize}
```

Cadrul `itemize` generează o listă “cu *bullets*”. Itemizarea poate fi încuibată una într-alta, până la patru niveluri în adâncime. De asemenea, aceste liste pot fi incorporate într-un paragraf creat cu alt cadru.

Fiecare element dintr-o listă începe cu comanda `\item`. În cadrul `itemize` trebuie să existe cel puțin o comandă `\item`.

Cadrul `itemize` poate utiliza cinci contori, de la `itemii` la `itemv` (subsecțiunea 2.4.1). Tipul de numerotare poate fi modificat redefinind `\theitemi`.

Exemplul 2.22

Ex22	<pre>\documentclass{report} \begin{document} \begin{itemize} \item Fiecare item ... bullet. \item Listele ... \^i ncuibate. \begin{enumerate} \item Etichetele ... sau litere. \item O list\u a ... 2item-uri. \end{enumerate} \LaTeX\ permite ... \^i ncuibare. \item O linie goal\u a dup\u a \par un item nu are efect. \end{itemize} \end{document}</pre>
------	---

□

2.3.11 Cadrul `letter`

Acest cadru este folosit pentru documente de tip scrisori și este tratat în Anexa A.

2.3.12 Cadrul `list`

```
\begin{list}{label}{spacing}
\item 1: Primul element
\item 2: Al doilea element
.
.
\end{list}
```

Cadrul `list` este un cadru generic, utilizat pentru a defini alte cadre specifice. În general, nu este folosit decât arareori în documente, dar în mod frecvent apare în macrodefiniții (secțiunea 2.5).

Argumentul *label* specifică modul cum sunt etichetate elementele listei (*items*). Argumentul este un text, inclus într-un *box* pentru a forma o etichetă. Acest argument poate conține alte comenzi LaTeX.

Argumentul *spacing* conține comenzi care setează parametrii de spațiere ai listei. Acest argument este, în general, nul, i.e. `{}`. În acest caz, sunt selectați parametrii de spațiere implicați (*default*), ceea ce este satisfăcător pentru majoritatea aplicațiilor.

În afara acestor dimensiuni, următoarele comenzi controlează formatarea.

`\makelabel{label}` este comanda care poziționează eticheta transmisă ca parametru opțional al comenzii `\item`. Poate fi redefinită de utilizator, cu ajutorul comenzii `\renewcommand`, în secțiunea de declarații.

`\usecounter{ctr}` pune la dispoziție contorul *ctr* (care trebuie mai întâi creat, conform subsecțiunii 2.4.1), pentru numerotarea internă a elementelor. Acest contor este automat inițializat la zero, iar la fiecare comandă `\item` fără parametru el va fi incrementat cu `\refstepcounter`, valoarea lui devenind valoarea curentă (subsecțiunea 2.4.1).

Exemplu 2.23

```
\documentclass{report}
\begin{document}
\newcounter{bean}
Textul ce precede lista.
\begin{list}
{B--\arabic{bean}}
{
\usecounter{bean}
\setlength{\rightmargin}{20mm}
\setlength{\leftmargin}{\rightmargin}
}
\item Acesta este primul element ... sunt egale cu 20mm.
\item Acesta este al doilea element.
\end{list}
^Incepe un nou paragraf doar ^\i n cazul ^\i n care cadrul
list este urmat de o linie goal\u a.

Un nou paragraf.
\end{document}
```



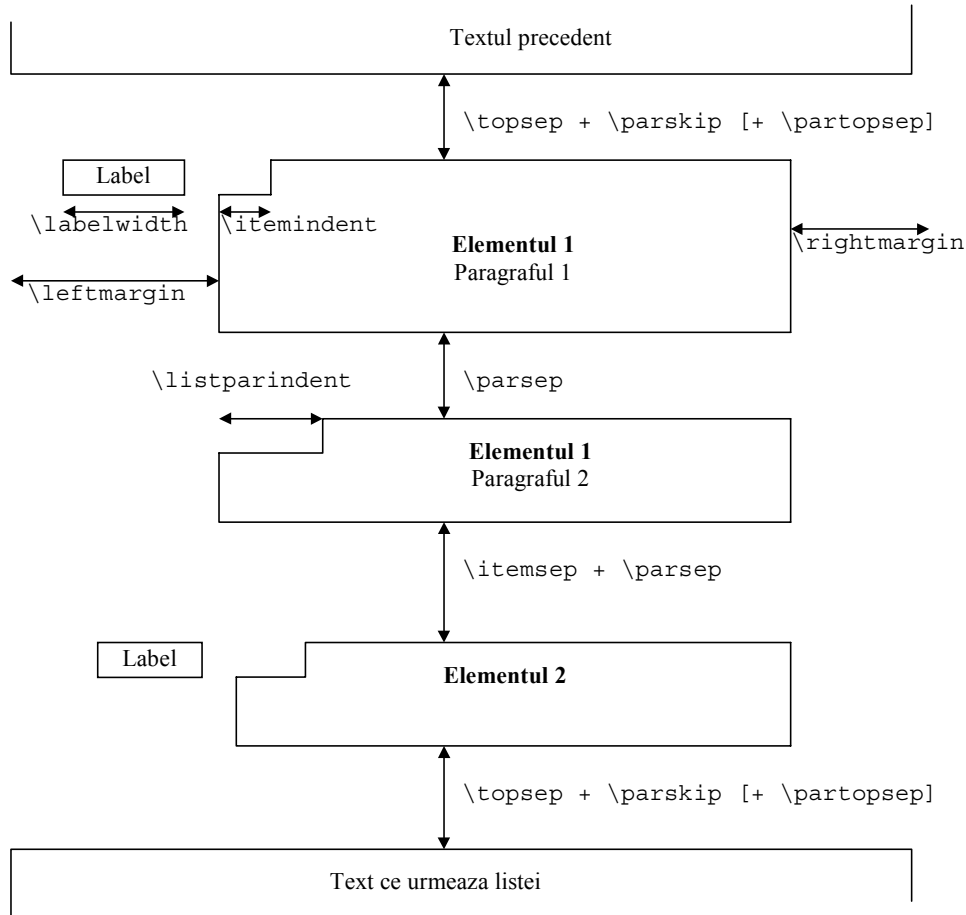


Fig. 2.3 - Semnificația parametrilor din cadrul `list` (a se vedea și subsecțiunea 2.4.2).

2.3.13 Cadrul `minipage` și comanda `\parbox`

```
\begin{minipage} [pos] {width}
  text
\end{minipage}
```

Parbox este o cutie (*box*) al cărei conținut este în modul paragraf, iar TeX generează o serie de linii ca în textul uzual. Cadrele *figure* și *table* creează *parbox*-uri. Există două modalități de a crea un *parbox* într-un

anumit loc din text: 1) cu ajutorul cadrului `minipage` și, respectiv, 2) prin comanda `\parbox`. Ele pot fi folosite pentru a include unul sau mai multe paragrafe de tip text, în interiorul unei figuri sau al unui tabel.

Cadrul `minipage` are un argument opțional *pos* și un argument *width* ce specifică lățimea *box*-ului. Argumentul opțional *pos* poate fi:

- *t* (*top*) – linia de sus a *parbox*-ului este aliniată la linia de text uzual sau
- *b* (*bottom*) – linia de jos a *parbox*-ului este aliniată la linia de text uzual.

În interiorul unui cadru `minipage`, se pot utiliza alte comenzi de tip cadru, ce creează paragrafe.

O comandă `\footnote` sau `\footnotetext` creează o notă de subsol în partea de jos a minipaginii și nu în subsolul paginii și utilizează contorul `mpfootnote` în locul contorului obișnuit pentru notele de subsol (subsecțiunea 2.4.1).

Observație. Dacă se utilizează note de subsol, atunci nu se include o minipagină în alta deoarece ele pot ajunge în subsolul altei minipagini decât cea corectă.

□

Exemplul 2.24

	<pre> \documentclass{report} \begin{document} \begin{minipage}[t]{0.5in} {\em vrabia} este o pas\u are \end{minipage} \ IAR \ \begin{minipage}[t]{1in} {\em iepurele} este un roz\u ator\footnote{vezi mamifere}. \end{minipage} \end{document} </pre>
--	--

□

Comanda

`\parbox`

este similară cu cadrul `minipage`, are același argument opțional *pos* și un argument *width*. Este utilizată, în general, pentru un *parbox* cu un volum redus de text, ce nu conține alte paragrafe create cu cadre.

Exemplul 2.25

	<pre> \documentclass{report} \begin{document} \parbox[b]{0.5in}{Parbox cu linia de JOS aliniat\u a cu linia TEXT.} \ TEXT \ \parbox[t]{1in}{Parbox cu linia de SUS aliniat\u a cu linia TEXT.} \end{document} </pre>
--	--

□

Un control mai fin al poziționării verticale se realizează cu comanda `\raisebox` (subsecțiunea 2.8).

2.3.14 Cadrul `picture` și folosirea sa în cadrul figure

```

\begin{picture} (width,height) (x_offset,y_offset)
  comenzi picture
\end{picture}

```

Cadrul `picture` permite crearea de desene ce conțin text, linii, săgeți și cercuri. În principiu, se comunică LaTeX locul unde se pun aceste obiecte prin specificarea coordonatelor lor.

O coordonată este un număr care poate avea zecimale și semn – de exemplu 5 sau -2.3 . O coordonată specifică lungimea, în multipli de unitatea `\unitlength`, astfel încât, dacă `\unitlength` a fost setată la 1cm, atunci coordonata 2.5 specifică o lungime de 2.5cm de la origine. Valoarea `\unitlength` poate fi modificată oricând, cu ajutorul comenzii `\setlength`. Această schimbare nu trebuie făcută însă în interiorul cadrului `picture`, deoarece efectul este imprevizibil.

O poziție este o pereche de coordonate – de exemplu, (2.4,-5), care specifică punctul cu coordonata 2.4 pe axa x și respectiv -5 pe axa y . Coordonatele sunt specificate, în mod uzual, în raport cu originea care este situată, de regulă, în colțul stânga-jos al desenului. Când o poziție apare ca argument, ea nu este inclusă între paranteze, acestea servind doar la delimitare.

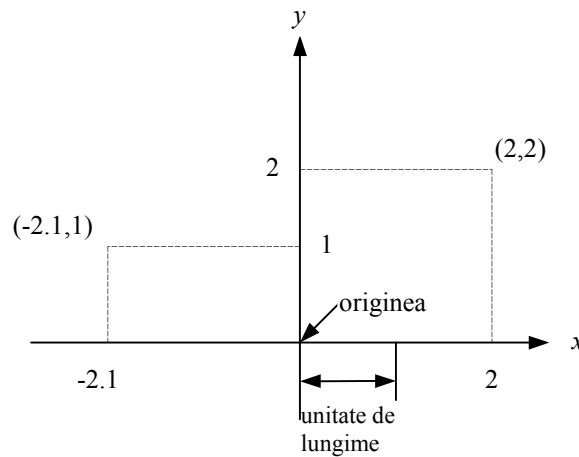


Fig.2.4 Puncte și coordonatele lor

Cadrul `picture` are un argument obligatoriu dimensional. Cadrul generează o cutie (*box*) dreptunghiulară, cu lungimea *width* și lățimea *height*, reprezente, respectiv, de coordonatele *x* și *y* ale argumentului.

Cadrul `picture` mai are și un argument opțional, ce urmează argumentului dimensional și care poate modifica originea. Spre deosebire de argumentele opționale uzuale, acesta *nu* este inclus între paranteze drepte. Acest argument opțional furnizează coordonatele noii origini, în raport cu colțul din stânga-jos al figurii. De exemplu, dacă `\unitlength` a fost setată la 1mm, comanda

```
\begin{picture}(100,200)(10,20)
```

generează o figură cu lățimea 100mm și lungimea 200mm, al cărei colț din stânga-jos este punctul (10,20) și al cărei colț dreapta-sus este, în consecință, (110,220). Când se desenează o figură, mai întâi se omite acest argument opțional, lăsându-se originea în colțul stânga-jos. Dacă apoi se dorește o deplasare a desenului, se adaugă argumentul opțional corespunzător.

Argumentul obligatoriu al cadrului determină mărimea nominală a desenului, dar desenul propriu-zis poate să depășească aceste dimensiuni.

Grosimea liniei cu care se desenează se selectează prin comenzile:

- `\thinlines` – selectează linia subțire;
- `\thicklines` – selectează linia groasă;

- `\linethickness{gros}` selectează penița groasă, iar argumentul *gros* specifică grosimea liniei.

Desenul este alcătuit cu comenzile:

`\put (coord_x,coord_y) {obj}` – pune obiectul *obj* în punctul de coordonate *coord_x, coord_y*;

`\multiput (coord_x,coord_y) (dx,dy) {nr} {obj}` – pune de *nr* ori obiectul *obj*, începând cu punctul de coordonate *coord_x, coord_y*, iar la fiecare pas coordonatele sunt incrementate cu *dx*, respectiv *dy*. Ca urmare, la a treia repetare a obiectului specificat prin *obj*, coordonatele sale vor fi *coord_x*3*dx*, respectiv *coord_y*3*dy*.

Comanda `\put` creează o cutie LR (secțiunea 2.8).

Obiectele reprezentate de argumentul *obj* pot fi text uzual sau desene create de comenzile descrise în continuare.

`\circle[*]{diam}` generează un cerc cu un diametru cât mai apropiat posibil de cel specificat prin *diam*; dacă este utilizată versiunea cu “*” a comenzii, atunci LaTeX generează un cerc plin (cu interiorul înnegrit).

`\dashbox{dash_length} (width,height) [pos] {text}` generează un *box* dreptunghiular, cu conturul punctat, ce conține textul *text*; argumentul opțional *pos* specifică poziția în care apare textul (implicit centrat pe orizontală și verticală) și care poate fi selectată prin una sau două din valorile:

- *t* (*top*) – plasare în partea de sus a dreptunghiului;
- *b* (*bottom*) – plasare în partea de jos a dreptunghiului;
- *l* (*left*) – plasare la stânga;
- *r* (*right*) – plasare la dreapta;
- *c* (*center*) – centrat.

Comanda are argumentul suplimentar *dash_length*, care specifică lungimea unei “liniuțe”, iar aspectul desenului este optim când parametrii *width* și *height* sunt multipli de *dash_length*.

`\frame{obj}` creează un contur dreptunghiular în jurul obiectului specificat de argumentul *obj*; punctul de referință este colțul stânga-jos și nu se lasă spațiu suplimentar între contur și obiect.

`\framebox (width,height) [pos] {text}` este similară cu comanda `\dashbox`, cu excepția faptului că generează un contur cu linie continuă, plasat în exterior, în jurul cutiei pe care o creează.

`\line (x_slope,y_slope) {length}` este comanda ce generează o linie cu lungimea *length* și panta (*slope*) x_slope/y_slope , unde x_slope și y_slope pot avea valori întregi cuprinse între -6 și 6.

`\makebox (width,height) [pos] {text}` este similară cu comanda `\dashbox`, cu excepția faptului că marginile *box*-ului sunt invizibile.

`\oval (width,height) [portion]` produce un oval (“dreptunghi cu vârfurile rotunjite”); argumentul opțional *portion* permite selectarea, pentru tipărire, doar a unei zone a ovalului, specificată prin una din valorile:

- t (partea de sus),
- b (partea de jos),
- r (partea din dreapta) sau
- l (partea din stânga).

Trebuie menționat că “vârfurile” ovalului sunt realizate cu cercuri cu raza maximă de 20 pt, astfel încât un oval “mare” va avea, mai degrabă, aspectul unui *box* cu colțurile rotunjite.

`\shortstack [pos] {obj1\obj2\...}` generează o stivă de obiecte; ultimul obiect este la baza stivei iar valorile argumentului opțional *pos* pot fi:

- r (*right*) – plasarea obiectului la dreapta stivei
- l (*left*) – plasarea obiectului la stânga stivei;
- c (*center*) – plasarea obiectului în centrul stivei (implicit).

`\vector (x_slope,y_slope) {length}` desenează o linie cu o săgeată, de lungimea specificată prin *length* și cu panta x_slope/y_slope .

În continuare sunt date câteva exemple de utilizare a cadrului `picture`.

Exemplu 2.26

	<pre> \documentclass{report} \begin{document} \begin{figure} \begin{picture}(100,100) \setlength{\unitlength}{1pt} \put(120,20){\dashbox{.5} (50,50)[b]{TEXT}} \end{picture} \caption{Un prim exemplu} \end{figure} \end{document} </pre>
--	---

*Exemplul 2.27*

	<pre> \documentclass{report} \begin{document} \begin{picture}(100,100) \setlength{\unitlength}{1pt} \put(1.4,5){\vector(1,2){20}} \put(1.4,2.6){\line(3,-1){50}} \end{picture} \end{document} </pre>
--	--

*Exemplul 2.28*

	<pre> \documentclass{report} \begin{document} \begin{picture}(100,100) \setlength{\unitlength}{1pt} \put(20,0){\circle{20}} \put(20,0){\vector(0,1){10}} \put(50,0){\circle*{10}} \end{picture} \end{document} </pre>
--	---



Exemplul 2.29

	<pre> \documentclass{report} \begin{document} \begin{picture}(100,100)\setlength{\unitlength}{1pt} \put(10,70){\shortstack{Acesta \\ este \\ un obiect}} \put(10,50){\oval(50,30)} \end{picture} \end{document} </pre>
--	--

□

Exemplul 2.30

	<pre> \documentclass{report} \begin{document} \begin{picture}(100,100) \setlength{\unitlength}{1pt} \multiput(3,3)(5,-3){3}{\oval(40,20)} \end{picture} \end{document} </pre>
--	---

□

Exemplul 2.31

	<pre> \documentclass{report} \begin{document} \begin{picture}(100,100) \setlength{\unitlength}{0.2cm} \put(5,4){\line(0,-1){2}} \put(6,2){\oval(2,2)[b1]} \put(6,1){\vector(1,0){6}} \end{picture} \end{document} </pre>
--	--

□

Observație. Pentru a realiza transformări geometrice și a include fișiere de tip .eps, care conțin grafică realizată în alte medii, se folosește pachetul graphics. Includerea graficii se realizează cu comanda `\includegraphics` (care suportă și forma cu “*”). Detalii privind pachetul graphics se găsesc în [Lamport 1994].

□

2.3.15 Cadrul quotation

```
\begin{quotation}
  text
\end{quotation}
```

Cadrul `quotation` se utilizează pentru evidențierea unui citat mai lung, compus din mai multe paragrafe. Marginile cadrului `quotation` sunt indentate la stânga și la dreapta. Textul este aliniat (*justified*) la aceste margini și paragrafele sunt indentate. O linie goală în text generează un nou paragraf.

Exemplul 2.32

```
\documentclass{report}
\begin{document}
Iată un câțeva sfaturi.
\begin{quotation}
Cadrele pentru realizarea citatelor pot fi folosite și în
alte scopuri. Este bine să citești cu atenție
manualul.

Multe probleme pot fi rezolvate prin folosirea cadrelor din
LaTeX. Pentru aceasta ai nevoie de experiență.
\end{quotation}
Acestea au fost unele din sfaturile ce vor fi necesare. Restul
le vei găsi în paginile acestei cărți sau în
alte lucrări.
\end{document}
```



2.3.16 Cadrul quote

```
\begin{quote}  
  text  
\end{quote}
```

Cadrul `quote` este similar cadrului `quotation`, dar este destinat evidențierii unor citate mai scurte, ce cuprind doar un singur paragraf sau unor secvențe de citate scurte, separate prin linii goale. Marginile din stânga și din dreapta sunt indentate, iar textul este aliniat la ambele margini.

Exemplul 2.33

```
\documentclass{report}  
\begin{document}  
Motto-ul  
\begin{quote}  
"\TeX\ este un sistem de culegere de texte, destinat cre\u arii  
unor c\u ar\c ti frumoase -  
\c si, \^\i n special, a celor care con\c tin mult\u a  
matematic\u a."  
  
\emph{- D.E Knuth, The \TeX book (1983)-}  
\end{quote}  
este un adev\u ar recunoscut, din ce \^\i n ce mai mult, de  
editori din \^\i ntreaga lume.  
\end{document}
```



2.3.17 Cadrul `tabbing`

```
\begin{tabbing}
  text \= text \= text \\
  a doua linie \> \> ultimul text \\
  .
  .
\end{tabbing}
```

Cadrul `tabbing` permite alinierea textului pe coloane și poate fi utilizat doar în modul paragraf (secțiunea 2.6). El oferă un mecanism pentru setarea și folosirea tabulatorilor, similar cu cel de la o mașină de scris. Este adecvat situațiilor în care lățimea fiecărei coloane este constantă și cunoscută dinainte. Spre deosebire de cadrul `tabular`, acest cadru poate trece de la pagină la alta. Opririle `tab`-ului sunt numerotate 0, 1, 2, etc. Oprirea cu numărul i a `tab`-ului este setată, dacă i se asociază o poziție pe orizontală în pagină.

În interiorul unui cadru `tabbing` pot fi folosite următoarele comenzi:

- `\=` setează un `tab` în poziția curentă;
- `\>` salt orizontal la următorul `tab`;
- `\<` decrementează valoarea numărului curent de opriri ale `tab`-ului, permițând inserarea unui text la stânga marginii locale, fără a o modifica; poate fi folosită doar la început de linie, unde anulează (pe linia respectivă) efectul unei comenzi `\+` precedente ;
- `\+` mută marginea din stânga a liniei următoare cu un `tab`, iar mai multe astfel de comenzi au efect cumulativ; inițial, tabulatorul marginii din stânga este setat la 0;
- `\-` decrementează cu 1 tabulatorul marginii din stânga a liniei următoare (care trebuie să fie pozitiv), anulând efectul unei comenzi `\+`; mai multe astfel de comenzi au efect cumulativ;
- `\'` mută textul din coloana curentă, situat înaintea comenzii, în coloana precedentă și îl aliniază la dreapta;
- `\‘` produce alinierea la dreapta tabelului a textului ce urmează după comandă; după această comandă, nu mai sunt acceptate comenzile `\>`, `\=`, `\'` sau `\‘` ;

`\kill` nu tipărește rândul curent;

`\pushtabs` salvează poziția curentă a tuturor tabulatorilor ce pot fi restaurați cu comanda `\poptabs`;

`\poptabs` restaurează pozițiile tabulatorilor salvați la ultima comandă `\pushtab`;

`\a`, `\a'`, `\a'` produc accentele pe care le-ar genera, în mod normal, respectiv comenzile `\=`, `\'`, `\'` (și care, aici, sunt redefinite).

Exemplul 2.34

	<pre> \documentclass{report} \begin{document} \begin{tabbing} function \= fact(n : integer) : integer;\ \> begin \= \+ \ \> if \= n \$>\$ 1 then \+ \ fact := n * fact(n-1) \- \ else \+ \ fact := 1; \- \- \ end;\ \end{tabbing} \end{document} </pre>
--	--

□

Exemplul 2.35

	<pre> \documentclass{report} \begin{document} \begin{tabbing} O coloan\u a \=\^i ngust\u a.\ Acest r\^ and este prea lung. \>///////////////////////////////// \end{tabbing} \end{document} </pre>
--	---

□

Exemplul 2.36

	<pre> \documentclass{report} \begin{document} \begin{tabbing} Coloana veche 1 \=Coloana veche 2\\ Col.1\>Col.2\\ Col. nou\u a 1 \=Col. nou\u a 2\\ Col.1\>Col.2 \end{tabbing} \end{document} </pre>
--	---

□

Exemplul 2.37

	<pre> \documentclass{report} \begin{document} \begin{tabbing} Eminescu\=Creang\u a\=\kill Mihai \>Ion \>Ion Luca \\ Eminescu \>Creang\u a \>Caragiale \end{tabbing} \end{document} </pre>
--	---

□

2.3.18 Cadrul table

```

\begin{table} [placement]
  corpul tabelului
\caption{titlul tabelului}
\end{table}

```

Tabelele, la fel ca și figurile, sunt obiecte al căror conținut trebuie să apară doar pe o singură pagină și ele pot fi deplasate (*float*) în text, - de exemplu, la începutul paginii următoare - astfel încât să nu existe pagini umplute parțial.

Argumentul opțional *placement* determină locul în care LaTeX va încerca să plaseze tabelul. Aceste poziții sunt specificate cu ajutorul unor caractere cu următoarea semnificație :

- `h` (*here*) – în locul în care apare cadrul `table` în text;
- `t` (*top*) – în capătul de sus al unei pagini de text;
- `b` (*bottom*) – la baza unei pagini de text;
- `p` (*page of floats*) – pe o pagină separată ce nu conține text, ci doar obiecte deplasate.

Poziția default este `tpb`. Corpul tabelului este compus din text, comenzi LaTeX etc. Includerea comenzii `\caption` permite etichetarea și scrierea unei legende asociate tabelului.

Un exemplu de utilizare a cadrului `table` va fi dat în secțiunea ce tratează cadrul `tabular`.

2.3.19 Cadrul `tabular`

```
\begin{tabular} [pos] {cols}
  linia 1 :col.1 & col.2   ... & col.n\\
  .
  .
  linia m :col.1 & col.2   ... & col.n\\
\end{tabular}
```

sau

```
\begin{tabular*} {width} [pos] {cols}
  linia 1 :col.1 & col.2   ... & col.n\\
  .
  .
  linia m :col.1 & col.2   ... & col.n\\
\end{tabular*}
```

Aceste cadre produc un *box*, constând dintr-o secvență de linii de elemente, aliniate vertical în coloane. Argumentele obligatorii și respectiv opționale sunt descrise în continuare.

- *width* – specifică lățimea cadrului `tabular*`.
- *pos* – specifică poziția pe verticală, *default* înseamnă alinierea centrat în cadru, iar valorile pot fi:
 - `t` – aliniere în capătul de sus al liniei;
 - `b` – aliniere în capătul de jos al liniei.

- *cols* – specifică formatul coloanei și constă dintr-o secvență alcătuită cu simbolurile de mai jos.

l – o coloană de elemente aliniate la stânga

r – o coloană de elemente aliniate la dreapta

c – o coloană de elemente centrate

| – separator de coloane, reprezentat de o linie verticală mărginită de spații

@{*text*} – separator de coloane ce specifică o @-expresie. Se inserează textul *text* în fiecare linie, între coloanele între care apare @-expresie. O @-expresie suprimă spațiul dintre coloane, inserat în mod normal și orice spațiu dorit între textul propriu-zis și celelalte elemente trebuie să fie inclus în *text*. O comandă `\extracolsep{wd}`, într-o @-expresie, generează un spațiu suplimentar, de lățime *wd*, la stânga tuturor coloanelor următoare, până la următoarea comandă `\extracolsep`. Spre deosebire de spațiul uzual dintre coloane, acest spațiu suplimentar nu poate fi suprimat printr-o @-expresie. O comandă `\extracolsep` poate fi folosită doar în argumentul *cols*, într-o @-expresie.

p{*wd*} – generează o coloană, cu fiecare element plasat într-o cutie cu lățimea *wd*, ca și cum ar fi argumentul unei comenzi `\parbox[t]{wd}`. Cu toate acestea, secvența `\\` nu este permisă decât în următoarele situații:

- în interiorul unui cadru de tip `array`, `minipage` sau `tabular`,
- în interiorul unui `\parbox` explicit sau
- în sfera de acțiune a unei comenzi `\centering`, `\raggedright` sau `\raggedleft`. Ultimele două comenzi trebuie să apară între paranteze sau într-un cadru, atunci când sunt folosite într-un element coloană p.

*{*num*}{*cols*} – echivalent cu *num* copii ale coloanelor, unde *num* este orice întreg pozitiv iar *cols* este orice listă de specificatori de coloană, care pot conține și o @-expresie.

Fiecare linie este separată, de următoarea, prin comanda `\\` și constă dintr-o secvență de elemente, separate prin caracterul `&`. O linie trebuie să conțină același număr de elemente ca cel specificat de argumentul *cols*. Fiecare element este procesat ca și cum ar fi între paranteze, astfel încât câmpul de

acțiune al fiecărei declarații dintr-un element – i.e. efectul declarației - se limitează la acel element. Următoarele comenzi pot fi folosite în specificarea unui element al unei linii:

- `\multicolumn{n}{pos}{item}` – face elementul *item* să se extindă pe *n* coloane, în poziția *pos*, care poate conține una din valorile: *r* (*right*), *c* (*centred*), *l* (*left*) și una sau mai multe @-expresii și caractere |; această comandă fie începe o linie a tabelului, fie urmează imediat după caracterul &;
- `\vline` – când este utilizată în interiorul unui element *l*, *c* sau *r*, produce o linie verticală, pe toată înălțimea liniei respective de elemente ale tabelului; o comandă `\hfill` poate fi utilizată pentru a muta linia la marginea coloanei; comanda poate fi utilizată într-o @-expresie.

Comenzile descrise în continuare pot genera linii orizontale, între două linii succesive de elemente ale tabelului. Ele trebuie să apară fie înaintea primei linii de elemente, fie imediat după o comandă `\\`. O linie orizontală, după ultima linie de elemente, este produsă de comanda `\\`, urmată de una din comenzile descrise mai jos.

Comanda

`\cline{i-j}`

subliniază elementele tabelului, începând de la coloana *i* până la coloana *j*.

Comanda

`\hline`

plasată după o secvență `\\` sau la începutul cadrului, trasează o linie orizontală sub toate coloanele. Două comenzi `\hline` succesive lasă un spațiu între linii; delimitările generate de caracterul |, din argumentul *cols*, nu apar în acest spațiu.

Exemplul 2.38

	<pre> \documentclass{report} \begin{document} \begin{table}[htbp] \begin{center} \begin{tabular}{cc} \multicolumn{2}{c}{\emph{Some real values}}\ \\ \hline \$x=0.1\$&\$y=2.2\$ \\ \$r=10\$&\$z=100\$ \\ \end{tabular} \end{center} \caption{Exemple 38} \end{table} \end{document} </pre>
--	--

□

Exemplul 2.39

	<pre> \documentclass{report} \begin{document} \begin{tabular}{ r r@{--}l p{1.25in} }\hline \multicolumn{4}{ c }{\emph{Firma Paradis}}\ \\ \hline\hline &\multicolumn{2}{c }{Pre\c t}&\ \\ \cline{2-3} \multicolumn{1}{ c } {An}&\multicolumn{1}{r@{\,}\vline\,}}{\min}&\max &\multicolumn{1}{c }{Observa\c tii} \\ \hline 1971&97&245&An slab.\ \\ \hline 72&245&245&Iarn\ u a grea.\ \\ \hline 73&245&2001&Un an ciudat. Prognoz\ u a bun\ u a.\ \\ \hline \end{tabular} \end{document} </pre>
--	---

□

Exemplul 2.40

	<pre> \documentclass{report} \begin{document} \begin{tabular*}{40mm}{@{}r@{.}l% @{\extracolsep{\fill}}cr% @{\extracolsep{0pt}}:}l@{} \multicolumn{5}{c}{\underline{Tabel }}\\ 1&22&\text{obiect 1} &\text{\textit{clasa}}\text{^ ang} \text{\textit{u a}} \\ 50&7 &\text{obiectul 2} &\text{\textit{clasa}}\text{\textit{ro}} \text{\textit{c sie}} \\ 9&99&\text{obiecte} &\text{\textit{clase}}\text{\textit{diverse}} \end{tabular*} \end{document} </pre>
--	---



2.3.19.1 Observații importante privind cadrul `tabular` și cadrul `array`

Următoarele proprietăți ale cadrelor `tabular` și `array` (secțiunea 2.3.1), deși au fost menționate, sunt adesea ignorate de utilizatorii începători.

- Aceste cadre creează o cutie (*box*) (a se vedea și secțiunea 2.3.13).
- Cutia creată cu aceste comenzi poate avea un spațiu alb, înainte de prima coloană și respectiv după ultima coloană, spațiu ce poate fi eliminat cu o @-expresie.
- Orice declarație într-o linie a tabelului (vezi *linia 1*, ..., *linia m*) aparține unui element al tabelului, iar câmpul său de acțiune este elementul respectiv (i.e. efectul se exercită doar asupra elementului).
- O @-expresie, în argumentul *cols*, suprimă spațiul implicit dintre coloane.

2.3.19.2 Parametrii de stil privind cadrul `array` și cadrul `tabular`

Formatul cadrelor `array` și `tabular` poate fi respectiv controlat cu următoarele comenzi, reprezentând parametrii de stil privind lungimile:

- `\arraycolsep` – reprezintă jumătate din lățimea specificată, prin argumentul `width`, a spațiului orizontal implicit dintre coloane, într-un cadru `array`;
- `\tabcolsep` – reprezintă jumătate din lățimea specificată, prin argumentul `width`, a spațiului orizontal implicit dintre coloane, într-un cadru `tabular` sau `tabular*`;
- `\arrayrulewidth` – reprezintă lățimea unei linii create, cu ajutorul caracterului `|`, în argumentul `cols` sau cu una din comenzile `\hline`, `\cline` sau `\vline`;
- `\doublerulesep` – reprezintă lățimea spațiului dintre linii, creat cu două caractere `|` succesive, în argumentul `cols`, sau cu două comenzi `\hline` succesive;
- `\arraystretch` – controlează spațierea dintre liniile tabelului; spațiul normal este multiplicat cu numărul generat de `\arraystretch`, astfel încât modificarea sa, de la valoarea implicită 1 la 1.5, crește spațiul dintre linii de 1.5 ori, în parte; valoarea sa poate fi modificată, din nou, cu o comandă `\renewcommand` (subsecțiunea 2.2.4 și secțiunea 2.5)

Parametrii de stil de mai sus pot fi schimbați, oricând, în afara unui cadru `array`, respectiv `tabular`. Ei pot fi, de asemenea, schimbați local, în interiorul unui element, dar câmpul de acțiune al modificării respective trebuie delimitat prin paranteze de grupare sau printr-un cadru.

2.3.20 Cadrul `thebibliography` și programul BibTeX

2.3.20.1 Cadrul `thebibliography` și generarea manuală a bibliografiei

```
\begin{thebibliography}{widest_label}
\bibitem[label]{cite_key}
.
.
\end{thebibliography}
```

Cadrul `thebibliography` permite crearea manuală a unei liste de referințe bibliografice. În clasa `article`, lista de referințe este intitulată “References”, iar în clasa `report`, “Bibliography”. Parametrul `widest_label` specifică un text care, tipărit, este aproximativ tot atât de lat cât elementul de lățime maximă, generat de comenzile `\bibitem`.

Comanda

```
\bibitem[label]{cite_key}
```

generează o înregistrare cu eticheta `label`. Dacă argumentul `label` lipsește, se generează un număr cu rolul etichetei, utilizând contorul `enumi` (subsecțiunea 2.4.1). Parametrul `cite_key` este orice secvență de litere, numere sau semne de punctuație diferite de virgulă “,”. Această comandă scrie, în fișierul cu extensia `.aux`, o înregistrare conținând `cite_key` și eticheta `label` a elementului. Când acest fișier `*.aux` este citit, la comanda `\begin{document}`, eticheta elementului este asociată cu `cite_key`, astfel încât, prin comanda `\cite`, referința respectivă generează eticheta `label` asociată.

Comanda

```
\cite[text]{key_list}
```

generează citarea, în `text`, a referințelor asociate cu etichetele din lista `key_list`, prin înregistrări din fișierul `*.aux`, citit la comanda `\begin{document}`. Argumentul `key_list` este o listă de etichete de citare. Argumentul opțional `text` va apărea după citare, i.e. comanda `\cite[p. 2]{knuth}` va genera “[Knuth, p. 2]”.

Comanda

```
\nocite{key_list}
```

nu generează nici un text, dar scrie *key_list*, care este o listă de una sau mai multe citări, în fișierul **.aux*.

Exemplul 2.41

```
\documentclass{report}
\begin{document}
Leslie Lamport \cite{LL:LaT} shows that\par
.....
\begin{thebibliography}{99}
\bibitem{LL:LaT} Leslie Lamport.
  {\em (1994) }\LaTeX\ A Document Preparation System. User's
  Guide and Reference Manual. Addison-Wesley Publ. Co.
\end{thebibliography}
\end{document}
```

□

Observație. De remarcat că, în clasa *report*, bibliografia este tipărită pe o pagină separată, după textul documentului propriu-zis, pe când la documentele de tip *article* referințele apar la sfârșitul documentului, fără a se genera o nouă pagină.

□

Exemplul 2.42

```

\documentclass{article}
\begin{document}
Leslie Lamport \cite{LL:LaT} shows that...\par.....
\begin{thebibliography}{Alexander 95}
\bibitem[Alexander 95]{Al:Mat}John Alexander. {\em (1995) }
Mathematical Analysis. Ed. MIR, Moscow.
\bibitem[Lamport 94]{LL:LaT} Leslie Lamport.{\em (1994) }
\LaTeX\ A Document Preparation System. User's Guide and
Reference Manual. Addison-Wesley Publ. Co.
\end{thebibliography}
\end{document}

```

În acest exemplu, parametrul *widest_label* este Alexander 95.



Observație. Atenție, pentru rezolvarea *referințelor încrucișate*, fiecare fișier *.tex, din exemplele 2.41 și respectiv 2.42, a fost procesat, în LaTeX, de *două ori*; reamintim că, la prima trecere, se marchează referințele și abia la a doua trecere acestea sunt alocate.



Exemplul următor arată cum se poate redacta textul din exemplul 2.42 în limba română, prin redefinirea numelor secțiunilor (conform subsecțiunii 2.2.4).

Exemplul 2.43

```

\documentclass{article}
\renewcommand{\refname}{Bibliografie}
\begin{document}
Leslie Lamport \cite{LL:LaT} arat\u a c\u a ...\par
.....
\begin{thebibliography}{Alexander 95}
\bibitem[Alexander 95]{Al:Mat}
John Alexander. {\em (1995) }
Mathematical Analysis. Ed. MIR, Moscow.
\bibitem[Lamport 94]{LL:LaT}
Leslie Lamport.
{\em (1994) } \LaTeX\ A Document Preparation System. User's
Guide and Reference Manual. Addison-Wesley.
\end{thebibliography}
\end{document}

```

□

2.3.20.2 Programul BibTeX și generarea automată a bibliografiei

Dacă, pentru întreținerea listei bibliografice, utilizați programul BibTeX, creat de Oren Patashnik (recomandat, în special, în cazul unei bibliografii bogate, care conține mai mult de câteva titluri), atunci nu se folosește cadrul thebibliography. În schimb, se introduc următoarele linii:

```

\bibliographystyle{style}
\bibliography{bibfile}

```


unde *style* se referă la fișierul *style.bst*, care definește felul cum vor arăta citatele. Stilurile standard, distribuite de BibTeX, sunt următoarele:

alpha – sortare în ordine alfabetică, cu etichete constituite din numele autorului și anul apariției publicației;

plain – sortare în ordine alfabetică, cu etichete numerice;

unsrt – similar cu *plain*, dar elementele apar în ordinea citării, fără sortare;

abbrv – similar cu *plain*, dar etichetele sunt mult mai compacte.

În plus, pot fi activate numeroase alte fișiere *style* pentru BibTeX, conform cerințelor diverselor edituri și conform *layout*-urilor specifice pentru diverse reviste.

Argumentul comenzii `\bibliography` se referă la fișierul *bibfile.bib*, care ar trebui să conțină baza Dvs. de date, în format BibTeX. Detalii privind organizarea bazei de date se găsesc în [Lamport 1994] și în fișierul *btxdoc.dvi*, ce însoțește documentația MiKTeX (disponibilă la adresa www.stat.uiowa.edu/resources/help/win95/latex.html) care conține și lucrarea BibTeXing, scrisă de Oren Patashnik [Patashnik 1988]. Câteva elemente din această lucrare sunt prezentate, succint, în Anexa B.

2.3.21 Cadrul `titlepage`

```
\begin{titlepage}
  text
\end{titlepage}
```

Cadrul `titlepage` creează o pagină de titlu, i.e. o pagină nenumerotată și fără *header*, iar pagina imediat următoare devine prima pagină, din punctul de vedere al numerotării. Formatarea paginii de titlu este lăsată, în întregime, la latitudinea autorului. Adesea, este folosită una din comenzile `\today` (pentru generarea datei curente) sau `\date` (pentru specificarea unei anumite date sau a celei curente). Pentru a produce o pagină de titlu standard, se folosește comanda `\maketitle` (subsubsecțiunea 2.2.5.2).

Sunt utile, de asemenea, următoarele cadre și comenzi: cadrul `center`, descris în subsecțiunea 2.3.2, comenzile de schimbare de stil precum și cele de schimbare a dimensiunii fontului, descrise în secțiunea 2.6.

2.3.22 Cadrul `verbatim` și comanda `\verb`

```
\begin{verbatim}
  text
\end{verbatim}
```

Cadrul `verbatim` generează un paragraf, în care LaTeX tipărește exact ceea ce se introduce de la tastatură. Cu alte cuvinte, LaTeX devine o “mașină de scris”, care procesează spațiile (*blank*-urile) și comenzile *enter* de la tastatură.

Exemplul 2.44

```
\documentclass{report}
\begin{document}
Urmează un un text scris  $\wedge$  în cadrul \texttt{verbatim}.
\begin{verbatim}
Text scris  $\wedge$  în cadrul \texttt{verbatim}
\end{verbatim}
\end{document}
```

□

Comanda `\verb`, cu sintaxa

```
\verb char literal_text char ,
```

suportă și forma cu “*”, având sintaxa

```
\verb*char literal_text char .
```

literal_text semnifică textul tipărit, în stilul *typewriter* (`\texttt`, secțiunea 2.6), exact cum este introdus de la tastatură, inclusiv caracterele speciale și spațiile. *char* este un caracter diferit de literă (de exemplu + sau |), ce delimitează, în cadrul paragrafului curent, textul *literal_text*. Se admite să nu fie nici un *blank* între *char* și `\verb`, respectiv `\verb*`. În forma cu “*”, spațiile sunt tipărite ca un caracter special.

Exemplul 2.45

	<pre>\documentclass{report} \begin{document} Se poate scrie:\\ \verb+\$x=y\$+ sau \verb?\$x = y\$?. \end{document}</pre>
--	--

□

Exemplul 2.46

	<pre>\documentclass{report} \begin{document} Se poate scrie:\\\verb*+\$x=y\$+ sau \verb*?\$x = y\$?. \end{document}</pre>
--	---

□

2.3.23 Cadrul verse

```
\begin{verse}
text
\end{verse}
```

Cadrul `verse` este destinat tipăririi de versuri, deși se poate folosi și în alte scopuri. Marginile textului sunt indentate, la stânga și la dreapta. Versurile dintr-o strofă sunt separate cu secvența `\\`, iar strofele sunt separate prin una sau mai multe linii goale.

Exemplul 2.47

	<pre>\documentclass{report} \begin{document} ^In 1876, Eminescu scria: \begin{verse} Cu g^andiri \c si cu imagini\\ ^Innegrit-am multe pagini ... Nu urma\c ti g^andirei mele,\\ C\u aci noianu-i de gre\c sele,...\\ \end{verse} \end{document}</pre>
--	---

□

2.4 VARIABILE INTERNE LaTeX

2.4.1 Contori

Pentru numerotare, în LaTeX se asociază, în mod automat, un contor. Numele contorului este același cu numele cadrului sau al comenzii care generează numărul (cu excepția celor fără \). În continuare, este prezentată lista contorilor utilizați, pentru controlul numerotării, în categoriile standard de documente LaTeX.

part	paragraf	figure	enumi
chapter	subparagraf	table	enumii
section	page	footnote	enumiii
subsection	equation	mpfootnote	enumiv
subsection			

Contorii `enumi` - `enumiv` sunt folosiți în cadrul enumerate încuibat: `enumi` pentru primul nivel (exterior), `enumii` pentru următorul, etc.(subsecțiunea 2.3.4). Contorul `mpfootnote` numără notele de subsol (*footnotes*), într-un cadru `minipage` (subsecțiunea 2.3.13). În plus față de cele de mai sus, un cadru creat utilizând comanda `\newtheorem` (secțiunea 2.5) are un contor cu același nume, cu excepția situației în care un argument opțional specifică numerotarea, similară cu cea din alt cadru.

Valoarea unui contor este un întreg, în general nenegativ. Numerele multiple sunt generate cu contori diferiți; de exemplu, cifrele “2” și “4” din “**Subsecțiunea 2.4**” sunt generate de contorii `section` și respectiv `subsection`. Valoarea unui contor este inițializată la zero și este incrementată, de către comenzile sau cadrele corespunzătoare. De exemplu, contorul `subsection` este incrementat, de comanda `\subsection`, înainte de generarea numărului subsecțiunii și este resetat, la zero, când contorul `section` este incrementat, astfel încât numărul subsecțiunii începe, într-o nouă secțiune, de la unu (a se vedea și exemplul 2.48).

Contorul `page` este folosit pentru numerotarea paginilor într-un document. Acesta diferă de alți contori, prin faptul că este incrementat *după*

generarea numărului paginii; de exemplu, comanda `\setcounter{page}{27}` face ca pagina curentă să capete numărul 27. Din acest motiv, contorul `page` este inițializat la unu în loc de zero. Stilul numerotării este definit prin comanda `\pagenumbering` (secțiunea 2.9).

În continuare, se descriu comenzile ce operează asupra contorilor:

`\addtocounter{ctr}{num}`

Incrementează contorul `ctr` cu valoarea specificată de argumentul `num`.

`\newcounter{newctr} [within]`

Definește un nou contor, cu numele `newctr`, inițializat la zero, cu valoarea numerică (returată de `\thenewctr`) afișată în stilul definit prin `\arabic{newctr}`. Este o declarație globală și nu poate fi folosită într-un fișier inclus, declarat cu comanda `\include`. Se utilizează doar în preambul (a se vedea și subsecțiunea 2.3.12).

`newctr` – este o secvență de caractere, diferită de numele unui contor deja existent.

`within` – specifică numele unui contor deja definit. Dacă acest argument este prezent, atunci `newctr` este resetat la zero, ori de câte ori contorul `within` este incrementat prin comenzile `\stepcounter` sau `\refstepcounter`.

`\refstepcounter{ctr}`

`\stepcounter{ctr}`

Comenzile incrementează valoarea contorului `ctr` cu unu, resetând valoarea oricărui contor numerotat “în interiorul” acestuia. De exemplu, contorul `subsection` este numerotat în interiorul contorului `section` care, pentru clasele de documente `report` sau `book`, este numerotat în interiorul contorului `chapter`. În plus, comanda `\refstepcounter` declară valoarea curentă returnată de `\ref` ca fiind textul generat de comanda `\thectr`.

`\setcounter{ctr}{num}`

Setează valoarea contorului `ctr` la `num`. Este o declarație globală.

`\value{ctr}`

Generează valoarea contorului `ctr`, care poate fi astfel folosită într-o expresie. Este utilizată, cu precădere, în argumentul `num` al comenzilor

`\setcounter` sau `\addcounter`. De exemplu, comanda `\setcounter{cont}{\value{page}}` setează contorul `cont` la valoarea curentă a contorului `page`.

Comanda `\usecounter{ctr}` este folosită în cadrul `list` (subsecțiunea 2.3.12).

Observație. Comenzile `\setcounter` și `\addcounter` afectează doar contorul specificat. De exemplu, schimbarea contorului `section` prin aceste comenzi nu afectează contorul `subsection`.

□

Exemplul 2.48

	<pre> \documentclass{report} \begin{document} \setcounter{footnote}{17} Deoarece\footnote{...} \addtocounter{footnote}{-2} un contor\footnote{...} este\\ incrementat \^i nainte de utilizare,\\ el trebuie setat la valoarea\\ cu unu mai mic\u a dec\^at num\u arul dorit. \end{document} </pre>
--	--

□

Următoarele comenzi tipăresc valoarea contorului `ctr`, în formatul indicat:

`\alph{ctr}` Litere latine mici (valoarea lui `ctr` trebuie să fie mai mică decât 27).

`\Alph{ctr}` Litere latine mari (valoarea lui `ctr` trebuie să fie mai mică decât 27).

`\arabic{ctr}` Cifre arabe.

`\roman{ctr}` Cifre romane mici.

`\Roman{ctr}` Cifre romane mari.

`\fnsymbol{ctr}` Generează simboluri pentru notă de subsol. Poate fi folosită doar în modul matematic. Valoarea argumentului `ctr` trebuie să fie mai mică decât 10.

`\thectr` Tipărește valoarea contorului *ctr*.

Pentru a schimba numerotarea secțiunilor și subsecțiunilor, se folosește comanda `\renewcommand` (secțiunea 2.5). De exemplu, comenzile

```
\renewcommand{\thesection}{Roman{section}}
\renewcommand{\thesubsection}{\thesection-{\Alph{subsection}}}
```

asigură tipărirea secțiunii 2, cu cifra II și respectiv a subsecțiunii 2.4, sub forma II-D.

2.4.2 Lungimi și designul unei pagini

În această accepțiune, prin lungime se înțelege măsura unei distanțe. În LaTeX, o lungime explicită este exprimată, într-un anumit sistem dimensional, printr-un număr zecimal cu semn (+ implicit sau – explicit), adică

<parte întreagă>.<parte zecimală><unitate_de_măsură>,

unde *unitate_de_măsură* poate fi unul din simbolurile listate în continuare:

cm Centimetri.

em Un em este egal cu lățimea literei *M* în fontul curent.

ex Un ex este egal cu înălțimea literei *x* în fontul curent.

in Inch.

pc Picas (1pc = 12pt).

pt Points (1in = 72.27pt).

mm Milimetri.

O altă modalitate de specificare a distanțelor este realizată prin comenzi de tip lungime, care au sau returnează o valoare reprezentând o lungime. De exemplu, valoarea comenzii `\parindent` specifică lățimea indentării, la începutul fiecărui paragraf, i.e. distanța, de la marginea din stânga a paragrafului, la care începe prima linie a acestuia.

Lungimile pot fi *rigide*, (de exemplu `1.5em` sau `\parindent`) sau *elastice*. Acestea din urmă pot varia, iar spațiul specificat de o astfel de lungime se poate lărgi sau îngusta, conform cerințelor. De exemplu, spațiile dintre cuvinte variază într-un paragraf *justified*, astfel încât fiecare rând să aibă aceeași lungime.

Următoarele comenzi fac parte din categoria lungimi și permit definirea parametrilor de stil ai unui text, i.e. designul acestuia:

`\parindent` Lățimea indentării la început de paragraf. Este setată la zero într-un *parbox*. Valoarea sa poate fi schimbată oriunde.

`\textheight` Înălțimea textului în pagină, i.e. a corpului paginii (fig. 2.5).

`\textwidth` Lățimea textului în pagină (fig. 2.5). Valoarea sa poate fi schimbată doar în preambul.

`\parskip` Distanța, pe verticală, dintre două paragrafe consecutive. Uzual, această distanță este zero (cu excepția documentelor din clasa `letter`). Este o lungime *elastică*, astfel încât se poate extinde pentru a adăuga spațiu vertical, între paragrafe, când acționează declarația `\flushbottom`, descrisă în această subsecțiune. Valoarea sa poate fi schimbată, oriunde, în fișierul sursă.

`\baselineskip` Distanța normală, pe verticală, de la baza unui rând la baza rândului următor, în același paragraf. Depinde de mărimea fontului curent (secțiunea 2.6). Dacă rândul conține obiecte “înalte”, atunci această distanță poate fi mai mare.

Observație. Numărul de rânduri, pe o pagină ce conține doar un singur paragraf, este raportul dintre valoarea lungimii definite de `\textheight` și valoarea definită de `\baselineskip`.

□

Comanda `\baselinestretch` nu este o lungime. Ea returnează un număr zecimal, cu valoarea implicită egală cu unu și care poate fi modificată prin comanda `\renewcommand` (secțiunea 2.5).

Următoarele două comenzi definesc lungimi elastice.

`\fill` Definește o lungime elastică având, în mod natural, valoarea zero și posibilitatea de a se extinde la o valoare pozitivă arbitrară. Dacă această comandă este multiplicată cu un număr zecimal, ea definește o lungime rigidă; de exemplu `1\fill` sau `0.7\fill` sunt lungimi rigide, ambele cu valoarea `0in`.

`\stretch{dec_num}` Definește o lungime elastică, având, în mod natural, valoarea zero și posibilitatea de a se extinde de *dec_num* ori valoarea definită de `\fill`, unde argumentul *dec_num* este un număr zecimal cu semn (i.e. un semn opțional, urmat de un șir de cifre și un punct zecimal opțional).

Următoarele comenzi permit modificarea lungimilor.

`\newlength{\cda}` Declară parametrul `\cda` ca fiind o comandă de tip lungime, unde `cda` este numele unei comenzi ce nu a fost încă definită. Valoarea definită de `\cda` este inițializată la zero.

`\setlength{\cda}{lun}` Setează valoarea comenzii de tip lungime `\cda` la valoarea argumentului `lun`.

`\addtolength{\cda}{len}` Setează valoarea comenzii de tip lungime `\cda` la valoarea sa curentă plus valoarea argumentului `lun`.

`\settowidth{\cda}{text}`

`\settoheight{\cda}{text}`

`\settodepth{\cda}{text}` Setează valoarea comenzii de tip lungime `\cda` la lățimea, înălțimea și respectiv adâncimea argumenului `text`.

O pagină tipărită are trei părți: *header*-ul, corpul textului și *footer*-ul (fig. 2.5). Dimensiunile acestora sunt determinate de parametrii de stil. Stilul paginii specifică conținutul *header*-ului și al *footer*-ului. Paginile din stânga și din dreapta pot avea parametrii, respectiv, diferiți. În stilul față-verso, paginile pare sunt pagini din stânga, respectiv cele impare sunt pagini din dreapta. La tipărirea pe o singură față, toate paginile sunt considerate pe dreapta.

Următoarele comenzi permit definirea stilului unei pagini.

`\pagestyle{style}` Este o declarație ce specifică stilul paginii curente.

Opțiunile *style* standard sunt următoarele:

`plain` - *Header*-ul este gol, *footer*-ul conține doar numărul paginii.

Este stilul implicit al paginii.

`empty` - Atât *header*-ul cât și *footer*-ul sunt goale.

`headings` - *Header*-ul conține informația determinată de categoria documentului (uzual, este un *header* asociat unității secționale), precum și numărul paginii; *footer*-ul este gol.

`myheadings` - La fel ca `headings`, cu excepția faptului că informația din *header* (fără numărul paginii) este specificată prin comenzile `\markboth` și `\markright`, descrise în această subsecțiune.

`\thispagestyle` La fel ca `\pagestyle`, cu excepția faptului că se aplică doar paginii curente (cea care tocmai urmează să fie “decupată” de către TeX). Este o declarație globală.

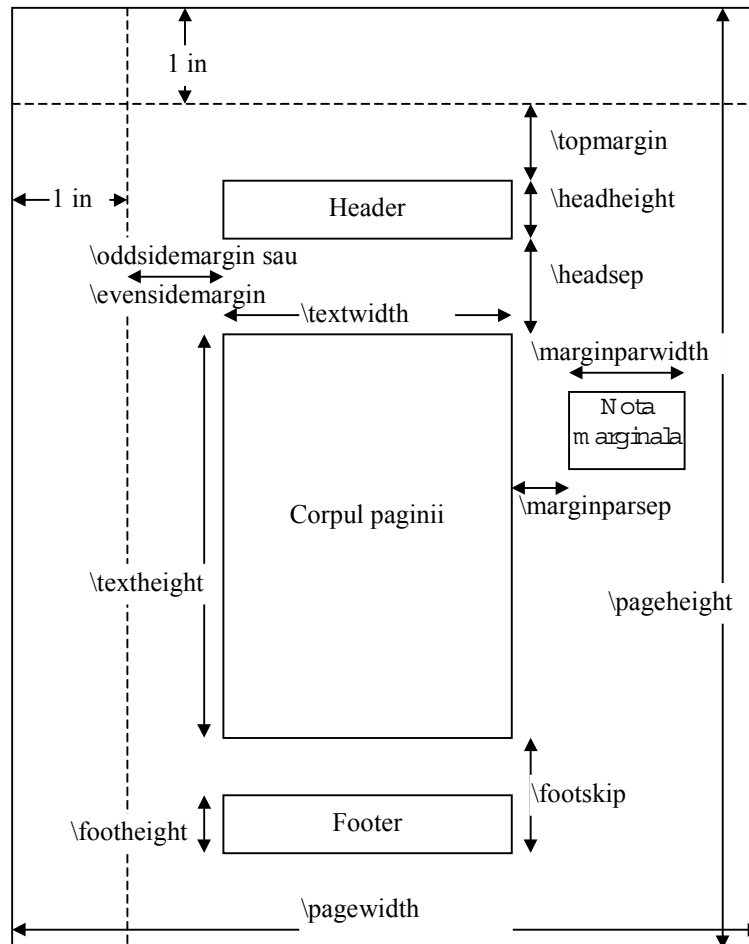


Fig. 2.5 - Parametrii de stil ai paginii; la paginile pare, notele marginale sunt în stânga corpului textului.

`\markright{right_head}`

`\markboth{left_head}{right_head}` Aceste comenzi specifică următoarea informație conținută în *header*, pentru stilurile de pagină `headings` și `myheadings`:

- pentru pagina din stânga, *header*-ul este definit de argumentul *left_head* al ultimei comenzi `\markboth`, ce apare înainte de sfârșitul paginii;
- pentru pagina din dreapta, *header*-ul este definit de argumentul *right_head* al ultimei comenzi `\markright` sau `\markboth` din

pagina, sau, dacă nu există nici una, al ultimei comenzi corespunzătoare, ce apare înainte de începutul paginii.

În stilul `headings`, comenzile de secționare setează *header*-ele paginilor cu ajutorul comenzilor `\markboth` și `\markright` astfel:

Stilul de tipărire	Comanda	Clasa documentului	
		<code>book, report</code>	<code>article</code>
față-verso	<code>\markboth</code> [*]	<code>\chapter</code>	<code>\section</code>
	<code>\markright</code>	<code>\section</code>	<code>\subsection</code>
o singură față	<code>\markright</code>	<code>\chapter</code>	<code>\section</code>

*) specifică un *header* vid pe pagina din dreapta

În mod implicit, prima pagină are *header*-ul vid. Dacă nu se dorește acest lucru, atunci se poate genera o primă pagină goală, cu ajutorul cadrului `titlepage`.

`\pagenumbering{num_style}` Specifică stilul numerelor paginilor și setează valoarea contorului `page` la unu (subsecțiunea 2.4.1). Este o declarație globală. Valorile posibile ale argumentului *num_style* sunt :

`arabic` Cifre arabe.
`roman` Cifre romane mici (i, ii, etc.).
`Roman` Cifre romane mari (I, II, etc.).
`alph` Litere latine mici.
`Alph` Litere latine mari.

Comanda `\pagenumbering` redefinește valoarea returnată de `\thepage` ca fiind `\num_style{page}` (secțiunea 2.4.1).

`\twocolumn[text]` Începe o nouă pagină, executând `\clearpage` și începe tipărirea pe două coloane. Dacă argumentul *text* este prezent, atunci este tipărit într-un *parbox* pe două coloane, la începutul noii pagini.

`\onecolumn` Începe o nouă pagină, executând comanda `\clearpage` și începe tipărirea pe o singură coloană.

`\flushbottom` Înălțimea textului, pe fiecare pagină, este aceeași, prin adăugarea, când este necesar, a unui spațiu vertical, de dimensiune corespunzătoare, între paragrafe.

`\raggedbottom` Anulează efectul comenzii `\flushbottom`. În LaTeX, se consideră implicit `\raggedbottom`, cu excepția cazului când acționează opțiunea `twoside` a comenzii `\documentclass` (subsecțiunea 2.2.5.1)

Majoritatea parametrilor de stil ce permit designul unei pagini sunt prezentați în fig. 2.5, în care dreptunghiul exterior reprezintă marginile coalei. Acești parametri sunt toți *lungimi rigide* și pot fi schimbați doar în preambulul fișierului cu extensia `.tex` (subsecțiunea 1.1). Paginile impare folosesc `\oddsidemargin`, respectiv cele pare comanda `\evensidemargin`. Nu este figurat parametrul `\topskip`, reprezentând minimul distanței măsurate de la marginea superioară a corpului figurii la punctul de referință al primei linii de text; el acționează ca o comandă `\baselineskip`.

2.4.3 Parametrii de stil pentru ecuații matematice

Atunci când se dorește modificarea stilului în care sunt scrise ecuațiile, trebuie modificate așa-numiții parametri de stil asociați acestor obiecte. În general, aceasta nu este o acțiune recomandată, deoarece valorile implicite ale parametrilor de stil au fost stabilite, în LaTeX, pe baza anumitor reguli tipografice (reguli ce au izorât din cerințele editurilor din SUA).

Parametrii de stil ai ecuațiilor sunt comenzi de tip lungimi (secțiunea 2.4.2). Iată, în continuare, descrierea lor.

`\jot` Definește dimensiunea unui spațiu vertical suplimentar, adăugat între două linii consecutive, într-un cadru `eqnarray` sau `eqnarray*` (subsecțiunea 2.3.5).

`\mathindent` Definește indentarea unei formule, măsurată de la marginea din stânga a paragrafului, când se folosește opțiunea `fleqn` pentru categorii (clase) de documente (subsecțiunea 2.2.5.1).

`\abovedisplayskip` Definește dimensiunea spațiului suplimentar, adăugat deasupra unei formule lungi – cu excepția cazului în care se folosește opțiunea `fleqn` pentru categorii de documente, caz în care se

consideră lungimea definită de comanda `\topsep`. Lungimea definită este elastică.

`\belowdisplayskip` Definește dimensiunea spațiului suplimentar, adăugat sub o formulă lungă – cu excepția cazului în care se folosește opțiunea `fleqn` pentru categorii de documente, caz în care se consideră lungimea definită de comanda `\topsep`. Lungimea definită este elastică.

`\abovedisplayshortskip` Definește spațiul suplimentar, adăugat deasupra unei formule scurte – cu excepția cazului în care se folosește opțiunea `fleqn` pentru categorii de documente, caz în care se consideră valoarea lungimii definite de comanda `\topsep`. Începutul unei formule scurte este situat la dreapta punctului unde sfârșește linia precedentă. Lungimea definită este elastică.

`\belowdisplayskip` Definește spațiul suplimentar adăugat la stânga unei formule scurte – cu excepția cazului în care se folosește opțiunea `fleqn` pentru categorii de documente, situație în care se consideră valoarea lungimii definite de comanda `\topsep`. Lungimea definită este elastică.

Formatul sau stilul de tipărire al numărului ecuației este controlat de comanda `\theequation`, ce poate fi eventual redefinită. De exemplu, pentru categoria de documente `book`, stilul de tipărire `\theequation` este predefinit ca fiind `{\thechapter.\arabic{equation}}`; în consecință, numărul formulei a 3-a din capitolul 2 este tipărit (2.3). Acest stil nu mai este valabil dacă se editează o formulă într-un document din categoria `report` (a se vedea și exemplul 2.20 din subsecțiunea 2.3.6). Stilul de numerotare poate fi, însă, redefinit, eventual doar local (i.e. doar pentru un grup de formule din document), cu ajutorul comenzii `\renewcommand` (subsecțiunea 2.5.1).

Observație. În cazul în care se dorește ca stilul de tipărire al numărului unei formule să fie modificat, la comanda

```
\renewcommand{\theequation}{noua definiție a stilului de numerotare}
```

doar pentru acea formulă din document, i.e. local, atunci întregul grup de comenzi asociate formulei, începând de la `\renewcommand` până la `\end{equation}`, trebuie inclus între acolade `{ și }`. În afara acestor acolade, definiția comenzii `\theequation` este cea implicită pentru clasa documentului respectiv.

□

În continuare este prezentat un exemplu de modificare locală a stilului de numerotare a formulelor.

Exemplul 2.49

```

\documentclass{report}
\begin{document}
Iată un exemplu de formulă scrisă cu parametrii de
stil nemodificați.
Definiția transformatei Laplace a semnalului  $f(t)$  este:
\begin{equation}
F(s)=\int_0^{\infty}f(t)e^{-st}dt
\end{equation}
Acum se modifică stilul de tipărire al contorului
\verb|equation|.
Numărul curent al capitolului este \arabic{chapter}.
{\renewcommand{\theequation}{\thechapter.\arabic{equation}a}
  \begin{equation}
    \label{ecu}x_i=x_{i-1}+x_{i-2}
  \end{equation}
  \addtocounter{equation}{-1}
\renewcommand{\theequation}{\thechapter.\arabic{equation}b}
  \begin{equation}
    \label{ecu+1}y=x_i \quad i>10
  \end{equation}
}
Acestea au fost trei exemple simple de formule.
\end{document}

```



2.4.4 Parametrii de stil privind cadrul `table` și cadrul `figure`

Tabelele sau figurile sunt obiecte *flotante* (i.e. pot fi deplasate), create cu cadrele `table` (subsecțiunea 2.3.18) și respectiv `figure` (subsecțiunea 2.3.7). O pagină *flotantă* conține numai astfel de obiecte și nu conține text. Așezarea în pagină a tabelelor sau figurilor se face, în LaTeX, respectând, în mod optim, reguli tipografice [Lamport 1994]. Iată câteva dintre acestea:

- Obiectul (tabel sau figură) este tipărit în locul cel mai apropiat posibil, față de punctul unde a fost generat, care respectă regulile de mai jos, cu mențiunea faptului că poziția, specificată prin valoarea `h` (*here*) a argumentului *placement*, este prioritară față de cea specificată prin valoarea `t` (*top*).
- Obiectul nu va fi tipărit pe o pagină ce precede locul din document unde este definit cadrul (`table` sau `figure`) asociat.
- O figură nu va fi tipărită înaintea unei figuri definite anterior, iar un tabel nu va fi tipărit înaintea unui tabel precedent definit.
- Obiectul poate să apară doar în poziția permisă de valoarea argumentului opțional *placement* (din definiția cadrelor `figure`, respectiv `tabular`) sau, în lipsa acestuia, de valoarea implicită `tpb`.
- O figură sau un tabel nu poate depăși o pagină.

Modificarea, în preambulul documentului, a următorilor parametri de stil se aplică începând cu prima pagină. Schimbările, intervenite ulterior, se aplică începând cu pagina imediat următoare locului unde au fost specificate.

`topnumber` Definește un contor a cărui valoare reprezintă numărul maxim de obiecte flotante, permise la începutul unei pagini de text.

`\topfraction` Definește procentul maxim, din spațiul unei pagini, care poate fi ocupat, în partea superioară a acesteia, de obiecte flotante. Astfel, valoarea 0.25 specifică faptul că primul sfert al paginii poate fi alocat obiectelor flotante. Poate fi modificat prin comanda `\renewcommand`.

`bottomnumber` Are aceeași semnificație ca `topnumber`, cu excepția faptului că obiectele sunt plasate în partea inferioară a paginii.

`\bottomfraction` Are aceeași semnificație ca `\topfraction`, cu excepția faptului că obiectele flotante sunt plasate în partea inferioară a paginii.

`totalnumber` Definește un contor, a cărui valoare reprezintă numărul maxim de obiecte flotante care pot să apară, pe o pagină ce conține text, oricare ar fi poziția acestora.

`\textfraction` Definește procentul minim, din spațiul unei pagini, dedicat textului. Restul procentului, de $1 - \text{\textfraction}$, trebuie ocupat de obiecte flotante. Valoarea sa poate fi redefinită la comanda `\renewcommand`.

`\floatpagefraction` Definește procentul minim, din spațiul unei pagini, care trebuie ocupat de obiecte flotante, limitând astfel spațiul gol permis pe o pagină flotantă. Valoarea sa poate fi redefinită la comanda `\renewcommand`.

`dbltopnumber` Analogul contorului `topnumber` pentru obiecte flotante, așezate pe două coloane, pe o pagină cu stilul `twocolumn`.

`\dbltopfraction` Analogul comenzii `\topfraction` pentru obiecte flotante, așezate pe două coloane, pe o pagină cu stilul `twocolumn`.

`\dblfloatpagefraction` Analogul comenzii `\floatpagefraction` pentru o pagină flotantă, cu stilul `twocolumn`.

`\floatsep` Definește spațiul vertical, adăugat între obiectele flotante, așezate în partea superioară sau inferioară a unei pagini de text.

`\textfloatsep` Definește spațiul vertical, adăugat între obiectele flotante, așezate în partea superioară sau inferioară a unei pagini de text și textul respectiv.

`\intextsep` Definește spațiul vertical, adăugat deasupra și dedesubtul unui obiect flotant, așezat în centrul paginii de text, cu argumentul opțional `h` (*here*). Valoarea sa este o lungime elastică.

`\dblfloatsep` Analogul comenzii `\floatsep` pentru obiecte flotante, de lățime dublă decât a unei coloane, situate pe o pagină cu stilul `twocolumn`. Valoarea sa este o lungime elastică.

`\dbltextfloatsep` Analogul comenzii `\textfloatsep` pentru obiecte flotante, de lățime dublă decât a unei coloane, situate pe o pagină cu stilul `twocolumn`. Valoarea sa este o lungime elastică.

2.5 MACRODEFINIȚII

De regulă, compilatoarele lucrează cu macrodefiniții numite, pe scurt, macrouri. Un macrou reprezintă, în esență, o secvență de comenzi ce va fi înlocuită, în timpul procesării, cu definiția macroului respectiv. Din acest punct de vedere, LaTeX poate fi considerat un pachet de macrouri.

Definiția unui macrou poate consta dintr-un text obișnuit împreună, eventual, cu alte macrouri și/sau primitive TeX. Primitivele TeX sunt secvențe de control, recunoscute de TeX și care *nu* pot fi înlocuite cu alte comenzi.

2.5.1 Definirea comenzilor

Un macrou poate fi definit sau, respectiv, redefinit prin comenzile:

```
\newcommand{\cda} [arg] [opt] {def}
\renewcommand{\cda} [arg] [opt] {def}
```

Parametrii comenzilor au următoarele semnificații:

- cda* – Numele unei comenzi. În cazul comenzii `\newcommand`, trebuie să *nu* fie numele unei comenzi deja definite și *nu* trebuie să înceapă cu `end`; în cazul comenzii `\renewcommand`, trebuie să fie numele unei comenzi deja definite.
- arg* – Un întreg între 1 și 9, reprezentând numărul de argumente al comenzii `\cda`. Implicit, comanda definită/redefinită nu are nici un argument.
- opt* – Dacă acest argument este prezent, atunci primul argument din totalul de *arg* ale comenzii `\cda` este opțional și are valoarea implicită *opt*. Dacă acest argument lipsește, atunci toate celelalte argumente ale comenzii `\cda` sunt obligatorii.
- def* – Reprezintă textul înlocuit ori de câte ori se întâlnește comanda `\cda`; un parametru, de forma `#n` în `\cda`, este înlocuit, când se face această substituție, cu textul asociat parametrului cu numărul *n*. *def* poate conține comenzi de definire a comenzilor sau cadrelor, doar dacă toate comenzile sau cadrele astfel definite nu au argumente.

În exemplul următor se arată cum se poate crea, în modul matematic, o comandă simplă, care apare de mai multe ori în textul documentului.

Exemplul 2.50

	<pre> \documentclass{article} \newcommand{\gn}{\\$\Gamma_{i}\$} \begin{document} Fie \gn\ valoarea asociat\u a m\u arimii \$\\$, \par unde \gn\ este o norm\u a euclidian\u a. \end{document} </pre>
--	--

Observație. După comanda `\gn`, s-a dat comanda `\` (i.e. *backslash* urmat de caracterul *blank*), deoarece TeX ignoră caracterul *blank* plasat după numele comenzii definite de `\newcommand`.

□

Comanda `\Gamma`, din exemplul 2.50, poate fi folosită doar în modul matematic și, de aceea, s-a folosit perechea de caractere `$` și `$`, pentru delimitarea ei ca argument *def* al comenzii `\newcommand`. Pe de altă parte, comanda `\gn` nu poate fi folosită în modul matematic deoarece, la întâlnirea ei, primul caracter `$` determină TeX să iasă din modul matematic și să încerce să execute `\Gamma` în modul paragraf. Dacă `\newcommand` definește o comandă ce referă un obiect matematic (funcție, expresie, formulă etc.) și care va trebui repetată și în modul matematic, atunci, în argumentul *def*, se folosește comanda

$$\backslashensuremath{expresie}$$

care asigură faptul că argumentul *expresie* (reprezentând un șir de caractere și comenzi ce descriu obiectul matematic în speță) este tipărit în modul matematic, indiferent de modul curent (subsecțiunile 2.6 și 2.7).

Exemplul 2.51

	<pre> \documentclass{article} \newcommand{\func}[2] {\ensuremath{ \mathit{var}(\#1;\#2) }} \begin{document} Deoarece \func{5x}{y} \c si \func{2x-1}{y}\par reprezint\u a valori aproximativ egale ale... \end{document} </pre>
--	--

□

Observație. Comanda `\mathit` este valabilă doar în modul matematic și permite scrierea argumentului său cu litere italice (secțiunea 2.7). Al doilea argument opțional al comenzii `\newcommand` specifică faptul că `\func` este o comandă cu două argumente. Parametrii #1 și #2 sunt înlocuiți cu primul și, respectiv, al doilea argument al comenzii `\func`, în momentul în care aceasta este apelată în textul documentului. □

O comandă poate fi definită și în funcție de altă comandă, după cum reiese din exemplul următor.

Exemplul 2.52

	<pre> \documentclass{article} \newcommand{\func}[2] {\ensuremath {\mathit{var}(\#1;\#2)}} \newcommand{\util} {\func{2x}{0}} \begin{document} Valoarea \func{5x}{y} \c si\par valoarea \util\ ... \end{document} </pre>
--	--

Parantezele rotunde, (și), din argumentul *def*, nu delimitează efectul unei declarații. Pentru a limita câmpul de acțiune al unei declarații, conținute în argumentul *def*, trebuie adăugate, în definiția comenzii, paranteze suplimentare, de tip acoladă { și }. Altfel, acțiunea unei astfel de comenzi se poate extinde, în mod nedorit, și în afara argumentului respectiv. □

Exemplul 2.53

	<pre> \documentclass{report} \newcommand{\bun}[3] {\#1\$\{(\#2);(\#3)}\$} \newcommand{\rau}[3] {\#1\$\#2;\#3\$} \begin{document} Expresia \bun{\em func}{x}{4} este tip\u arit\u a \par corect, dar dup\u a \rau{\em finc}{x}{4} totul \par apare cu caractere italice, ca efect \par al comenzii \verb \em din primul\par argument al comenzii \verb \rau . \end{document} </pre>
--	---

□

Observație. Comanda `\newcommand` poate fi folosită și pentru abrevierea unor comenzi, frecvent folosite într-un document. De exemplu, dacă se editează multe ecuații și se dorește să se evite repetarea comenzilor `\begin{equation}` și `\end{equation}`, de fiecare dată când se definește un cadru `equation`, atunci se pot folosi comenzile

```
\newcommand{\bee}{\begin{equation}}
\newcommand{\ene}{\end{equation}}
```

Acestea definesc grupul `\bee ... \ene` ca fiind echivalent cu comenzile LaTeX

```
\begin{equation}...\end{equation}
```

Dacă se folosesc foarte des ecuații numerotate, atunci, în loc să se scrie, pentru fiecare ecuație, grupul `\begin{equation}\label{...} formula \end{equation}`, se poate utiliza comanda

```
\newcommand{\ec}[2]{\begin{equation}\label{#1}#2\end{equation}}
```

în care primul parametru specifică eticheta, iar cel de-al doilea formula [Pusztai 1994].

□

În subsecțiunea **2.2.4**, s-a exemplificat folosirea comenzii `\renewcommand`. Programe scurte, ce utilizează `\renewcommand`, sunt descrise în exemplul 2.43 (subsecțiunea **2.3.20**) și în exemplul 2.49 (subsecțiunea **2.4.3**).

2.5.2 Definirea cadrelor

Un cadru poate fi definit sau, respectiv, redefinit, utilizând comenzile:

```
\newenvironment{num}[arg][opt]{begdef}{enddef}
\renewenvironment{num}[arg][opt]{begdef}{enddef}
```

Parametrii acestor comenzi au următoarele semnificații:

num - Reprezintă numele cadrului; poate fi orice secvență de litere, numere sau caracterul `*`, care nu începe cu `end`. În cazul comenzii `\newenvironment`, trebuie să nu mai existe nici un alt cadru cu numele *num* și nici o altă comandă `\num`. În cazul comenzii `\renewenvironment`, cadrul cu numele *num* trebuie să fie deja definit.

arg - Este un întreg cuprins între 1 și 9, reprezentând numărul de argumente ale cadrului nou definit. Implicit, cadrul nou definit/redefinit nu are nici un argument.

- opt* – Dacă acest argument este prezent, atunci primul din cele *arg* argumente ale cadrului este opțional și are valoarea implicită *opt*. Dacă acest argument lipsește, atunci toate argumentele cadrului sunt obligatorii.
- begdef* – Reprezintă textul folosit, în fișierul sursă, pentru a înlocui instrucțiunea `\begin{num}`; un parametru de forma #n în *begdef* este înlocuit cu textul corespunzător argumentului cu numărul n al comenzii `\begin{num}`, în momentul în care are loc această substituție.
- enddef* – Reprezintă textul folosit, în fișierul sursă, pentru a înlocui instrucțiunea `\end{num}`. Nu este permis să aibă parametri de tip argument.

În general, un nou cadru este definit folosind un cadru deja existent.

Exemplul 2.54

	<pre> \documentclass{report} \newenvironment{evident} {\begin{itemize}\em} {\end{itemize}} \begin{document} Acesta este un exemplu de cadru\par definit de utilizator. \begin{evident} \item Acest cadru, numit \verb evident ,\par genereaz\u a o list\u a de elemente\par scrise cu caractere italice. \item Cadrul \verb evident este scris\par folosind cadrul \verb itemize din\par \LaTeX\ \c si comanda \verb \em . \end{evident} Acum a luat sf\^ar\c sit lista generat\u a\par de cadrul \verb evident . \end{document} </pre>
--	---

Observație. Argumentele *begdef* și *enddef* pot conține comenzi de definire a comenzilor sau de redefinire a cadrelor, doar dacă toate comenzile sau cadrele pe care le acestea le definesc nu au, la rândul lor, argumente. Parantezele de delimitare ale unui cadru, care a fost definit sau redefinit cu `\newenvironment`, respectiv cu

`\renewenvironment`, nu delimitează și câmpul de acțiune al unei declarații conținute în argument. Acest aspect a fost evidențiat și în cazul comenzii `\newcommand` (subsecțiunea 2.5.1). Comanda `\renewenvironment` poate fi folosită, doar dacă utilizatorul cunoaște foarte bine cadrul pe care dorește să-l redefinească.

□

Exemplul 2.55

```
\documentclass{report}
\newenvironment{descr}[1]{\begin{quote}\textit{#1}:}
  {\end{quote}}
\begin{document}
Observați cum o nouă structură a logicii a -\^i în acest
exemplu, o descriere etichetată a unui singur element -
poate fi definită a \^i în funcție de un cadru deja
existent.
\begin{descr}{Ceai}
Această descriere a ceaiului a fost generată de cadrul
\verb|descr|.
\end{descr}
Urmează alt text.
\end{document}
```

Observație Comanda `\textit{txt}` are ca efect tipărirea argumentului *txt* cu caractere italice.

□

2.5.3 Definirea cadrelor de tip teoremă

Comanda `\newtheorem` permite definirea unui cadru în care se pot scrie teoreme, leme, propoziții etc. și poate avea una din următoarele sintaxe:

```
\newtheorem{env_name}{caption} [within]
\newtheorem{env_name} [numbered_like] {caption}
```

Parametrii comenzilor au, respectiv, următoarele semnificații:

env_name – Este un șir de litere, reprezentând numele teoremei, (lemei, axiomei, etc.) ce trebuie scrise și care trebuie să difere de numele unui alt cadru sau contor.

caption – Textul tipărit la începutul cadrului, chiar înainte de număr, de exemplu “Teoremă” sau “Lemă”.

within – Numele unui contor deja definit, asociat, în general, unei unități secționale (capitol, secțiune etc.) și care permite resetarea contorului teoremei, în cadrul unității secționale respective. Dacă acest argument este prezent, atunci comanda `\theenv_name` este definită prin

$$\text{\thewithin.\arabic{env_name}},$$

iar contorul *env_name* va fi resetat de una din comenzile `\stepcounter{within}` sau `\refstepcounter{within}` (subsecțiunea 2.4.1). Dacă argumentul *within* lipsește, atunci `\theenv_name` este definită prin `\arabic{env_name}`.

numbered_like – Numele unui cadru de tip teoremă, deja definit. Dacă acest argument este prezent, atunci *env_name* va fi numerotat în aceeași secvență (utilizându-se același contor) ca și cadrul *numbered_like* și va declara valoarea `\ref` curentă, ca fiind textul generat de comanda `\thenumbered_like`.

Cu excepția cazului în care apare argumentul opțional *numbered_like*, comanda `\newtheorem` creează un contor numit *env_name*, iar cadrul declară valoarea curentă a `\ref` ca fiind textul generat de `\theenv_name`.

Exemple 2.56

```
\documentclass{report}
\begin{document}
\newtheorem{guess}{Propozi\c tia}
Dup\ u a cum se \c stie, dou\ u a matrice echivalente au acela\c
si rang. Av\ ^and \ ^i n vedere acest lucru, din teorema
anterioar\ u a rezult\ u a imediat urm\ u atorul enun\c t.
\begin{guess}
Rangul unei matrice este egal cu num\ u arul valorilor sale
singulare nenule.
\end{guess}
\end{document}
```



Prezența argumentului opțional final al comenzii `\newtheorem` determină numerotarea cadrului, în interiorul unității secționale respective. □

Exemplul 2.57

```
\documentclass{report}
\renewcommand{\chaptername}{Capitolul}
\newtheorem{guess}{Propozi\c tia}[chapter]
\begin{document}
\chapter{Rangul unei matrice}\label{unu}
Aceasta este prima propozi\c tie din capitolul \ref{unu}:
\begin{guess}
Rangul unei matrice este egal cu num\c arul valorilor sale
singulare nenule.
\end{guess}
\end{document}
```



□

Argumentul opțional *numbered_like* se folosește, dacă se dorește ca structuri distincte, de tip teoremă, să fie numerotate secvențial, cu același contor.

Exemplul 2.58

	<pre> \documentclass{report} \newtheorem{teor1}{Teorema} \newtheorem{teor2}[teor1] {Axioma} \begin{document} \begin{teor1} Unele teoreme sunt numerotate. \end{teor1} \begin{teor2} To\c ti oamenii sunt muritori. \end{teor2} \end{document} </pre>
--	--

□

Uneori, este indicat să se precizeze autorul teoremei sau un alt nume, sub care rezultatul respectiv este celebru.

Exemplul 2.59

	<pre> \documentclass{article} \newtheorem{teor1}{Teorema} \begin{document} \begin{teor1}[Schur] Fie perechea de\par matrice reale p\u atrate (A,B), ... \end{teor1} \end{document} </pre>
--	---

□

Observație. În general, este recomandat ca declarația `\newtheorem` să figureze în preambulul fișierului sursă.

□

Observație. În sintaxa comenzilor `\newcommand`, `\newenvironment` și `\newtheorem` este interzisă folosirea spațiilor între argumente. De exemplu, corectă este sintaxa `\newtheorem{teor}{Teorema}` și nu `\newtheorem{teor} {Teorema}`.

□

2.5.4 Pachetul `ifthen`

Pachetul `ifthen` pune la dispoziție comenzile `\ifthenelse` și `\whiledo`, pentru scrierea de programe ce conțin teste și bucle. Pachetul `ifthen` este încărcat la comanda `\usepackage{ifthen}`. În continuare, sunt descrise sintaxele și modurile de utilizare ale comenzilor din acest pachet.

```
\ifthenelse{test}{then_txt}{else_txt}
```

Este o instrucțiune condițională, iar argumentul *test* este o expresie logică care, evaluată în LaTeX, poate avea valoarea *adevărat* sau *fals*. Dacă argumentul *test* are valoarea *adevărat*, atunci se procesează argumentul *then_txt*; dacă *test* are valoarea *fals*, se procesează *else_txt*. Argumentul *test* poate fi exprimat prin următoarele expresii:

`num1 op num2` Este o relație numerică, unde *op* poate fi unul din următoarele trei caractere: `>` `=` `<`, iar *num₁* și *num₂* sunt numere. De exemplu, `\value{page}>17` este evaluată la valoarea *adevărat*, dacă și numai dacă valoarea curentă a contorului `page` este mai mare ca 17.

`\equal{str1}{str2}` Este evaluată la valoarea *adevărat*, dacă și numai dacă TeX consideră *str₁* și *str₂* ca fiind egale. TeX poate considera două argumente (reprezentând comenzi) ca fiind diferite, chiar dacă au drept rezultat tipărirea aceluiași text. De exemplu `\today` și `May 1, 2001` nu sunt egale, chiar dacă prin aplicarea lor rezultă textul: `May 1, 2001`. TeX va considera argumentele *str₁* și *str₂* ca fiind egale dacă, prin înlocuirea fiecărei comenzi din fiecare argument cu definiția sa, ele sunt identice.

`\lengthtest{len1 op len2}` Evaluează relația între lungimile specificate de argumentele *len₁* și *len₂*; *op* poate fi unul din următoarele trei caractere: `>` `=` `<`. De exemplu, `\length{\parindent<1cm}` are valoarea *adevărat*, dacă și numai dacă valoarea curentă a declarației `\parindent` este mai mică decât 1cm. În procesul de evaluare, o lungime elastică este înlocuită cu valoarea sa naturală (subsecțiunea 2.4.2).

`\isodd{num}` Este evaluată la valoarea *adevărat*, dacă și numai dacă numărul specificat de argumentul *num* este impar (*odd*). Este folosită pentru a genera texte diferite, în funcție de paritatea paginilor. Totuși, comanda `\isodd{\value{page}}` nu lucrează corect, deoarece contorul `page` poate avea, de exemplu, valoarea curentă 42, chiar dacă textul procesat se extinde pe pagina cu numărul 43. Pentru a rezolva această problemă, se folosesc comenzile `\label{key}` și `\isodd{\pageref{key}}`. În momentul când se procesează *header*-ul sau *footer*-ul unei pagini, contorul `page` are chiar valoarea presupusă de utilizator,

`\boolean{nam}` Este evaluată la valoarea curentă a registrului logic *nam*, unde *nam* poate fi o secvență de litere. Acest registru trebuie definit de comanda `\newboolean{name}`. Valoarea sa este setată, prin comanda `\setboolean{name}{bool}`, unde parametrul *bool* este fie *true* (*adevărat*), fie *false* (*fals*).

Expresiile complexe, reprezentate de argumentul *test*, pot fi construite din expresii mai simple, între care se aplică operatorii logici `\and`, `\or` și `\not`, cu parantezele de grupare specificate de caracterele `\(` și `\)`.

`\whiledo{test}{body}`

Procesează în mod repetat argumentul *body*, până ce argumentul logic *test* devine *fals*, unde *test* este descris la fel ca în cazul instrucțiunii `\ifthenelse`. Nu execută nimic dacă, inițial, *test* are valoarea *fals*.

Exemplul 2.60

	<pre> \documentclass{article} \usepackage{ifthen} \newcommand{\test}[1] {\ifthenelse{#1=1}{bun}{r\u au}} \begin{document} Variantele pot fi:\par om \test1 sau om \test2. \end{document} </pre>
--	---

□

2.5.5 Comenzi fragile

Comenzile LaTeX pot fi împărțite în categoriile: *fragile* sau *robuste*. Comenzile de schimbare de stil (sau font), descrise în secțiunea 2.6, precum și majoritatea comenzilor din modul matematic (secțiunea 2.7) sunt robuste. Orice comandă cu cel puțin un argument opțional sau care acceptă o formă cu “*” este fragilă.

Unele argumente ale unor comenzi se numesc *mobile*. O comandă fragilă care apare într-un argument mobil produce eroare, dacă nu este protejată prin comanda `\protect`. Această comandă de protejare se aplică doar comenzii pe care o precede; comenzile fragile care, la rândul lor, apar în argumentul comenzii respective, trebuie să fie și ele protejate cu alte comenzi `\protect`. Următoarele comenzi și cadre au argumente mobile:

- Comenzile cu un argument ce poate figura în cuprins, în lista de figuri sau în lista de tabele, respectiv asociate documentului: `\addcontentsline`, `\addtocontents`, `\caption`, precum și comenzile de secționare (subsecțiunea 2.2.5.3). Dacă un argument opțional este folosit la o comandă `\caption` sau la o comandă de secționare, atunci chiar acesta este argumentul mobil.
- Comenzile de tipărire la terminal: `\typeout` și `\typein`. Argumentul opțional al comenzii `\typein` nu este mobil.
- Comenzile utilizate la generarea *heading*-urilor: `\markboth` (ambele argumente sunt mobile) și `\markright` (subsecțiunea 2.4.2). Comenzile de secționare deja menționate intră, și ele, în această categorie.
- Cadrul `letter` (definit în categoria de documente `letter`).
- Comanda `\thanks`.
- Argumentul opțional al comenzii `\bibitem`.
- declarația `@`, într-un cadru `array` sau `tabular`. (Deși `@` nu este o comandă, totuși comenzile fragile dintr-o `@`-expresie trebuie protejate prin comanda `\protect`, ca și în cazul în care ar figura într-un argument mobil.)

Toate comenzile de tip lungimi (subsecțiunea 2.4.2) sunt robuste și nu trebuie precedate de o comandă `\protect`. Comanda `\protect` nu trebuie să apară într-un argument al unei comenzi `\setcounter` sau, respectiv, `\addtocounter`.

2.6 MODUL PARAGRAF

Când TeX procesează un text, el se află într-unul din următoarele moduri: modul paragraf, modul matematic sau modul “de la stânga la dreapta” (numit și modul LR, de la *Left-to-Right*). Modul normal în care procesează TeX este modul paragraf. În modul paragraf, TeX consideră intrările ca o secvență de cuvinte și propoziții ce trebuie secționată pe rânduri, paragrafe și pagini.

Modurile matematic și LR vor fi prezentate în secțiunile următoare.

2.6.1 Fonturi

Stilul de tipărire al unui text, setat prin alegerea tipului și mărimii fontului, este folosit pentru a indica structura logică a documentului.

Iată un exemplu privind diferite tipuri de fonturi disponibile în LaTeX.

Exemplul 2.61

	<pre> \documentclass{article} \begin{document} \textup{Stil Upright. De obicei, implicit.}\par \textit{Stil Italic. Pentru eviden\c tiere.}\par \textsl{Stil Slanted. U\c sor diferit de italic.}\par \textsc{Stil small caps. Utilizat mai rar.}\par \textmd{Medium series. Adesea implicit.}\par \textbf{Boldface series. Pentru titluri.}\par \textrm{Roman family. Adesea implicit.}\par \textsf{Familia sans serif.}\par \texttt{Familia typewriter.} \textnormal{Stil normal, acum Roman.} \end{document} </pre>
--	--

□

Fiecărei comenzi de setare a stilului de tipărire îi corespunde o declarație. Acestea sunt listate în tabelul următor.

Tabelul 2.1: Comenzi și declarații de setare a stilului fontului

<i>Comanda</i>	<i>Declarația</i>	<i>Comanda</i>	<i>Declarația</i>
<code>\textmd{text}</code>	<code>\mdseries</code>	<code>\textup{text}</code>	<code>\upshape</code>
<code>\textbf{text}</code>	<code>\bfseries</code>	<code>\textit{text}</code>	<code>\itshape</code>
<code>\textrm{text}</code>	<code>\rmfamily</code>	<code>\textsl{text}</code>	<code>\slshape</code>
<code>\textsf{text}</code>	<code>\sffamily</code>	<code>\textsc{text}</code>	<code>\scshape</code>
<code>\texttt{text}</code>	<code>\ttfamily</code>	<code>\textnormal{text}</code>	<code>\normalfont</code>

De exemplu, *text* se poate tipări în stilul bold fie la comanda `\textbf{text}`, fie la apelarea declarației `{\bfseries text}`.

Observație. Nici una din comenzile sau declarațiile din tabelul 2.1 nu pot fi folosite în modul matematic. Comenzile de schimbare a stilului de tipărire, în modul matematic, sunt descrise în secțiunea 2.7.

□

Mărimea implicită a fontului folosit la tipărire, în LaTeX, este 10pt, dar aceasta poate fi setată, după dorință, la 11pt sau 12 pt, conform opțiunii din comanda `\documentclass` (subsecțiunea 2.2.5.1)

Următoarele declarații permit schimbarea mărimii fontului în cadrul unui document, lăsând stilul acestuia nemodificat (stilul implicit este Roman.)

Tabelul 2.2 : Declarații de setare a mărimii fontului

<code>\tiny</code>	<code>\small</code>	<code>\large</code>	<code>\huge</code>
<code>\scriptsize</code>	<code>\normalsize</code>	<code>\Large</code>	<code>\Huge</code>
<code>\footnotesize</code>		<code>\LARGE</code>	

Observație. Nici una din aceste comenzi nu poate fi folosită în modul matematic. Toate sunt comenzi fragile.

□

Mărimea efectivă pe care o generează aceste declarații depinde de mărimea implicită a fontului cu care se tipărește textul documentului. În exemplele următoare, mărimea implicită este 10pt.

Declarația `\normalsize` produce mărimea implicită, comanda `\footnotesize` generează mărimea corespunzătoare scrierii notelor de subsol, iar comanda `\scriptsize` produce mărimea adecvată scrierii indicilor inferiori și superiori, în cazul formulelor tipărite cu `\normalsize`.

Exemplul 2.62

	<pre> \documentclass{article} \begin{document} {\tiny Exemplu}\par {\scriptsize Exemplu}\par {\footnotesize Exemplu}\par {\small Exemplu}\par {\normalsize Exemplu}\par {\large Exemplu}\par {\Large Exemplu}\par {\LARGE Exemplu}\par {\huge Exemplu}\par {\Huge Exemplu}\par \end{document} </pre>
--	--

Aceste declarații pot fi combinate cu declarațiile sau comenzile de selectare a fontului. □

Exemplul 2.63

	<pre> \documentclass{article} \begin{document} \sffamily un text scris\par \textit{cu litere din}\par \large ce ^\i n ce \Large mai\par mari \bfseries \c si mai \par \LARGE groase. \end{document} </pre>
--	--

□

2.6.2 Simboluri speciale și accente

În LaTeX, se pot genera simboluri speciale și accente ce nu sunt comune limbii engleze. Ele permit inserarea, într-un text în limba engleză, a unor texte scurte redactate în alte limbi. Toate comenzile prezentate în această secțiune sunt robuste.

În tabelul 2.3, sunt prezentate câteva simboluri care nu sunt comune limbii engleze și comenzile ce le produc; aceste simboluri pot fi generate atât în modul paragraf, cât și în modul LR.

Tabelul 2.3 : Simboluri speciale permise doar în modurile paragraf și LR

--

În tabelul 2.4 sunt prezentate comenzile ce generează șase semne speciale de punctuație.

Tabelul 2.4 : Simboluri speciale (permise în orice mod)

--

Alte simboluri speciale și comenzile ce le generează au fost prezentate la începutul secțiunii **2.2**.

În tabelul 2.5 sunt prezentate comenzile ce permit generarea literei o, cu diferite accente specifice altor limbi decât engleza. Aceste comenzi pot fi folosite doar în modul LR și în modul paragraf. Ele se pot aplica, în forma specificată, oricărei litere.

Tabelul 2.5 : Comenzi pentru generarea de accente aplicate literei o

--

Alte tipuri de accente, specifice limbii române, au fost prezentate în subsecțiunea **2.2.3**.

Observație. Comenzile `\i` și `\j` generează literele i, respectiv j fără punct deasupra. Reamintim că `\i` este folosită, pentru a construi litera î, la comanda `\^{\i}` sau `\^{\i}`.



2.6.3 Gestionarea rândurilor și a paginilor

În general, TeX realizează optim organizarea textului pe rânduri, paragrafe și pagini, astfel încât fiecare rând dintr-un paragraf (cu excepția, eventual, a ultimului) și respectiv fiecare pagină (cu excepția, eventual, a ultimei pagini) să fie pline. Uneori sunt, însă, necesare comenzi speciale în acest scop. Ca o recomandare generală, este bine ca formatarea textului să fie ultimul aspect tratat în realizarea unui document în LaTeX. Altfel spus, nu se începe rafinarea formatării, înainte să se fi introdus tot textul.

2.6.3.1 Gestionarea rândurilor

`\linebreak [num]`

`\nolinebreak [num]`

Comenzile `\linebreak` și `\nolinebreak` forțează și, respectiv, inhibă trecerea la un rând nou. Parametrul opțional *num* este un număr între 0 și 4. O valoare mai mare a lui *num* constituie o sugestie mai puternică pentru a trece, respectiv pentru a nu trece la rândul nou. Dacă *num* este absent, atunci comenzile acționează, respectiv, ca în cazul *num* = 4, adică trecerea la rândul nou este forțată, respectiv interzisă. Un mesaj `\hbox underfull` avertizează că o comandă `\linebreak` generează spații prea mari între cuvinte pe un rând. Ambele comenzi sunt fragile și pot fi folosite doar în modul paragraf.

`\ [len]`

`* [len]`

`\newline`

Efectul acestor comenzi este trecerea la un rând nou, fără a se lăsa margini egale la stânga și la dreapta rândului curent (i.e. rândul curent *nu* este *justified*). Argumentul opțional al comenzii `\` adaugă, deasupra noului rând, un spațiu vertical suplimentar de dimensiune *len*. Forma cu “*” a comenzii `\` inhibă trecerea la o pagină nouă, înainte de noul rând. Comanda `\newline` poate fi folosită doar în modul paragraf, într-un paragraf; ea generează un mesaj de avertizare `\hbox underfull`, dacă este utilizată la sfârșitul unui paragraf și, respectiv, un mesaj de eroare, dacă este utilizată între paragrafe. Comanda `\` se comportă în mod similar, când este utilizată în modul paragraf. Ambele comenzi sunt fragile.

`\|-`

Această comandă permite despărțirea în silabe: se trece la rândul următor și se inserează liniuța de despărțire la finele rândului curent. Este o comandă robustă.

`\hyphenation{cuvinte}`

Sunt declarate permise puncte de despărțire, unde *cuvinte* este o listă de cuvinte, separate prin spații, iar fiecare punct de despărțire, în lista *cuvinte*, este marcat prin caracterul -. Este o declarație globală și robustă.

`\sloppy`

`\fussy`

Aceste declarații controlează trecerea la un rând nou. Declarația `\fussy` este implicită și gestionează rândul, astfel încât previne spații prea mari între cuvinte. Declarația `\sloppy` face ca LaTeX să nu fie atât de rigid cu spațierea cuvintelor. `\sloppy` are același efect ca secvența de instrucțiuni

```
\begin{sloppypar} pare \end{sloppypar},
```

ce definește cadrul `sloppypar`. Argumentul *pare* reprezintă mai multe paragrafe pentru care este activă declarația `\sloppy`.

Trecerea, de la un paragraf la altul, se face lăsând un rând liber în text sau la comanda `\par`.

2.6.3.2 Gestionarea paginilor

`\pagebreak[num]`

`\nopagebreak[num]`

Comenzile `\pagebreak` și `\nopagebreak` forțează și, respectiv, inhibă trecerea la o nouă pagină, în stilul `onecolumn`. Parametrul opțional *num* este un număr între 0 și 4. O valoare mai mare a lui *num* constituie o sugestie mai puternică pentru a trece, respectiv pentru a nu trece la pagina nouă. Dacă *num* este absent, atunci comenzile acționează, respectiv, ca în cazul *num* = 4, adică trecerea la pagina nouă este forțată, respectiv interzisă. Dacă sunt utilizate într-un paragraf, aceste comenzi se aplică punctului imediat următor liniei pe care au fost apelate. Dacă este activă declarația `\flushbottom` (subsecțiunea

2.4.2), atunci se generează un mesaj de avertizare `\vbox underfull`, în cazul în care `\pagebreak` are drept efect prea puține rânduri pe pagină. Comanda `\nopagebreak` nu are efect, în cazul în care o altă comandă LaTeX a permis, în mod explicit, trecerea la pagină nouă în punctul specificat. Aceste comenzi nu au nici un efect în modul LR sau într-un *box*.

```
\enlargethispage{len}
\enlargethispage*{len}
```

Aceste comenzi măresc înălțimea paginii procesate, în mod curent, de LaTeX (i.e. `\textheight`) cu valoarea lungimii specificate de argumentul *len*; *len* reprezintă o lungime rigidă și poate avea valoare negativă. De exemplu, `\enlargethispage{\baselineskip}` va permite introducerea unei linii suplimentare în pagină (subsecțiunea **2.4.2**). Forma cu “*” a comenzii îngustează, cât mai mult posibil, spațiul vertical dintre rânduri, comprimând textul. Este însoțită, în general, de o comandă `\pagebreak`. Sunt comenzi fragile.

```
\newpage
\clearpage
\cleardoublepage
```

Aceste comenzi sunt plasate în finalul paragrafului și, respectiv, al paginii curente, în cazul paginilor cu textul pe o singură coloană (`onecolumn`). Orice spațiu vid în corpul paginii (subsecțiunea **2.4.2**) apare amplasat în partea inferioară a paginii, chiar dacă declarația `\flushbottom` este activă. Comenzile `\clearpage` și `\cleardoublepage` determină, în plus, și tipărirea tabelor și figurilor care au fost create și nu au fost încă așezate în pagini; tipărirea se poate realiza, dacă este necesar, pe una sau mai multe pagini flotante (ce conțin doar tabele sau figuri). În stilul de tipărire față-verso, `\cleardoublepage` transformă pagina următoare într-una cu număr impar, inserând, dacă este necesar, o pagină goală.

Dacă se folosește stilul `twocolumn`, atunci `\newpage` va avea ca efect terminarea coloanei curente, mai degrabă decât a paginii; comenzile `\clearpage` și `\cleardoublepage` termină pagina, producând, dacă este necesar, o coloană vidă. Aceste comenzi se utilizează doar în modul paragraf. Comenzile `\newpage` și `\clearpage` sunt robuste.

2.6.4 Spații

Unele spații, necesare la scrierea unui text, sunt predefinite și reprezintă parametri de stil (a se vedea și subsecțiunea 2.4.2). În continuare, sunt descrise comenzile ce permit modificarea spațiilor, măsurate pe verticală și pe orizontală, din structura unei pagini.

```
\hspace{len}
```

```
\hspace*{len}
```

Aceste comenzi au ca efect generarea unui spațiu, pe orizontală, de lățime *len*; parametrul *len* este o lungime exprimată într-o formă acceptată în LaTeX (subsubsecțiunea 2.4.2). Spațiul generat de `\hspace` este suprimat, dacă apare în punctul de trecere la un rând nou; în același caz, spațiul generat de `\hspace*` nu este suprimat. Ambele comenzi sunt robuste.

```
\vspace{len}
```

```
\vspace*{len}
```

Adaugă un spațiu, pe verticală, de înălțime *len*; parametrul *len* este o lungime exprimată într-o formă acceptată în LaTeX (subsubsecțiunea 2.4.2). Spațiul generat de `\vspace` este suprimat, dacă apare în punctul de trecere la o pagină nouă; în același caz, spațiul generat de `\vspace*` nu este suprimat. Ambele comenzi pot fi folosite doar în modul paragraf și sunt fragile.

```
\bigskip
```

```
\medskip
```

```
\smallskip
```

Sunt echivalente, respectiv, cu următoarele comenzi:

```
\vspace{\bigskipamount}      \vspace{\smallskip}
```

```
\vspace{\medskipamount},
```

unde comenzile `\bigskipamount`, `\medskipamount` și `\smallskipamount` sunt de tip lungimi și reprezintă parametri de stil. Aceste comenzi, generatoare de spații, pot fi folosite în definiții de cadre, pentru a asigura spații verticale standard. Sunt comenzi fragile.

```
\addvspace{len}
```

În mod normal, această comandă adaugă un spațiu, pe verticală, de înălțime *len*. Cu toate acestea, dacă în punctul respectiv a fost adăugat,

deja, un spațiu, printr-o comandă `\addspace` precedentă, atunci noua comandă nu va mai adăuga spațiu, decât până la completarea dimensiunii specificate de parametrul *len*. Este utilizată pentru adăugarea spațiilor suplimentare, pe verticală, situate deasupra și, respectiv dedesubtul celor mai multor cadre LaTeX care încep un nou paragraf. Poate fi folosită doar în modul paragraf, între paragrafe, i.e. imediat după o linie goală sau după comanda `\par`. Este o comandă fragilă.

`\hfill`

Este echivalentă cu `\hspace{\fill}` (comanda `\fill` a fost descrisă în subsecțiunea 2.4.2).

`\vfill`

Este echivalentă cu o linie goală, urmată de comanda `\vspace{\fill}`. Se utilizează doar în modul paragraf.

`\dotfill`

Acționează la fel ca `\fill` (subsecțiunea 2.4.2), dar produce puncte.

`\hrulefill`

Acționează la fel ca `\dotfill`, dar produce o linie orizontală.

`\noindent`

Suprimă alineatul (i.e. indentarea) la început de paragraf. Nu are nici un efect dacă este utilizată în interiorul unui paragraf.

`\par`

Sfârșit de paragraf (echivalentă cu o linie goală în text).

`\indent`

Introduce un spațiu, pe orizontală, a cărei lățime este egală cu lățimea indentării paragrafului. Este utilizată pentru a adăuga o indentare la începutul unui paragraf, în cazul în care, în rest, indentarea a fost suprimată.

Observație. Toate aceste comenzi pot primi, ca parametri, lungimi date în oricare din formele acceptate în LaTeX: se acceptă atât mărimi exprimate într-una din unitățile de măsură a lungimilor (subsecțiunea 2.4.2), cât și multipli sau submultipli ai unor comenzi de tip lungimi. De exemplu, `\vspace{0.5cm}` înseamnă adăugarea unui spațiu vertical de 0.5cm; `\vspace{2\parskip}` înseamnă adăugarea unui spațiu vertical egal cu dublul distanței reprezentate de `\parskip` (comanda `\parskip` este descrisă în subsecțiunea 2.4.2).

□

Exemplul 2.64

	<pre> \documentclass{article} \begin{document} Comanda \verb \hspace{1cm} las\u a un\par spa\c tiu orizontal \par de 1cm, deci \hspace{1cm} etc.\par Numele\hrulefill \par Prenumele\hrulefill\par Nr.\dotfill \par din \dotfill\par Urmeaz\u a spa\c tiu vertical de 1cm.\par \vspace{1cm} \noindent Paragraf f\u ar\u a alineat. \end{document} </pre>
--	---

**2.6.5 Note marginale**

Uneori, este necesar să se insereze, în text, comentarii, sub formă de note marginale. În acest scop, se poate utiliza comanda

```
\marginpar [left_text] {right_text},
```

care generează o notă marginală, având drept conținut textul specificat prin argumentul *right_text*, dacă nota este așezată la dreapta sau dacă nu există argument opțional, respectiv textul specificat prin *left_text*, altfel. Textul este tipărit într-un *parbox*.

Pentru stilul de tipărire față-verso și pe o singură coloană (opțiunile *twoside* și *onecolumn*), așezarea implicită a note marginale este pe marginea exterioară a paginii, la stânga textului, în cazul paginilor numerotate par, respectiv la dreapta textului, în cazul paginilor impare. În cazul stilului de tipărire pe o singură față și pe o singură coloană, așezarea implicită a notei marginale este pe marginea dreaptă a textului. Aceste amplasări implicite pot fi schimbate prin comenzile următoare:

```
\reversemarginpar – determină așezarea notelor marginale pe marginea opusă celei implicite;
```

```
\normalmarginpar – determină așezarea notelor pe marginea implicită.
```

La apariția unei note marginale într-un paragraf, așezarea sa este determinată de declarația activă în punctul unde începe linia goală ce încheie paragraful. În cazul tipăririi pe două coloane (opțiunea `twocolumn`), nota marginală apare, întotdeauna, la marginea cea mai apropiată de coloana ce conține nota, indiferent de aceste declarații.

De regulă, o notă marginală este așezată astfel încât linia sa de sus să fie la același nivel cu linia de text ce conține comanda `\marginpar`; dacă această comandă apare între paragrafe, atunci nota este, în general, aliniată cu ultima linie a paragrafului precedent comenzii. Oricum, dacă apare o suprapunere cu nota marginală precedentă, nota curentă este deplasată în josul paginii.

Parametrii de stil asociați notelor marginale sunt următoarele comenzi de tip lungimi (a se vedea și Fig.2.5, subsecțiunea **2.4.2**):

`\marginparwidth` – reprezintă lățimea *parbox*-ului ce conține nota marginală;

`\marginparsep` – reprezintă spațiul orizontal dintre marginea textului și nota marginală;

`\marginparpush` – specifică spațiul vertical minim permis între două note marginale succesive.

Exemplul 2.65

```
\documentclass[twoside]{article}
\begin{document}
\stepcounter{page}
Suntem la pagina \thepage.\par
Aici apare prima not\u a. \marginpar{\em Nota 1. A\c ti \^i
nv\u a\c tat la analiz\u a?}.\par
^\Incepe sesiunea (se creeaz\u a nota 2).\marginpar{\em Nota 2.
Ave\c ti grij\u a cum v\u a petrece\c ti timpul liber}.\par
Acum este mai interesant\u a via\c ta de student.
\end{document}
```



2.7 MODUL MATEMATIC

TeX este considerat în modul matematic, când generează formule matematice. În acest mod, literele sunt considerate simboluri matematice; de exemplu, ab este interpretat ca produsul lui a cu b . Spațiile dintre simboluri sunt ignorate.

2.7.1 Cadre și stiluri specifice modului matematic

O formulă matematică poate să apară, într-un document, fie pe un rând de text, fie pe un rând separat și dedicat. Pentru fiecare din cele două cazuri sunt disponibile mai multe comenzi echivalente, ce definesc, de fapt, cadre specifice modului matematic.

TeX poate folosi, doar în modul matematic, una din următoarele patru declarații de stil pentru editarea formulelor:

`\displaystyle` – stilul *display*, implicit pentru simboluri normale, la formulele scrise pe rânduri dedicate;

`\textstyle` – stilul *text*, implicit pentru simboluri normale, la formulele scrise pe un rând de text;

`\scriptstyle` – stilul *script*, implicit pentru indici inferiori și superiori de nivelul unu;

`\scriptscriptstyle` – stilul *scriptscript*, implicit pentru indici de niveluri superioare.

Cadrele asociate modului matematic sunt: `math`, `displaymath`, `equationarray` (subsecțiunea 2.3.5) și `equation` (subsecțiunea 2.3.6). În continuare, este descris modul de creare a cadrelor `math` și `displaymath`.

`$ formula $`

`\(formula)\)`

`\begin{math} formula \end{math}`

Aceste trei forme echivalente permit generarea unei formule așezate într-o linie de text uzual, prin editarea argumentului *formula* în stilul *text*. Pot fi utilizate în modul LR.

`\[formula \]`

`\begin{displaymath} formula \end{displaymath}`

Aceste forme echivalente permit generarea unei formule pe un rând separat, prin editarea argumentului *formula* în stilul *display*. Pot fi utilizate doar în modul paragraf (secțiunea 2.6). Formula apare centrată pe rând, cu excepția cazului în care opțiunea comenzii `\documentclass` este `fleqn` (subsecțiunea 2.2.5.1). Formula este nenumerotată

Observație. Cadrul `equation` este similar cu `displaymath`, cu excepția faptului că generează o formulă numerotată, utilizând contorul `equation`.

□

2.7.2 Comenzi pentru crearea unor structuri uzuale

Comenzile frecvent folosite la scrierea formulelor matematice sunt următoarele:

`_{sub}` - Tipărește argumentul *sub* ca indice inferior.

`^{sup}` - Tipărește argumentul *sup* ca indice superior.

`^` - Tipărește simbolul prim ($\hat{}$).

`\frac{numar}{numit}` - Generează o fracție cu argumentul *numar* la numărător și respectiv *numit* la numitor.

`\sqrt[n]{arg}` - Generează simbolul pentru rădăcina de ordinul *n* a argumentului *arg*. Dacă nu apare argumentul opțional, atunci se generează rădăcina pătrată.

Trei puncte pot fi generate printr-una din următoarele comenzi:

`\ldots` - Generează trei puncte așezate orizontal, în partea inferioară a rândului. Poate fi utilizată în orice mod.

`\cdots` - Generează trei puncte în centrul rândului.

`\vdots` - Generează trei puncte așezate vertical.

`\ddots` - Generează trei puncte dispuse în diagonală, de la stânga-sus la dreapta-jos.

Exemplul 2.66

	<pre> \documentclass[fleqn]{article} \begin{document} \$x^{2y}, x^{y^2}\$, x^y_{1}\$\par \$x_{2y}, x^{y_1}\$, x_1^y\$\par \^{i}nmul\c tind cu \$n/2\$ rezult\ u a \((m+n)/n\).\par \ (x=\frac{u+z/2}{y^2+1})\ \par O r\ u ad\ u acina p\ u atrat\ u a este \(\sqrt{x+y}\).\par O r\ u ad\ u acina de ordinul \$n\$ este \$\sqrt[n]{2}\$.\par Se scrie \c sirul: \[x_1, \ldots, x_n]\par Puncte centrate: \$a + \cdots + z\$.\par \$\textstyle{a+b}\$ \scriptstyle{c} \scriptscriptstyle{d}\$ \end{document} </pre>
--	--

□

2.7.3 Simboluri matematice**Litere grecești**

O literă grecească se poate genera prin tastarea caracterului *backslash* (\backslash), urmat de numele literei dorite, scris cu litere latine mici, în cazul literelor grecești mici, respectiv cu prima literă latină mare, în cazul literelor grecești mari. Comenzile necesare scrierii literelor grecești sunt listate în tabelul 2.6.

Litere caligrafice

În TeX sunt disponibile 26 de litere caligrafice mari, asociate respectiv literelor latine A, ..., Z și generate de comanda $\backslash\mathcal$.

Exemplul 2.67

	Alege\c ti $\backslash\mathcal{B}$.
--	--------------------------------------

□

Tabelul 2.6 : Litere grecești

--

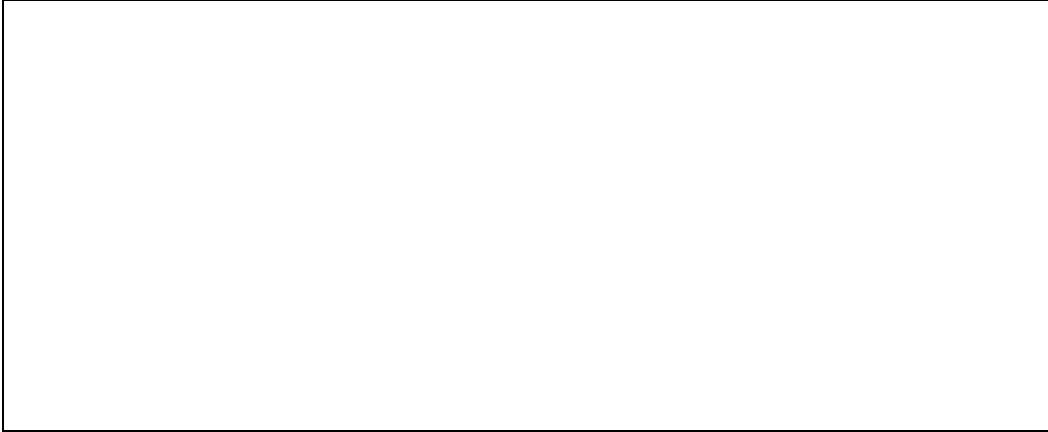
Alte simboluri matematice

În TeX sunt disponibile mai multe seturi de simboluri matematice speciale. Câteva dintre acestea se pot introduce direct de la tastatura standard, cum ar fi $+$ sau $>$. Altele se obțin prin aplicarea comenzilor listate în tabelele 2.7, 2.8, 2.9, 2.10 și 2.11.

Tabelul 2.7 : Operatori binari¹

--

¹ Nu sunt listate comenzile `\lhd`, `\rhd`, `\unlhd` și `\unrhd` disponibile în pachetul `latexsym`.

Tabelul 2.8 : Operatori relaționali²

O serie de comenzi generatoare de simboluri matematice, specificate în notele de subsol 1-4, respectiv asociate tabelelor 2.7-2.10, necesită încărcarea pachetului `latexsym`, prin comanda `\usepackage{latexsym}`.

Tabelul 2.9 : Simboluri de tip săgeată³

² Nu sunt listate comenzile `\sqsubset`, `\sqsupset` și `\Join`, disponibile în pachetul `latexsym`.

³ Nu este listată comanda `\leadsto`, disponibilă în pachetul `latexsym`.

Tabelul 2.10 : Diverse simboluri matematice⁴

--

Tabelul 2.11 : Operatori mari (de mărime variabilă)

--

Simbolurile din tabelul 2.11 au fost generate în stilul *text*. Când sunt generate în cadrul `equation` sau `displaymath`, ele apar mărite.

Exemplul 2.68

```
\begin{document}Exemplu  $\int$  în stilul
\textit{display}:\int\sum_{j=1}^m y_j = \int_0^1 g \int \par
\c si  $\int$  în stilul \textit{text}:
\int\sum_{j=1}^m y_j = \int_0^1 g \int
\end{document}
```

□

⁴ Nu sunt listate comenzile `\mho`, `\Box` și `\Diamond`, disponibile în pachetul `latexsym`.

Simboluri suplimentare, de negație, se pot construi prin aplicarea comenzii `\not`, înaintea comenzii ce generează simbolul nenegat.

Exemplul 2.69

	Dacă $x \not< y$, atunci $x \not\leq y-1$.
--	--

În cazul în care simbolul de negare / nu apare exact în locul dorit, se utilizează comenzile de spațiere, descrise în subsecțiunea 2.7.6. □

Simbolurile funcțiilor matematice uzuale

De regulă, funcțiile matematice uzuale `log`, `exp`, `sin`, etc. se tipăresc cu fontul Roman drept, iar argumentele sunt afișate cu caractere italice. Dacă, pentru editarea unei formule în modul matematic, numele funcțiilor uzuale s-ar introduce direct de la tastatură, atunci ele ar fi tipărite cu font italic. Pentru a evita acest fenomen, se folosesc comenzile listate în tabelul următor.

Tabelul 2.12 : Comenzi pentru generarea numelor unor funcții uzuale

<code>\arccos</code>	<code>\cot</code>	<code>\exp</code>	<code>\lim</code>	<code>\min</code>	<code>\tan</code>
<code>\arcsin</code>	<code>\coth</code>	<code>\gcd</code>	<code>\liminf</code>	<code>\Pr</code>	<code>\tanh</code>
<code>\arctan</code>	<code>\csc</code>	<code>\hom</code>	<code>\limsup</code>	<code>\sec</code>	
<code>\arg</code>	<code>\deg</code>	<code>\ker</code>	<code>\ln</code>	<code>\sin</code>	
<code>\cos</code>	<code>\det</code>	<code>\inf</code>	<code>\log</code>	<code>\sinh</code>	
<code>\cosh</code>	<code>\dim</code>	<code>\lg</code>	<code>\max</code>	<code>\sup</code>	

Pentru tipărirea numelui funcției mod (modulo) se folosesc comenzile `\bmod` (pentru o relație binară) și `\pmod` (pentru o relație cuprinsă între paranteze).

Exemplul 2.70

	$\min(m,n) = a \bmod b \text{ par}$ $x \equiv y \pmod{a+b}$
--	--

Unele din comenzile din tabelul 2.12 se comportă, în raport cu generarea de indici inferior sau superiori, similar cu cele din tabelul 2.11: indicii apar mai mici dacă formula este “în text”, respectiv mai mari dacă aceasta este pe un rând dedicat. □

Observație. TeX poate fi forțat să afișeze altă mărime decât cea asociată, implicit, stilului curent, pentru simbolurile din tabelul 2.11 sau, respectiv, pentru indicii formulilor generate prin unele din comenzile din tabelul 2.12. În acest scop, se folosesc comenzile

`\displaystyle`, `\textstyle`, `\scriptstyle` sau `\scriptscriptstyle`
(subsecțiunea 2.7.1). □

Exemplul 2.71

```
\documentclass{article}
\begin{document}
O formul\u a \^i n stilul \textit{display}:
\[\lim_{n \rightarrow \infty} y = 0.5\]
arat\u a \textit{"^i n text"} astfel:
\(\lim_{n \rightarrow \infty} y = 0.5\).\par
^In text putem scrie mic: $ \frac{1}{C} \int_i = u $
sau mare: $ \frac{1}{C} \int_i = u $.\par
Putem scrie cu indice mic: $e^{x(i)}$ sau cu indice mare:
$e^{\textstyle{y(i)}}$.
\end{document}
```

□

2.7.4 Tablouri matriceale, delimitatori și formule lungi

Tablouri matriceale

Tablourile matriceale se pot scrie folosind cadrul `array`, descris în subsecțiunea 2.3.1 și în subsubsecțiunea 2.3.19.2. Reamintim că sintaxa corespunzătoare este următoarea :

```
\begin{array} [pos] {cols} linii \end{array}
```

TeX asociază, fiecărei formule, câte o linie de centru, pe care va fi plasat, dacă este cazul, semnul + sau -. Un tablou generat de cadrul `array` este, el însuși, o formulă cu o linie de centru. Elementele unei linii a tabloului

sunt poziționate vertical, astfel încât liniile lor de centru să fie, toate, la aceeași înălțime. În mod normal, linia de centru a unui tablou matriceal este situată la jumătatea distanței dintre linia superioară și, respectiv, cea inferioară. Poziția liniei de centru poate fi adusă la nivelul liniei superioare sau inferioare a tabloului, cu valorile t (*top*) și, respectiv b (*bottom*), ale argumentului opțional *pos*.

Exemplul 2.72

	<pre> \documentclass{article} \begin{document} $\left \begin{array}{c} a_1 \\ \vdots \\ a_n \end{array} \right$ $-\left \begin{array}{t} c \\ 1+i \\ \vdots \\ 3- \end{array} \right$ $\left \begin{array}{b} r \\ i \\ j \end{array} \right$ \end{array} \right </pre>
--	--

□

Delimitatori

Tablourile matriceale din exemplul precedent sunt separate, la dreapta și la stânga, prin delimitatori de tip bară verticală. Acest tip și alți delimitatori, frecvent utilizați în formulele matematice, sunt prezentați în tabelul 2.13.

Comenzile din tabelul 2.13 generează delimitatori de mărimea indicată, corespunzătoare mărimii implicite a fontului utilizat (aici 10 pt). Pentru a ajusta mărimea delimitatorilor la mărimea formulei delimitate – lucru necesar în special în cazul tablourilor generate de cadrul `array` - se folosesc comenzile `\left` și `\right`, în pereche (exemplul 2.72). Fiecare din aceste comenzi se aplică chiar înaintea comenzii de generare a delimitatorului.

Deși `\left` și `\right` trebuie să apară în pereche, se poate crea un delimitator invizibil prin introducerea caracterului “.” imediat după comanda `\left` sau `\right`. Această proprietate este utilizată în exemplul 2.73, la scrierea unei funcții cu acoladă.

Tabelul 2.13 : Delimitatori⁵

--

Exemplul 2.73

	<pre> \documentclass{article} \begin{document} \$f(x)=\left\{ \begin{array}{ll} \sin(x)& \mbox{dac\ u a} \end{array} \right. \begin{array}{l} x \leq \pi \\ 0 & \mbox{altfel} \end{array} \end{array} \right.\$ \end{document} </pre>
--	---

□

Formule multilinie

Formulele lungi se pot tipări pe mai multe linii (sau rânduri), folosind cadrele `eqnarray` și `eqnarray*` (subsubsecțiunea 2.3.5). Fiecare rând conține o subformulă. Dacă se folosește cadrul `eqnarray`, atunci fiecare rând – ce conține câte o subformulă – în parte este numerotat. Pentru a evita acest lucru, la sfârșitul fiecărei linii, cu excepția celei din urmă, se introduce comanda `\nonumber` (exemplul 2.18).

Un caracter `+` sau `-`, situat la începutul unei formule sau al unei subformule, este considerat, de TeX, ca un operator unar și va fi tipărit “lipit” de termenul situat la dreapta sa. Dacă termenul respectiv apare într-o subformulă ce face parte dintr-o singură formulă lungă, atunci caracterul `+` sau `-` devine un operator binar și el trebuie tipărit la o mică distanță de termenul

⁵ Pot fi folosite, drept delimitatori, și o parte din simbolurile de tip săgeată din tabelul 2.9; de exemplu, simbolurile generate prin comenzile `\uparrow`, `\downarrow`, `\updownarrow` și prin tandemurile acestora, scrise respectiv cu prima literă mare după caracterul `\`.

situat la dreapta sa. Acest lucru se realizează introducând comanda `\mbox{ }`, înaintea caracterului + sau -.

Exemplul 2.74

	<pre> \documentclass{article} \begin{document} \centering {Aici "+" apare "lipit" de \i\$;} \begin{eqnarray*} y&=&a+b+c\\ && &+i+j+k. \end{eqnarray*}\par \centering {Corect este a\c sa:} \begin{eqnarray*} y&=&a+b+c\\ && &+\mbox{ }+i+j+k. \end{eqnarray*} \end{document} </pre>
--	---

□

2.7.5 Sublinieri și suprapuneri de simboluri

O serie de formule matematice necesită scrierea de simboluri suprapuse. Cadrul `array` permite suprapunerea, pe verticală, a mai multor simboluri, dar nu este comod de utilizat doar pentru sub- sau supralinieri.

Linii și paranteze sub- și suprapuse

Supra- și sublinierile se pot realiza cu următoarele comenzi:

`\overline{formula}` - tipărește argumentul *formula*, cu o linie orizontală deasupra;

`\underline{formula}` - tipărește argumentul *formula*, cu o linie orizontală dedesubt și poate fi folosită în toate modurile.

Următoarele comenzi permit suprapunerea acoladelor orizontale și a formulele matematice.

`\overbrace{formula}` - tipărește argumentul *formula*, cu o acoladă orizontală deasupra;

`\underbrace{formula}` – tipărește argumentul *formula*, cu o acoladă orizontală dedesubt.

Exemplul 2.75

	<pre> \documentclass{article} \begin{document} Se supraliniaz\ u a: \$\overline{\overline{x}^2} +1\$.\par Expresia \$\underbrace{\overbrace{ 1+\sqrt{6}^2+i+j+k}_5}\$ \par are 5 termeni din care 2 sunt numerici. \end{document} </pre>
--	--

□

Accente

Comenzile pentru tipărirea accentelor, descrise în subsecțiunea 2.6.2, se aplică doar în modul paragraf și nu pot fi folosite la scrierea formulelor. În tabelul 2.14, sunt listate comenzile pentru aplicarea de accente în editarea formulelor; litera *a* este aleasă doar spre exemplificare, iar comenzile se pot folosi pentru orice altă literă.

Tabelul 2.14 : Accente în modul matematic

--

Versiuni ale simbolurilor generate de comenzile `\hat` și `\tilde`, care se adaptează la lungimea formulei, se pot obține prin comenzile `\what` și respectiv `\wtilde` (numele lor provine de la *wilde version*). Comenzile `\imath` și `\jmath` generează literele *i* și respectiv *j* fără punct.

Simboluri suprapuse

Comanda

`\stackrel{top}{bot}`

tipărește argumentul reprezentat de *top*, deasupra celui reprezentat de *bot*; mărimea caracterelor argumentului *top* corespunde stilului *superscript*, iar dacă se dorește schimbarea acesteia, atunci se aplică, corespunzător, una din comenzile descrise în subsecțiunea 2.7.1.

Exemplul 2.76

	<pre> \documentclass{article} \begin{document} \[\widehat{x+y}= \widehat{-z}\]\par \[\vec{x} \stackrel{\mathrm{def}}{=} (x_1 \dots x_n)^T\] \end{document} </pre>
--	---

□

2.7.6 Spațieri

Următoarele comenzile generează spații orizontale în modul matematic:

- $\backslash,$ generează un spațiu îngust spre dreapta;
- $\backslash!$ generează un spațiu îngust negativ (i.e. la stânga);
- $\backslash:$ generează un spațiu mediu spre dreapta;
- $\backslash;$ generează un spațiu lat spre dreapta;
- \backslashsqcup generează un spațiu dintre două cuvinte consecutive (\sqcup este *blank*).

Exemplul 2.77

	<pre> \documentclass{article} \begin{document} \begin{tabular}{lcl} \$\sqrt{2} \backslash, x\\$& \wedge \text{i n loc de} \\ & \& \\$\sqrt{2}x\\$ \\ \\$n/ \backslash! \log\{n\}\\$& \wedge \text{i n loc de} \\ & \& \\$n/\log\{n\} \\$ \\ \int \backslash! \int\{z\}\backslash, dx\backslash, dy\\$& \\ & \wedge \text{i n loc de} \\ & \& \int \int\{z\} dx dy \\$ \\ \end{tabular} \end{document} </pre>
--	--

□

Comanda $\backslash,$ poate fi folosită și în modurile paragraf și LR.

2.7.7 Schimbări de stiluri în formulele matematice

În subsecțiunea 2.7.1, au fost descrise comenzile pentru schimbarea mărimii simbolurilor în modul matematic.

Comenzile din tabelul următor permit schimbarea stilului fontului, în cazul numerelor, al literelor latine și grecești. Alte elemente ale formulelor, cum ar fi numele funcțiilor matematice \log , \sin , etc. (tabelul 2.12), rămân neschimbate.

Tabelul 2.15 : Comenzile de setare a fontului în mod matematic și efectele lor

--

Comanda `\mathcal` a fost descrisă și în subsecțiunea 2.7.3. Simbolurile italice, generate de comanda `\mathit`, sunt spațiate diferit de cele generate în stilul italic uzual; `\mathit` este recomandată pentru simboluri matematice compuse din mai multe litere.

Comanda `\boldmath` afectează toate caracterele unei formule și tipărește formulele matematice în stilul bold; poate fi folosită doar în modurile paragraf sau LR și, de aceea, trebuie introdusă, împreună cu argumentul său, într-un *box* generat cu `\mbox` (secțiunea 2.8).

Câteva din comenzile de schimbare de font, ce au fost descrise, sunt aplicabile în exemplul următor.

Exemplul 2.78

```

\documentclass{article}
\begin{document}
$matematic$ -  $\hat{i}$  n mod matematic uzual \par
$\mathit{matematic}$ - la comanda \verb|\mathit| \par
$x+\mbox{\boldmath{\Re{(g)}}}=0$ -
  cu \verb|\boldmath|  $\hat{i}$  n \verb|\mbox| pentru  $\Re{(g)}$  \par
 $\frac{d}{dx} \Psi \log{[\Psi]}$  -
   $\hat{i}$  n mod matematic uzual\par
 $\mathbf{\frac{d}{dx} \Psi \log{[\Psi]}}$  -
  cu \verb|\mathbf| | \par
\mbox{\boldmath{\frac{d}{dx} \Psi \log{[\Psi]}}} -
  cu \verb|\boldmath|  $\hat{i}$  n mod paragraf \par
\end{document}

```



Observație. Presupunem că lucrăm în modul paragraf și că este necesar să edităm o formulă, numită *formula_1*, cu caractere bold și apoi *formula_2*, cu caractere uzuale în modul matematic. Dacă în locul comenzilor

$$\mbox{\boldmath{\$formula_1}}$$

s-ar folosi doar

$$\boldmath{\$formula_1},$$

atunci, în mod nedorit, și *formula_2* ar apărea cu caractere bold (dacă `\mbox` lipsește, compilatorul LaTeX nu dă eroare).

2.8 MODUL LR

În modul LR (*Lef-to-Right*), la fel ca și în modul paragraf, TeX consideră simbolurile primite la intrare ca o secvență de cuvinte cu spații între ele. Însă, spre deosebire de modul paragraf, în modul LR TeX generează un text de la stânga la dreapta, fără a se începe un rând nou și fără a se verifica încadrarea în spațiul disponibil, astfel încât există riscul ca un text, astfel procesat, să iasă din pagină.

Textul procesat în modul LR este introdus într-un *box*, iar la ieșirea din acest mod *box*-ul este așezat în pagină. Esențial este faptul că textul din *box* nu este secționat pe mai multe rânduri.

În continuare sunt descrise comenzile care permit culegerea unui text în modul LR, creându-se *box*-uri. Ele pot fi folosite în orice mod.

$$\mbox{\textit{text}}$$

$$\makebox[lat][pos]{\textit{text}}$$

Aceste comenzi tipăresc argumentul *text* în modul LR, plasându-l într-un *box*; în cazul comenzii `\makebox`, argumentul opțional *lat*

specifică lățimea *box*-ului, iar poziția argumentului *text* în *box* este determinată de următoarele valori ale argumentului *pos*:

c (*center*) – centrat, implicit;

r (*right*) – aliniat la dreapta;

l (*left*) – aliniat la stânga;

s (*stretch*) – spațiile dintre cuvinte sunt ajustate astfel încât textul să umple întregul *box*.

`\fbox{text}`

`\framebox [lat] [pos] {text}`

Aceste două comenzi sunt similare cu `\mbox` și respectiv `\makebox`, cu excepția faptului că se generează și un cadru dreptunghiular, în jurul *box*-ului ce conține argumentul *text*.

`\newsavebox{\cda}`

Declară comanda specificată de argumentul `\cda` ca fiind un *buffer* pentru salvarea unor *box*-uri; comanda `\cda` trebuie să nu fi fost deja definită, i.e. numele *cda* trebuie să fie nou.

`\sbox{\cda} {text}`

`\savebox{\cda} [lat] [pos] {text}`

`\begin{lrbox}{\cda} text \end{lrbox}`

Aceste comenzi introduc argumentul *text* într-un *box* (similar cu `\mbox`), dar în loc să-l și tipărească, îl introduc într-un *buffer*, generat deja la comanda `\newsavebox`. În cadrul `lrbox`, *text* este salvat fără spațiile de la început și de la sfârșit.

`\usebox{\cda}`

Tipărește *box*-ul cel mai recent salvat în *buffer*-ul `\cda` (creat la comanda `\newsavebox`).

`\raisebox{raise_len} [hght] [dpth] {text}`

Această comandă creează un *box* prin tipărirea argumentului *text* în modul LR; *text* este ridicat în *box* cu distanța specificată de *raise_len* și extins, deasupra și dedesubtul rândului, cu distanțele specificate de argumentele *hght* și respectiv *dpth*. Dacă argumentul *dpth* sau ambele argumente opționale lipsesc, atunci TeX va folosi extensia curentă a *box*-ului.

`\rule{raise_len}{lat}{hght}`

Generează un dreptunghi solid, de lățime *lat* și înălțime *hght*, ridicat deasupra rândului la distanța specificată de argumentul *raise_len* (sau coborât, dacă *raise_len* are valoare negativă). *raise_len* are valoarea implicită 0in.

`\width`

`\height`

`\depth`

`\totalheight`

Sunt comenzi de tip lungimi, ce pot fi aplicate argumentelor *lat* al comenzilor `\makebox`, `\framebox` și `\savebox` și argumentelor *raise_len*, *hght* și *dpth* ale comenzii `\raisebox`. Se referă la dimensiunile (lățime, înălțime, adâncime și respectiv înălțime+adâncime ale) *box*-ului obținut prin tipărirea argumentului *text*.

Stilul este controlat de următoarele comenzi:

`\fboxrule` precizează lățimea liniei trasate de `\fbox` și `\framebox`;

`\fboxsep` precizează spațiul dintre conținutul și marginile *box*-ului generat cu `\fbox` și `\framebox`.

Comanda `\mbox` este folosită în exemplul 2.78 din finalul secțiunii precedente. Iată alte exemple de generare de text în modul LR.

Exemplul 2.79

	<pre> \documentclass{article} \begin{document} Apare un \framebox[1in][1]{om} la \fbox{orizont}.\par Compara\c ti \fbox{acest \em{box}} cu \fbox{ \rule[-0.5cm]{0cm}{1cm} acest \em{box} }.\par XXX\framebox[0.5in] {diver\c si oameni}XXX. \end{document} </pre>
--	---

□

Exemplul 2.80

	<pre> \documentclass{article} \begin{document} Text \raisebox{0.6ex}{\em{ridicat}} sau \raisebox{-0.6em}{\em{cobor\^at}}. \end{document} </pre>
--	---

□

Exemplul 2.81

	<pre> \documentclass{article} \begin{document} \newsavebox{\joc} \savebox{\joc}[1.5cm]{ap\u a} Este doar \usebox{\joc},\par \usebox{\joc}, \usebox{\joc},\par peste tot. \end{document} </pre>
--	--

□

Observație. Alte comenzi pentru crearea de *box*-uri au fost descrise în subsecțiunea 2.3.13.

□

2.9 OBSERVAȚII FINALE

Dificultățile legate de însușirea limbajului LaTeX sunt mai mult de natură psihologică. Odată deprins, LaTeX asigură editarea textelor, conform celor mai exigente standarde de tehnoredactare. În plus, fișierul de ieșire, cu extensia `.dvi`, este – după cum reiese și din denumirea sa în engleză: *Device Independent* – independent de *drive*-ul de imprimantă.

Compilerul LaTeX este disponibil, gratuit, pe Internet. Reamintim că una din adresele de la care poate fi adusă versiunea sa pentru Win95 este:

www.stat.uiowa.edu/resources/help/win95/latex.html

Prezentul capitol este doar o introducere privind limbajul LaTeX. Textul său este o combinație între informația de tip “*User’s Guide*” și cea de tip “*Reference Manual*”. Anexa C conține o aplicație din domeniul teoriei ecuațiilor diferențiale, iar Anexa D cuprinde o listă a principalelor comenzi

LaTeX, sub formă de memorator. Cea mai detaliată lucrare privind LaTeX rămâne, până în prezent, cartea lui Leslie Lamport [Lamport 1994].

Din motive pur conjuncturale, acest capitol, la fel ca și întreaga lucrare, a fost editat în Microsoft Word97, iar exemplele și o parte din tabele au fost procesate în LaTeX, captate și apoi inserate în text. Dintre lucrările publicate în limba română, realizate în întregime în LaTeX și care pun în evidență avantajele deosebite ale procesării textelor științifice în acest limbaj, recomandăm tratatul de calcul numeric matriceal de B. Dumitrescu, C. Popea și B. Jora [Dumitrescu 1998].