

5.3 L^AT_EX

5.3.1 Uvod

L^AT_EX je sistem za logično urejanje besedil. T_EX, ki je osnova sistema L^AT_EX, je leta 1977 začel razvijati Donald Knuth z Univerze Stanford v ZDA [8]. Profesor Knuth je napisal veliko knjig s področja računalništva. Ker ni imel na voljo dovolj dobrega orodja za pisanje besedil, ki bi omogočal med drugim tudi pisanje zapletenih matematičnih izrazov, je sklenil, da ga napiše kar sam. Tako je nastal T_EX, sistem za oblikovanje besedil, ki je posebej primeren za pisanje znanstvenih besedil z veliko matematike. T_EX [7] je hitro postal priljubljen tudi pri drugih znanstvenikih in na drugih znanstvenih področjih ter pri vseh tistih, ki želijo zelo kvalitetno in fleksibilno tipografsko oblikovanje. Danes je T_EX standard na številnih založniških področjih. Uporablja se za tehnične in naravoslovne knjige, konferenčne zbornike, slovarje in leksikone, večjezične knjige itd. Implementacije sistema T_EX obstajajo praktično za vse obstoječe operacijske sisteme. Zaradi stabilnosti in kompatibilnosti se je Donald Knuth odločil, da se T_EX ne bo več spreminjal.

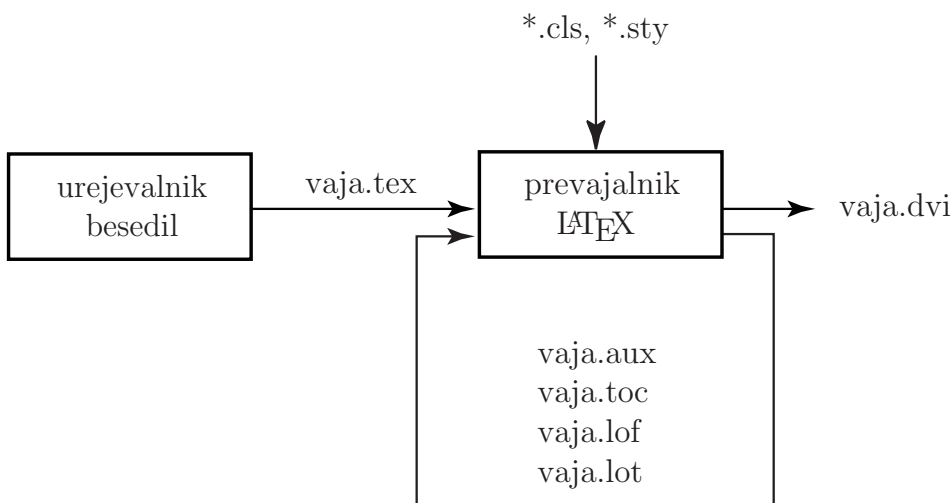
V začetku 80. let je Leslie Lamport začel razvijati L^AT_EX [9], ki je sistem makro ukazov na osnovi sistema T_EX. Kot vsak sistem oziroma programska oprema se je tudi L^AT_EX razvijal skozi več različic. Trenutno je v uporabi različica L^AT_EX2 ϵ , ki jo obravnavamo tudi v tej knjigi. T_EX in L^AT_EX sta v javni rabi in zato obstaja cela vrsta njunih brezplačnih implementacij, ki so dosegljive na svetovnem spletu.

Osnovne datoteke in način dela

V urejevalniku besedil pišemo besedilo, ki ga opremimo z ukazi za formatiranje. To z ukazi za formatiranje obogateno besedilo moramo nato prevesti, da dobimo formatirano besedilo za izpis na tiskalniku (slika 5.30). Prevajalnik pri prevajanju upošteva tudi razne stilske datoteke, ki definirajo obliko formatiranega besedila. Pri prevajanju se poleg formatiranega besedila generira še cela vrsta pomožnih datotek, ki jih prevajalnik uporabi pri ponovnem prevajanju.

Vhodno datoteko v sistemu L^AT_EX, ki jo lahko pišemo s poljubnim urejevalnikom, označimo s podaljškom `tex`: na primer `vaja.tex`. Oglejmo si podaljške in vlogo najpomembnejših datotek:

- `*.tex` vhodna datoteka z besedilom in ukazi za formatiranje,
- `*.dvi` formatirana izhodna datoteka oziroma prevod datoteke `*.tex`,
- `*.aux` pomožna datoteka,



Slika 5.30: Način dela s sistemom L^AT_EX in osnovne datoteke

- *.toc kazalo,
- *.lof seznam slik,
- *.lot seznam tabel,
- *.sty stilske datoteke,
- *.cls oblikovne predloge.

Pri obsežnejših besedilih si pri navajanju literature lahko pomagamo tudi z datotekami BIB_TE_X, v katere shranjujemo podatke o citirani literaturi (podaljška `bbl` in `bib`).

5.3.2 Značilnosti sistema L^AT_EX pod Linuxom

Pisanje kode L^AT_EX v urejevalniku

Ena izmed tipičnih napak začetnih uporabnikov sistema L^AT_EX, ki smo jo spoznali skozi prakso, je ta, da želijo kodo zapisati kar v urejevalniku OpenOffice.org Writer. Ta urejevalnik v osnovi seveda ne shranjuje datotek v tekstovni obliki, zato pozorno preberite naslednji odstavek, ki opisuje pravilen postopek pisanja kode!

Pri opisu dela v terminalskem načinu Linuxa smo spoznali ukaz `emacs`, s katerim zaženemo istoimenski urejevalnik! V primeru, da na priloženi zgoščenki Slix tega ukaza ne pozna, poskusite z ukazom `xemacs` ali pa z ukazom `Emacs`. Če želimo ustvariti novo datoteko z imenom `vaja.tex`, v ukazni vrstici napišemo `emacs vaja.tex &` in ukaz potrdimo s pritiskom

na tipko `<Enter>`. Odpre se nam okno urejevalnika, ki vsebuje vsebino zahtevane datoteke. Seveda, če ta datoteka še ne obstaja, je vsebina okna prazna. Uporabljamo seveda lahko tudi katerega od preprostejših urejevalnikov besedil.

Obstajajo tudi razna integrirana orodja, nekakšni vmesniki, ki omogočajo večjo preglednost nad sintakso kode s pomočjo barv, lažji dostop do prevajalnika in prikazovalnika, do osnovnih gradnikov ipd. Na Slixu se takšno orodje imenuje `kile`, `texmacs` pa je sorodno orodje, ki omogoča urejanje po načelu WYSIWYG.

Prevajanje kode \LaTeX

Ko končamo z delom v urejevalniku `emacs`, kar pomeni, da smo shranili vsebino v datoteko s končnico `tex`, moramo ustvarjeno datoteko prevesti (slika 5.30). Če želimo prevesti datoteko z imenom `vaja.tex`, moramo v ukazni vrstici napisati:

```
latex vaja.tex
```

Če v kodi ni napak, bo \LaTeX javil, da je uspešno ustvaril prevod, sicer pa izpostavi mesto, ki ga ne more prevesti in poskuša identificirati napako. Ta informacija skoraj vedno že zadostuje za odpravo napake.

Prikaz prevoda

Ko odpravimo vse napake v kodi in uspešno prevedemo datoteko, moramo le še prikazati prevod na ekranu. To storimo s klicem prikazovalnika datotek s končnico `dvi`, na primer z ukazom `xdvi`. Če želimo prikazati prevod ustvarjene datoteke z imenom `vaja.tex`, moramo v ukazni vrstici napisati:

```
xdvi vaja ali xdvi vaja.dvi
```

Recimo, da želimo sedaj prevod, ki ima podaljšek `dvi`, spremeniti v datoteko PostScript, ki ima podaljšek `ps`. To naredimo tako, da v ukazni vrstici uporabimo ukaz `dvips`:

```
dvips -o vaja.ps vaja.dvi
```

Datoteko PostScript lahko sedaj pošljemo direktno na tiskalnik PostScript z ukazom `lpr vaja.ps`.

Podobno lahko z ukazom `pdflatex` ustvarimo tudi datoteko, ki ima podaljšek `pdf`:

```
pdflatex vaja.tex
```

Na izhodu dobimo datoteko `vaja.pdf`. Format PDF (Portable Document Format) podjetja Adobe Systems Incorporated postaja standarden format za prenos elektronskih dokumentov po svetovnem spletu.

5.3.3 Zgradba datotek s končnico `tex`

Vhodna datoteka sistema L^AT_EX je sestavljena iz glave, jedra in repa. Jedro predstavlja vsebino dokumenta, zato si oglejmo, kako izgledata glava in rep datoteke s končnico `tex`:

```
\documentclass[12pt,a4paper]{article}
\usepackage[slovene]{babel}
\begin{document}

\end{document}
```

Prve tri vrstice bodo za nas predstavljale glavo dokumenta, zadnja pa rep. Med glavo in rep vnesemo vsebino dokumenta.

Tukaj pa velja omeniti naslednjo tipično napako začetnih uporabnikov sistema L^AT_EX: datoteko pozabijo opremiti z glavo in repom. Takšna datoteka pa se seveda ne more prevesti uspešno. Zato v datoteki vedno najprej ustvarite glavo in rep, nato pa dodajte med ta dva dela željeno vsebino!

Na kratko komentirajmo osnovne lastnosti sistema L^AT_EX, ki so razvidne iz zapisane kode:

- Vsebina dokumenta mora vedno biti vpeta med glavo in rep dokumenta, sicer se dokument ne bo prevedel.
- Vsak ukaz se v sistemu L^AT_EX začne z znakom `\`.
- Oznaka `12pt` podaja v pikah osnovno velikost pisave v dokumentu. Prav tako bi lahko na primer uporabili oznako `11pt` za pisavo velikosti 11 pik. Če ne predpišemo velikosti, L^AT_EX samodejno uporabi pisavo velikosti 10 pik.
- Oznaka `a4paper` podaja velikost papirja, ki ga uporabljamo za natis dokumenta.
- Oznaka `article` podaja vrsto dokumenta. L^AT_EX pozna več vrst dokumentov, ki pravzaprav predstavljajo oblikovne vzorce, največkrat uporabljene pa so:
`article` – oblikovna predloga za pisanje člankov,

`report` – oblikovna predloga za pisanje poročil,

`book` – oblikovna predloga za pisanje knjig.

- V drugi vrstici glave sistemu \LaTeX povemo, da bomo uporabljali paket `babel` (Babilon po slovensko). Z izbiro jezika `slovene` zahtevamo, da se vse pomožne besede, ki se samodejno generirajo v dokumentu, izpišejo v slovenščini namesto v angleščini: *contents/kazalo*, *chapter/poglavje*, *bibliography/literatura*, *figure/slika* itd. Paket `babel` vsebuje opcije za skoraj vse jezike, ki uporabljajo latinsko pisavo.
- Osnovni gradniki datotek sistema \LaTeX so okolja, ki imajo skupno osnovno sintakso:

```
\begin{xyz}
```

```
\end{xyz}
```

pri čemer `xyz` ponazarja ime nekega okolja: `document`, `figure`, `table`, `array`, `eqarray`, `tabular`, `verbatim`, `itemize`, `enumerate`, `equation`, `thebibliography` itd. Med `begin` in `end` postavimo ukaze, bolj ali manj specifične za določeno okolje. Okolja lahko tudi gnezdimo. Več o okoljih bomo spregovorili v nadaljevanju. Na tem mestu pa izpostavimo zgolj dejstvo, da je okolje vseh okolij okolje `document`:

```
\begin{document}
```

```
\end{document}
```

5.3.4 Kako pišemo šumnike?

Šumniki sicer niso edini posebni znaki, ki jih angleščina ne pozna, vendar si zaradi njihovega pomena pri pisanju slovenščine najprej oglejmo pisanje šumnikov. Šumnike lahko pišemo na več načinov:

- Z ukazom `\{c}` dobimo č.
- Z ukazom `\v c` prav tako dobimo č.
- Če v glavi dokumenta za prvo vrstico dodamo naslednjo kodo:

```
\catcode'\ '=13
```

```
\def'\ '#1{\v #1}
```

ali če uporabljamo paket `babel` z opcijo `slovene`:

```
\usepackage[slovene]{babel}
```

lahko č zapišemo še krajše in bolj pregledno kot "c³.

- Če pišemo besedilo v kodnem naboru ISO-8859-2 (Latin 2) (ta je v slovenskem okolju navadno privzeti nabor), potem lahko v glavo dokumenta za prvo vrstico dodamo ukaz:

```
\usepackage[latin2]{inputenc}
```

Ta nam omogoča, da lahko šumnike pišemo kar direktno v urejevalnik in jih po prikazu prevoda na zaslonu tudi pravilno vidimo.

Podobno, če uporabljamo kodni nabor Windows-1250 za pisanje šumnikov, uporabimo ukaz:

```
\usepackage[cp1250]{inputenc}
```

5.3.5 Posebni znaki

Podobno kot šumnike lahko zapišemo tudi črke drugih jezikov in naglase. Oglejmo si nekaj najpogostejših:

ć	<code>\' {c}</code>	ê	<code>\^ {e}</code>
è	<code>\' {e}</code>	ë	<code>\" {e}</code>
ç	<code>\c {c}</code>	ł	<code>\l</code>
å	<code>\aa</code>	ß	<code>\ss</code>
£	<code>\pounds</code>		

V sistemu L^AT_EX imamo deset znakov s posebnim pomenom. Pomen znaka `\` smo že spoznali, ostali pa so:

`# $ % & ~ _ ^ { }`

Pomen posameznih posebnih znakov bomo spoznali v nadaljevanju. Včasih je potrebno te znake tudi izpisati. To storimo na naslednji način:

#	<code>\#</code>	&	<code>\&</code>
-	<code>_</code>	\$	<code>\\$</code>
%	<code>\%</code>	{	<code>\{</code>
}	<code>\}</code>	\	<code> \\$\backslash\$</code>
^	<code> \$^\wedge\$</code>	~	<code> \$^\sim\$</code>

V zadnjih treh primerih je ukaz znotraj matematičnega okolja, to je med znakoma `$`. Več o matematičnih okoljih bomo povedali v nadaljevanju.

³V kodi in tukaj uporabljamo dvojni narekovaj.

5.3.6 Slogi in velikosti pisav

V sistem \LaTeX je vključena družina pisav *Computer Modern*, ki jo je oblikoval avtor \TeX -a Donald Knuth. V sistemu \LaTeX pa seveda lahko uporabljamo tudi druge pisave, predvsem je enostavna uporaba pisav v formatu PostScript. Ta učbenik je napisan s pomočjo sistema \LaTeX in z družino pisav Computer Modern.

V sistemu \TeX lahko stavimo tudi besedila, ki ne uporabljajo latinske pisave, na primer v cirilici, grščini, arabščini, hebrejščini, japonščini in še celi vrsti drugih pisav, ki se celo ne pišejo le z leve proti desni in od zgoraj navzdol kot latinske pisave.

Najprej si oglejmo ukaze za osnovne sloge pisav:

```
\rm pokončno – {\rm pokon\v cno}
\it kurziva – {\it kurziva}
\bf krepko – {\bf krepko}
\sc MALE KAPITELKE – {\sc male kapitelke}
\sf linearna pisava – {\sf linearna pisava}
\tt pisalni stroj – {\tt pisalni stroj}
```

Privzeti slog pisave v sistemu \LaTeX je pokončni (roman \rm). Kot je razvidno iz zgornjih primerov, delovanje omenjenih ukazov omejimo tako, da jih vstavimo med zavita oklepaja $\{\}$.

Pomanjkljivost zgornjih ukazov je, da jih ni možno kombinirati, zato zapišimo še primere ekvivalentnih ukazov, ki omogočajo tudi kombiniranje slogov pisav:

```
\textrm pokončno – \textrm{pokon\v cno}
\textit kurziva – \textit{kurziva}
\textbf krepko – \textbf{krepko}
\textsc MALE KAPITELKE – \textsc{male Kapitelke}
\textsf linearna pisava – \textsf{linearna pisava}
\texttt pisalni stroj – \texttt{pisalni stroj}
```

Sedaj lahko zapišemo tudi poljubne kombinacije zgornjih slogov pisav:

```
Pomembno pa je, \textsf{\textbf{Pomembno pa je,}}
DA LAHKO PIŠEMO \texttt{\textsc{da lahko pi\v semo}}
v takšnem ali \textsf{\textit{v tak\v snem ali}}
drugačnem slogu. \texttt{\textit{druga\v cnem slogu.}}
```

Osnovna velikost pisave je, kot že rečeno, podana v glavi datoteke s končnico *tex*, vendar jo lahko znotraj dokumenta spreminjamo z naslednjimi relativnimi ukazi:

<code>\tiny</code>	<code>beseda – {\tiny beseda}</code>
<code>\scriptsize</code>	<code>beseda – {\scriptsize beseda}</code>
<code>\footnotesize</code>	<code>beseda – {\footnotesize beseda}</code>
<code>\small</code>	<code>beseda – {\small beseda}</code>
<code>\normalsize</code>	<code>beseda – {\normalsize beseda}</code>
<code>\large</code>	<code>beseda – {\large beseda}</code>
<code>\Large</code>	<code>beseda – {\Large beseda}</code>
<code>\LARGE</code>	<code>beseda – {\LARGE beseda}</code>
<code>\huge</code>	<code>beseda – {\huge beseda}</code>
<code>\Huge</code>	<code>beseda – {\Huge beseda}</code>

5.3.7 Struktura dokumenta

Omenili smo že priporočljivo strukturo seminarских nalog, v tem poglavju pa si oglejmo, kako s sistemom L^AT_EX ustvarimo naslovno stran, poglavja, dvokolonski dokument, spisek uporabljene literature, kako se sklicujemo na literaturo in kako ustvarimo kazalo, spisek slik in tabel.

Vzemimo pod drobnogled besedilo v sistemu L^AT_EX na sliki 5.31. Njegov prevod je prikazan na sliki 5.32. Komentirajmo neznane ukaze:

- S posebnim znakom % lahko zapišemo komentarje v dokumentu. Ti komentarji v prevodu niso vidni.
- Če za ukazom `\begin{document}` uporabimo ukaze `\title`, `\author` in `\date`, kjer med zavitima oklepajema {} navedemo naslov, avtorje in datum, nato pa za temi ukazi uporabimo ukaz `\maketitle`, bo L^AT_EX ob prevodu ustvaril naslov, če je izbrani tip dokumenta `article`, ali naslovno stran, če je izbrani tip dokumenta `book` ali `report`.
- Poglavja in razdelke ustvarjamo z ukazi: `\section` (razdelek), `\subsection` (podrazdelek), `\subsubsection` (pod-podrazdelek) in `\chapter` (glavno poglavje, ki ga lahko uporabimo le v tipu dokumenta `book` ali `report`). Ime poglavja navedemo med zavitima oklepajema {}, ki sledijo ukazu.
- O okolju `\thebibliography` bomo spregovorili malo kasneje.

In kako ustvarimo dvokolonski dokument? Enostavno tako, da prvo vrstico glave dokumenta iz

```
\documentclass[12pt]{article}
```

spremenimo v


```

\documentclass[12pt,a4paper]{article} %Ta dokument bo enokolonski!
\usepackage[slovene]{babel}

\begin{document}

\title{Superkvadri\v cni modeli}
\author{Franc Solina}
\date{6. julij 2000}
\maketitle

\section{Definicija}
Superkvadriki so 3D modeli, ki se uporabljajo v ra\v cunalni\v skem
vidu za mo\deliranje in segmentacijo globinskih slik
\cite{Kluwer_2000}.

\subsection{Razvoj metode}
Rekonstrukcijo posami\v cnih superkvadrikov~\cite{london_87} smo
zdru\v zili z metodo segmentacije ‘opi\v si in izberi’
\cite{leonardis93}. Rekonstrukcijo in segmentacijo superkvadrikov
smo testirali na globinskih slikah~\cite{PAMI97}.

\begin{thebibliography}{1}
\bibitem{london_87}
R. Bajcsy and F. Solina. Three dimensional object representation
revisited. In {\em Proceedings First International Conference on
Computer Vision}, pages 231--240, London, England, June 1987.
\bibitem{Kluwer_2000}
A. Jakli\v{c}, A. Leonardis, and F. Solina. {\em Segmentation and
Recovery of Superquadrics}. Kluwer, Dordrecht, 2000.
\bibitem{leonardis93}
A. Leonardis. {\em Image Analysis Using Parametric Models:
Model-Recovery and Model-Selection Paradigm}. PhD thesis,
University of Ljubljana, Faculty of Electrical Engineering and
Computer Science, Ljubljana, Slovenia, 1993.
\bibitem{PAMI97}
A. Leonardis, A. Jakli\v{c}, and F. Solina. Superquadrics for
segmentation and modeling range data. {\em IEEE Transactions on
Pattern Recognition and Machine Intelligence}, 19(11):1289--1295,
November 1997.
\end{thebibliography}

\end{document}

```

Slika 5.31: Izvorno besedilo v sistemu L^AT_EX

Superkvadrični modeli

Franc Solina

6. julij 2000

1 Definicija

Superkvadriki so 3D modeli, ki se uporabljajo v računalniškem vidu za modeliranje in segmentacijo globinskih slik [2].

1.1 Razvoj metode

Rekonstrukcijo posamičnih superkvadrikov [1] smo združili z metodo segmentacije “opiši in izberi” [3]. Rekonstrukcijo in segmentacijo superkvadrikov smo testirali na globinskih slikah [4].

Literatura

- [1] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proceedings First International Conference on Computer Vision*, pages 231–240, London, England, June 1987.
- [2] A. Jaklič, A. Leonardis, and F. Solina. *Segmentation and Recovery of Superquadrics*. Kluwer, Dordrecht, 2000.
- [3] A. Leonardis. *Image Analysis Using Parametric Models: Model-Recovery and Model-Selection Paradigm*. PhD thesis, University of Ljubljana, Faculty of Electrical Engineering and Computer Science, Ljubljana, Slovenia, 1993.
- [4] A. Leonardis, A. Jaklič, and F. Solina. Superquadrics for segmentation and modeling range data. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 19(11):1289–1295, November 1997.

Superkvadrični modeli

Franc Solina

6. julij 2000

1 Definicija

Superkvadrki so 3D modeli, ki se uporabljajo v računalniškem vidu za modeliranje in segmentacijo globinskih slik [2].

1.1 Razvoj metode

Rekonstrukcijo posamičnih superkvadrikov [1] smo združili z metodo segmentacije "opiši in izberi" [3]. Rekonstrukcijo in segmentacijo superkvadrikov smo testirali na globinskih slikah [4].

of Electrical Engineering and Computer Science, Ljubljana, Slovenia, 1993.

- [4] A. Leonardis, A. Jaklič, and F. Solina. Superquadrics for segmentation and modeling range data. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 19(11):1289–1295, November 1997.

Literatura

- [1] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *Proceedings First International Conference on Computer Vision*, pages 231–240, London, England, June 1987.
- [2] A. Jaklič, A. Leonardis, and F. Solina. *Segmentation and Recovery of Superquadrics*. Kluwer, Dordrecht, 2000.
- [3] A. Leonardis. *Image Analysis Using Parametric Models: Model-Recovery and Model-Selection Paradigm*. PhD thesis, University of Ljubljana, Faculty

```
\documentclass[12pt,twocolumn]{article}
```

Rezultat te zamenjave v izvornem besedilu je viden na sliki 5.33.

Komentirajmo še osnovne gradnike v poglavju **Literatura** s slike 5.32:

- Vire naštejemo v okviru okolja `thebibliography`.
- Z oznako `{1}` povemo sistemu L^AT_EX, da bomo navedli največ 9 virov. Z oznako `{11}` pa bi povedali, da bo v spisku literature največ 99 virov. S pomočjo te informacije L^AT_EX ustrezno vertikalno poravna oštevilčenje virov.
- Ukaz `\bibitem` uporabimo za navajanje posameznih virov. Med zavitima oklepajema `{}` nato podamo simbolično ime vira, ki ga bomo v dokumentu uporabili pri sklicevanju na ta vir, nadaljujemo pa z opisom vira.
- Največkrat ime vira poimenujemo po prvem avtorju vira, pri tem pa moramo seveda biti pozorni, da se nam imena ne ponavljajo. V našem primeru smo prvi vir poimenovali po kraju konference (`london_87`), drugega po založbi (`Kluwer_2000`), tretji vir pa po avtorju (`leonardis93`). Razlikovati moramo med velikimi in malimi črkami!

Ko ustrezno poimenujemo vire, se lahko nanje v besedilu tudi sklicujemo, L^AT_EX pa poskrbi, da se številčenje referenc samodejno obnavlja. Za pravilno številčenje je po vsaki spremembi v literaturi potrebno izvorno kodo L^AT_EX **vedno prevesti dvakrat**. V izvornem besedilu se na posamezen vir sklicujemo s pomočjo ukaza `cite`, kar je razvidno iz primera na sliki 5.31. Kot privzet način prikazovanja citiranih virov je v sistemu L^AT_EX uporabljeno številčenje v oglatih oklepajih. Z uporabo drugih slogov citiranja je možno citate prikazati tudi na druge načine, na primer s priimki avtorjev in letnico izdaje v okroglih oklepajih – (Bajcsy and Solina, 1991).

Omenjeni način podajanja virov, ko kar v okviru izvornega besedila naštejemo vse vire, je smotrno, če je virov malo in če istih virov ne uporabljamo večkrat. Sicer pa se nam izplača vzdrževati lastno bazo virov (datoteke s podaljškom `bib`), iz katere s pomočjo programa BIB_TE_X samodejno generiramo spisek citirane literature. To so datoteke s podaljškom `bb1`, ki jih L^AT_EX vključi namesto okolja `thebibliography`.

V našem primeru smo v spisku literature na slikah 5.31 in 5.32 uporabili:

1. članek, objavljen v konferenčnem zborniku,
2. knjigo

3. doktorsko disertacijo in
4. članek, objavljen v znanstveni reviji.

To so štiri najpogostejše vrste publikacij, ki jih citiramo v strokovnih in znanstvenih besedilih. Pozorno pogledjte, katere elemente in po kakšnem vrstnem redu jih je potrebno navesti v spisku literature za posamezno vrsto publikacije.

Zaradi logičnega urejanja besedil nam \LaTeX omogoča enostavno ustvarjanje kazal ter spiskov slik in tabel. Na mestu, kjer v dokumentu želimo, da se pojavi določen seznam, enostavno zapišemo ustrezen ukaz:

```
\tableofcontents    ustvari kazalo
\listoffigures      ustvari spisek slik
\listoftables       ustvari spisek tabel
```

prevedemo dvakrat in dobimo rezultat.

5.3.8 Okolja

Do sedaj smo spoznali že dve osnovni okolji: `document` in `thebibliography`. V tem poglavju bomo omenili še ostala pomembna okolja, z izjemo tistih, ki so specifična za pisanje matematičnih izrazov.

Sredinska poravnava

Če želimo nek del dokumenta poravnati na sredino strani oziroma kolone, uporabimo okolje `center`:

```
\begin{center}
\end{center}
```

Kot smo že omenili, želeni del dokumenta damo med obe vrstici, torej med `begin` in `end`. To velja za vsa okolja tipa `begin-end`.

Naštevanje

Okolje za naštevanje se imenuje `itemize`:

```
\begin{itemize}
\item
\end{itemize}
```

Vsako točko spiska naredimo z ukazom `\item`, ki mu sledi želeni del besedila. Na primer: `\item Potrdilo o dr\v zavljanstvu`.

Številčenje

Okolje za številčenje se imenuje `enumerate`:

```
\begin{enumerate}
\item
\end{enumerate}
```

Tudi tukaj naredimo točko spiska z ukazom `\item`, le da jih v tem primeru L^AT_EX oštevilči.

Opisno naštevanje

Okolje za opisno naštevanje se imenuje `description` in nam omogoča, da namesto začetnega znaka ali številke spiska uporabimo (skoraj) poljubno zaporedje:

```
\begin{description}
\item[]
\end{description}
```

Podobno kot pri prejšnjih dveh okoljih, tudi tukaj naredimo točko spiska z ukazom `\item`. Željeno zaporedje, ki ga bomo uporabili namesto znaka ali številke spiska, zapišemo med oglate oklepaje `[]`. Na primer:

```
\item[Potrdila:] o dr\v zavljanstvu,... .
```

Okolje za dobesedni izpis

Okolje `verbatim` nam omogoča, da izpišemo besedilo točno tako, kot smo ga zapisali v vhodni datoteki s končnico `tex`:

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

To okolje nam pride prav predvsem tam, kjer želimo izpostaviti strukturo nekega dela dokumenta (koda programa, ukaz `LATEX` itd.). `LATEX` bo prikazal vsak presledek, vsak znak, vsako število in vsak razlom vrstice točno tako, kot smo zapisali. Primer:

```
\begin{verbatim}
For i:= 1 To 20 Do
  Write('\v{c}\v{s}\v{z}')
\end{verbatim}
```

bomo v prevodu videli kot:

```
For i:= 1 To 20 Do
  Write('\v{c}\v{s}\v{z}')
```

Vrstični dobessedni izpis

Okolje za vrstični dobessedni izpis se pomensko razlikuje od prej opisanega okolja le po tem, da lahko prikaže le nekaj besed ali največ eno vrstico besedila, saj tega okolja `LATEX` ne razdeli na več vrstic, tudi če je okolje daljše od možne dolžine vrstice:

```
\verb++
```

Največkrat nam to okolje pride prav tam, kjer moramo med samim besedilom dokumenta izpisati besedilo točno tako, kot smo ga zapisali v vhodni datoteki s končnico `tex`. Ukaz, ki ga uporabimo, se imenuje `\verb`; besedilo, ki ga izpisujemo, pa vkleščimo med dva enaka znaka, na primer `+`. Primer:

```
\verb+__Write('\v{c}\v{s}\v{z}')+
```

bomo v prevodu videli kot: `__Write('\v{c}\v{s}\v{z}')`.

Pisanje opomb

Če želimo v dokumentu zapisati opombo, ki se bo pojavila na dnu strani, se postavimo na tisto mesto v dokumentu, na katero se opomba nanaša, in zapišemo opombo s pomočjo ukaza `\footnote`, kjer med zavita oklepaja `{}` zapišemo želeno opombo:

`\footnote{}`

Okolje za poravnavo

Okolje za poravnavo `tabular` je pravzaprav osnovno okolje za gradnjo tabel, ki si ga bomo ogledali v naslednjem poglavju. Razložimo delovanje na naslednjem primeru:

```
\begin{tabular}{|clr|}\hline
ena & dva & tri\\\cline{2-2}
1 & 2 & 3\\\hline
\end{tabular}
```

V prevodu ta del kode vidimo kot:

ena	dva	tri
1	2	3

Komentar kode:

- Osnovno okolje:

```
\begin{tabular}
\end{tabular}
```

- Pomen izraza med zavitima oklepajema `{|clr|}`:

- znak `c` podaja ukaz sistemu L^AT_EX, da naj bo besedilo v prvem stolpcu sredinjeno (*c* – *center*),
- znak `l` podaja ukaz sistemu L^AT_EX, da naj bo besedilo v drugem stolpcu poravnano levo (*l* – *left*),
- znak `r` podaja ukaz sistemu L^AT_EX, da naj bo besedilo v tretjem stolpcu poravnano desno (*r* – *right*),
- znak `|` podaja ukaz sistemu L^AT_EX, da naj bo na podanem mestu v tabeli navpična črta,
- celoten izraz pove, da smo ustvarili strukturo s tremi stolpci, ki je na začetku in na koncu omejena z navpično črto. Skupno število znakov `c`, `l` in `r` torej določa število stolpcev v tabeli.

- Ukaz `\hline` naredi vodoravno črto.
- Ukaz `&` določa mejo med stolpci.

- Ukaz `\cline{2-2}` naredi vodoravno črto čez drugi stolpec. Če bi namesto tega napisali `\cline{2-3}`, bi \LaTeX naredil črto čez drugi in tretji stolpec.
- Ukaz `\` določa zaključek vrstice.

Na tem mestu omenimo le še ukaz za združevanje stolpcev:

```
\multicolumn{}{}{}
```

Primer:

```
\begin{tabular}{|clr|}\hline
\multicolumn{2}{|c|}{1+2} & 3\\\hline
ena & dva & tri\\\cline{2-2}
1 & 2 & 3\\\hline
\end{tabular}
```

V prevodu ta del kode vidimo kot:

1+2	3	
ena	dva	tri
1	2	3

Komentar kode v ukazu `\multicolumn`:

- Med prvim parom zavutih oklepajev `{}` podamo število, ki pove, koliko stolpcev združujemo.
- Med drugim parom zavutih oklepajev `{}` podamo izgled novega, združenega stolpca. V našem primeru ga sredinimo, na levem in desnem robu pa ga omejimo z dvema navpičnima črtama.
- Med tretjim parom zavutih oklepajev `{}` zapišemo besedilo, ki naj se izpiše v združenem stolpcu.

Tabele

Za gradnjo tabel uporabimo okolje `table`, znotraj njega pa že omenjeno okolje `tabular`:

```
\begin{table}[htb]
```

```

\begin{tabular}
...
\end{tabular}

\caption{}
\label{}
\end{table}

```

Okolje `table` je plavajoče okolje, kar pomeni, da L^AT_EX sam smiselno uvrsti tabelo po prevodu dokumenta in jo oštevilči. Če bi uporabili zgolj okolje `tabular`, pa bi se ta struktura pojavila točno na tistem mestu v besedilu, kjer smo jo zapisali. Problem pri razporejanju nastopi seveda takrat, ko na strani ni več dovolj prostora za neko tabelo in zato na tistem mestu ostane prazen prostor. Te težave reši plavajoče okolje `table`, ki mu lahko še namignemo, kam naj tabelo postavi. Opcije v oglatih oklepajih [], ki sledijo ukazu `\begin{table}`, pomenijo:

- S črko `h` sistemu L^AT_EX namignemo, da želimo imeti tabelo točno na tem mestu (*h* – *here*).
- S črko `t` sistemu L^AT_EX namignemo, da želimo imeti tabelo na vrhu strani (*t* – *top*).
- S črko `b` sistemu L^AT_EX namignemo, da želimo imeti tabelo na dnu strani (*b* – *bottom*).
- S črko `p` sistemu L^AT_EX namignemo, da naj ustvari posebno stran, na kateri naj bodo le plavajoča okolja (tabele in slike) (*p* – *page*).
- Možne so seveda tudi smiselne kombinacije. Kombinacija `htb` pomeni, da naj najprej poskuša tabelo prikazati točno na tem mestu, nato na vrhu strani, sicer pa ob njenem vznožju.

Ukaz `\caption` nam omogoča, da tabeli dodamo spremno besedilo, ki ga vpišemo med zavita oklepaja `{}`. Številčenja ni potrebno zapisovati, saj za to poskrbi L^AT_EX sam.

Tudi tabele lahko poimenujemo, kar nam nato omogoča, da se nanje sklicujemo v besedilu, podobno kot na vire iz poglavja o literaturi. Ime predpišemo z ukazom `\label`, kjer dejansko ime zapišemo med zavita oklepaja `{}`:

```
\label{rezultati}
```

V besedilu se na tabelo z imenom `rezultati` sklicujemo z ukazom `\ref`:

```
\ref{rezultati}
```

Da bo ustrezen sklic viden v prevedenem dokumentu, moramo seveda tudi tu dvakrat prevesti naš dokument. Ker sklic predstavlja zgolj ustrezno številko, ga moramo navadno opremiti tudi z ustreznim kontekstom. Oglejmo si primer kode in prevoda:

```
V tabeli~\ref{rezultati} lahko...
```

```
V tabeli 1 lahko...
```

Z znakom `~` smo sistemu \LaTeX ukazali, da naj prevod besedila `tabeli~\ref{rezultati}` prikaže v isti vrstici, torej naj ga na tem mestu ne razdeli na dve vrstici. Ustrezen kontekst k številki tabele v zgornjem primeru je beseda `tabeli`.

Podobno velja tudi za sklicevanje na poglavja, razdelke, slike in enačbe. Pri dodeljevanju imen moramo biti pozorni, da se nam imena ne ponavljajo.

Vključevanje slik

Čeprav je v sistemu \LaTeX možno izdelati preproste diagrame, večino slikovnega gradiva izdelamo z drugimi grafičnimi orodji. Vse slike v formatu PostScript lahko enostavno vključimo v dokumente \LaTeX s pomočjo paketov `graphics` in `epsfig`.

Okolje `figure` je v osnovi zelo podobno okolju `table`:

```
\begin{figure}[htb]

\psfig{figure=,width=}

\caption{}
\label{}
\end{figure}
```

zato omenimo le bistvene razlike in zahteve:

- V tretjo vrstico glave dokumenta moramo dodati vrstico, ki vključuje paketa za delo s slikami:

```
\usepackage{graphics,epsfig}
```
- Slike vključujemo z ukazom `\psfig`, kjer med zavita oklepaja `{}` zapišemo ime slike in njeno zahtevano širino. Recimo, da želimo vključiti sliko z imenom `UPO.eps`, ki naj bo široka 8 cm. Potem bo koda izgledala takole:

```
\psfig{figure=UP0.eps,width=8cm}
```

V tem primeru mora slika biti v istem imeniku kot datoteka s končnico `tex`, sicer pa moramo ustrezno navesti pot do slike.

- Z ukazom `psfig` lahko vključujemo le slike v formatu PS (PostScript) ali EPS (Encapsulated PostScript). Slike v formatu EPS vidimo tudi v prevedeni L^AT_EX datoteki na računalniškem zaslonu. Za prikaz slik v formatu PS moramo prevedeno datoteko natisniti na tiskalniku, ki zna interpretirati jezik PostScript, lahko pa jo prevedemo tudi v format PDF in jo nato prikažemo s programom Adobe Reader. Primeri vključevanja slik so v 9. poglavju.

5.3.9 Matematični izrazi

Matematična okolja

Poznamo dva tipa matematičnih okolij:

- vrstičnega in
- sredinjenega.

Vrstično matematično okolje uporabljamo tam, kjer želimo znotraj besedila zapisati nek matematični izraz. V tem primeru matematični izraz vkleščimo med znaka `$`.

Primer kode in njenega prevoda:

Komutativnost: `$a+b=b+a$`

Komutativnost: $a + b = b + a$

Sredinjeno matematično okolje pa uporabljamo tam, kjer želimo matematičen izraz izpostaviti, zato se izpiše na sredino, hkrati pa se z vertikalnim razmikom loči od besedila pred izrazom in za njim. Poznamo več sredinjenih matematičnih okolij:

- Osnovno sredinjeno matematično okolje je zelo podobno prej omenjenemu vrstičnemu, le da sedaj matematični izraz vkleščimo med para znakov `$$`. Primer kode in prevoda:

`$$a+b=b+a$$`

$a + b = b + a$

- Okolje `array` lahko uporabljamo le znotraj matematičnih okolij in je namenjeno ustvarjanju tabel izrazov. Po svoji sintaksi je podobno okolju `tabular`. Razložimo delovanje na naslednjem primeru:

```

$$
\begin{array}{ccl}
a+b & & a-b & & a+b+c \\
1 & & 2 & & 3
\end{array}
$$

```

V prevodu ta del kode vidimo kot:

$$\begin{array}{ccc}
 a + b & a - b & a + b + c \\
 1 & 2 & 3
 \end{array}$$

Komentar kode:

- Osnovno okolje:

```

\begin{array}
\end{array}

```

- Pomen izraza med zavitima oklepajema `{ccl}`:

- * znak `c` podaja ukaz sistemu \LaTeX , da naj bo izraz v prvem stolpcu sredinjen (*c* – *center*),
- * znak `l` podaja ukaz sistemu \LaTeX , da naj bo izraz v drugem stolpcu poravnan levo (*l* – *left*),
- * znak `r` podaja ukaz sistemu \LaTeX , da naj bo izraz v tretjem stolpcu poravnan desno (*r* – *right*),
- * celoten izraz pove, da smo ustvarili strukturo s tremi stolpci.

- Ukaz `&` določa mejo med stolpci.
- Ukaz `\\` določa zaključek vrstice.

Okolje največkrat uporabimo pri ustvarjanju matrik, determinant, deljenju izrazov na več vrstic itd. Oglejmo si primer kode in prevoda, nato pa še nekaj ukazov, ki nam pri tem ustvarjanju pridejo prav:

```

$$
\left\{
\{\mathbf{X}\}=
\left[
\begin{array}{c}

```

```

a+b & a-b & a+b+c\\
1 & 2 & 3
\end{array}
\right]+...\right.
$$

```

$$\left\{ \mathbf{X} = \begin{bmatrix} a+b & a-b & a+b+c \\ 1 & 2 & 3 \end{bmatrix} + \dots \right.$$

```

\left(      veliki okrogli oklepaj
\right)    veliki okrogli zaklepaj
\left[      veliki oglati oklepaj
\right]    veliki oglati zaklepaj
\left\{     veliki zaviti oklepaj
\right\}   veliki zaviti zaklepaj
\left|     velika navpična črta na levi strani
\right|    velika navpična črta na desni strani
\left.     na levi strani ni prikazan oklepaj
\right.    na desni strani ni prikazan zaklepaj

```

Ukaza `\left.` in `\right.` uporabimo za logični zaključek (začetek ali konec) nekega izraza ali njegovega dela, kjer ne želimo imeti para oklepajev.

- L^AT_EX nam omogoča tudi samodejno številčenje matematičnih izrazov. V ta namen uporabimo okolji `equation` in `equation*`. Najprej si oglejmo na primeru koda in prevoda okolje `equation`:

```

\begin{equation}
a+b=b+a
\label{Komutativnost}
\end{equation}

```

$$a + b = b + a \tag{5.1}$$

Komentar:

- Okolje deluje podobno kot okolje za sredinjenje `$$`, z razliko, da izrazu predpiše zaporedno številko, ki jo prikaže na desnem robu med okroglimi oklepaji.
- Podobno kot pri okolju `table` in `figure` lahko tudi enačbam damo imena in se nanje nato sklicujemo v besedilu. V našem primeru smo ime `Komutativnost` predpisali z ukazom `\label`:

```
\label{Komutativnost}
```

In kako se na to enačbo sklicujemo v besedilu? Tudi tukaj uporabljamo ukaz `\ref`. Oglejmo si sklic na primeru kode in prevoda:

```
...v ena\v cbi~(\ref{Komutativnost}) je prikazana...
...v enačbi (5.1) je prikazana...
```

Tudi tukaj seveda velja, da moramo dokument prevesti dvakrat, da bo ustrezen sklic viden v prevedenem dokumentu. V primeru, da ga prevedemo le enkrat, je v prevedenem dokumentu na mestu vsakega sklica viden vprašaj. In ker sklic predstavlja zgolj ustrezno številko, ga moramo opremiti z ustreznim kontekstom. V našem primeru smo to naredili s parom okroglih oklepajev `()` in besedo `ena\v cbi`.

Sedaj si oglejmo na primeru kode in prevoda še okolje `eqnarray`, ki nam omogoča številčenje vrstic tabel izrazov:

```
\begin{eqnarray}
a+b & = & b+a
\nonumber\\
a+(b+c) & = & (a+b)+c
\label{Asociativnost}\\
a*(b+c) & = & a*b+a*c
\label{Distributivnost}
\end{eqnarray}
```

$$a + b = b + a$$

$$a + (b + c) = (a + b) + c \quad (5.2)$$

$$a * (b + c) = a * b + a * c \quad (5.3)$$

Komentar:

- Okolje deluje podobno kot okolje `array`, z dvema bistvenima razlikama: vsaki vrstici predpiše zaporedno številko, ki jo prikaže na desnem robu med okroglimi oklepaji in ukazu `\begin{eqnarray}` ne sledi ukaz za oblikovanje (na primer `{clr}`).
- Če določene vrstice ne želimo oštevilčiti, potem pred zaključkom vrstice uporabimo ukaz `\nonumber`.

S tem smo osvojili koncept delovanja matematičnega okolja v sistemu L^AT_EX, zato podajmo ostale priročne ukaze v obliki spiskov:

<code>\sqrt{x}</code>	\sqrt{x}	<code>\dot{a}</code>	\dot{a}
<code>\underline{x}</code>	\underline{x}	<code>\ddot{a}</code>	\ddot{a}
<code>\overline{x+y}</code>	$\overline{x+y}$	<code>\vec{a}</code>	\vec{a}

<code>\leftarrow</code>	\leftarrow	<code>\Leftarrow</code>	\Leftarrow
<code>\rightarrow</code>	\rightarrow	<code>\Rightarrow</code>	\Rightarrow
<code>\leftrightarrow</code>	\leftrightarrow	<code>\Leftrightarrow</code>	\Leftrightarrow

<code>\oint</code>	\oint	<code>\nabla</code>	∇	<code>\sum</code>	\sum	<code>\forall</code>	\forall
<code>\prod</code>	\prod	<code>\exists</code>	\exists	<code>\bigcap</code>	\bigcap	<code>\neg</code>	\neg
<code>\bigcup</code>	\bigcup	<code>\leq</code>	\leq	<code>\bigvee</code>	\bigvee	<code>\geq</code>	\geq
<code>\bigwedge</code>	\bigwedge	<code>\subset</code>	\subset	<code>\pm</code>	\pm	<code>\subseteq</code>	\subseteq
<code>\cdot</code>	\cdot	<code>\equiv</code>	\equiv	<code>\times</code>	\times	<code>\propto</code>	\propto
<code>\Re</code>	\Re	<code>\in</code>	\in	<code>\Im</code>	\Im	<code>\notin</code>	\notin
<code>\partial</code>	∂	<code>\sim</code>	\sim	<code>\infty</code>	∞	<code>\simeq</code>	\simeq
<code>\emptyset</code>	\emptyset	<code>\approx</code>	\approx	<code>\neq</code>	\neq	<code>\doteq</code>	\doteq

<code>\alpha</code>	α	<code>\iota</code>	ι	<code>\varrho</code>	ϱ
<code>\beta</code>	β	<code>\kappa</code>	κ	<code>\sigma</code>	σ
<code>\gamma</code>	γ	<code>\lambda</code>	λ	<code>\varsigma</code>	ς
<code>\delta</code>	δ	<code>\mu</code>	μ	<code>\tau</code>	τ
<code>\epsilon</code>	ϵ	<code>\nu</code>	ν	<code>\upsilon</code>	υ
<code>\varepsilon</code>	ε	<code>\xi</code>	ξ	<code>\phi</code>	ϕ
<code>\zeta</code>	ζ	<code>\omicron</code>	\omicron	<code>\varphi</code>	φ
<code>\eta</code>	η	<code>\pi</code>	π	<code>\chi</code>	χ
<code>\theta</code>	θ	<code>\varpi</code>	ϖ	<code>\psi</code>	ψ
<code>\vartheta</code>	ϑ	<code>\rho</code>	ρ	<code>\omega</code>	ω
<code>\Gamma</code>	Γ	<code>\Xi</code>	Ξ	<code>\Phi</code>	Φ
<code>\Delta</code>	Δ	<code>\Pi</code>	Π	<code>\Psi</code>	Ψ
<code>\Theta</code>	Θ	<code>\Sigma</code>	Σ	<code>\Omega</code>	Ω
<code>\Lambda</code>	Λ	<code>\Upsilon</code>	Υ		

Kot ste najbrž opazili, so praktično vsi ukazi zelo razumljivi, saj so to večinoma kar ustrezne angleške besede. Tako zapišemo na primer *podčrtaj* x enostavno kot `\underline{x}`, da dobimo \underline{x} .

5.3.10 Za konec še nekaj podrobnosti

Seveda pa to še zdaleč ni vse, kar nam L^AT_EX ponuja. Tukaj smo omenili le osnove, nič pa nismo spregovorili o ukazih za risanje slik, oblikovanju makrojev ipd. Na samem koncu podajmo le še nekaj koristnih ukazov:

- Z ukazom `\newpage` skočimo na novo stran.
- Z ukazom `\vspace` ukažemo, da L^AT_EX naredi ustrezen vertikalni presledek med deloma dokumenta. Če želimo na primer narediti vertikalni razmik treh centimetrov, to dosežemo z ukazom `\vspace{3cm}`.
- Z ukazom `\hspace` dosežemo, da L^AT_EX naredi predpisan horizontalni presledek med deloma dokumenta. Horizontalni razmik za 15 milimetrov dosežemo z ukazom `\hspace{15mm}`.
- Z ukazom `\vfill` dosežemo, da se besedilo, ki temu ukazu sledi, poravna na spodnji rob strani.
- Z ukazom `\hfill` dosežemo, da se besedilo, ki temu ukazu sledi, poravna na desni rob.
- Z ukazom `\mbox` dosežemo, da se besedilo v matematičnem okolju izpiše kot običajno besedilo, saj sicer matematično okolje ignorira vse presledke:

Pravilno:	koda	→	<code>\$z_n=x_n^2+y_n^2 \mbox{ za } n=1..5\$</code>
	prevod	→	$z_n = x_n^2 + y_n^2$ za $n = 1..5$
Napačno:	koda	→	<code>\$z_n=x_n^2+y_n^2 za n=1..5\$</code>
	prevod	→	$z_n = x_n^2 + y_n^2$ za $n = 1..5$
- Če uporabljamo ukaz `twocolumn` za dvokolonsko oblikovanje besedil in želimo prikazati tabelo ali sliko preko obeh stolpcev (to je možno le na vrhu strani), to naredimo tako, da za ukazom `table` oziroma `figure` zapišemo znak `*`:

```
\begin{figure*}
...
\end{figure*}
```

- L^AT_EX oziroma T_EX ima vgrajena pravila za deljenje besed ameriške angleščine. Ker so pravila za deljenje besed v različnih jezikih različna, L^AT_EX besed v drugih jezikih ne bo vedno delil pravilno. Sistemu L^AT_EX je možno dodati pravila za deljenje besed za druge jezike, med drugimi tudi za slovenščino, vendar to presega osnovno znanje. Če beseda na koncu vrstice ni pravilno deljena, lahko

sistemu \LaTeX pokažemo, kje lahko deli besedo; na primer besedo “delitev” na naslednji način: `de\li\tev`.

- \LaTeX omogoča tudi barvni tisk, kar lahko s pridom izkoristimo, če imamo barvni tiskalnik ali če je izpis namenjen le prikazu na računalniškem zaslonu (npr. za svetovni splet).
- \LaTeX je pravzaprav pravi programski jezik, v katerem je možno celo računati. Kot primer je na sliki 5.34 prikazano besedilo, ki je poravnano vzdolž parametrično podane krivulje. Na sliki 5.35 pa je koledar, ki ga \LaTeX pri prevajanju vsakokrat na novo izračuna za poljubno podano leto.

My Watch (An Instructive Little Tale)—Mark Twain, 1870

My beautiful new watch had run to the watchmaker to be regulated. end of twenty-four hours, it would ing along quietly for nearly eight
 eighteen months without losing or He asked me if I had ever had it trot up to the judges' stand all right hours, everything inside would
 gaining, and without breaking any repaired. I said no, it had never and just in time. It would show let go all of a sudden and be-
 part of its machinery or stopping. needed any repairing. He looked a fair and square average, and no gin to buzz like a bee, and the
 I had come to believe it infallible a look of vicious happiness and man could say it had done more hands would straightway begin to
 in its judgments about the time of eagerly pried the watch open, and or less than its duty. But a correct spin round and round so fast that
 day, and to consider its constitu- then put a small dice-box into his average is only a mild virtue in a their individuality was lost com-
 tion and its anatomy imperishable. eye and peered into its machinery. watch, and I took this instrument pletely, and they simply seemed
 But at last, one night, I let it run He said it wanted cleaning and oil- to another watchmaker. He said a delicate spider's web over the
 down. I grieved about it as if it ing, besides regulating—come in the king-bolt was broken. I said I face of the watch. She would reel
 were a recognized messenger and a week. After being cleaned and was glad it was nothing more seri- off the next twenty-four hours in
 forerunner of calamity. But by and oiled, and regulated, my watch ous. To tell the plain truth, I had no six or seven minutes, and then
 by I cheered up, set the watch by slowed down to that degree that idea what the king-bolt was, but I stop with a bang. I went with a
 guess, and commanded my bod- it ticked like a tolling bell. I be- did not choose to appear ignoran heavy heart to one more watch-
 ings and superstitions to depart. gan to be left by trains, I failed all to a stranger. He repaired the king- maker, and looked on while he
 Next day I stepped into the chief appointments. I got to missing my bolt, but what the watch gained took her to pieces. Then I prepared
 jeweler's to set it by the exact time, dinner; my watch strung out three in one way it lost in another. It to cross-question him rigidly, for
 and the head of the establishment days' grace to four and let me go would run awhile and then stop this thing was getting serious. The
 took it out of my hand and pro- to protest; I gradually drifted back awhile, and then run awhile again, watch had cost two hundred dol-
 ceeded to set it for me. Then he into yesterday, then day before, and so on, using its own discre- lars originally, and I seemed to
 said, "She is four minutes slow— then into last week, and by and by tion about the intervals. And every have paid out two or three thou-
 regulator wants pushing up." I the comprehension came upon me time it went off it kicked back like sand for repairs. While I waited
 tried to stop him—tried to make that all solitary and alone I was lin- a musket. I padded my breast for and looked on I presently recog-
 him understand that the watch uring along in week before last, a few days, but finally took the nized in this watchmaker an old
 kept perfect time. But no; all this and the world was out of sight. watch to another watchmaker. He acquaintance—a steamboat engi-
 human cabbage could see was I seemed to detect in myself a picked it all to pieces, and turned neer of other days, and not a good
 that the watch was four minutes sort of sneaking fellow-feeling for the ruin over and over under his engineer, either. He examined all
 slow, and the regulator *must* be the mummy in the museum, and a glass; and then he said there ap- the parts carefully, just as the other
 pushed up a little; and so, while desire to swap news with him. I peared to be something the matter watchmakers had done, and then
 I danced around him in anguish, went to a watchmaker again. He with the hair-trigger. He fixed it, delivered his verdict with the same
 and implored him to let the watch took the watch all to pieces while and gave it a fresh start. It did confidence of manner.
 alone, he calmly and cruelly did I waited, and then said the barrel well now, except that always at ten He said: "She makes too much
 the shameful deed. My watch be- was "swelled." He said he could minutes to ten the hands would steam—you want to hang the
 gan to gain. It gained faster and reduce it in three days. After this shut together like a pair of scis- monkey-wrench on the safety-
 faster day by day. Within the week the watch *averaged* well, but noth- sors, and from that time forth they would travel together. The oldest I brained him on the spot, and had
 it sickened to a raging fever, and ing more. For half a day it would man in the world could not make him buried at my own expense.
 its pulse went up to a hundred go like the very mischief, and keep head or tail of the time of day My uncle William (now deceased,
 and fifty in the shade. At the end up such a barking and wheezing by such a watch, and so I went alas!) used to say that a good horse
 of two months it had left all the and whooping and sneezing and again to have the thing repaired. was a good horse until it had run
 timepieces of the town far in the snorting, that I could not hear my- self think for the disturbance; and This person said that the crystal away once, and that a good watch
 rear, and was a fraction over thir- self think for the disturbance; and This person said that the crystal away once, and that a good watch
 teen days ahead of the almanac. It as long as it held out there was had got bent, and that the main- was a good watch until the re-
 was away into November enjoy- not a watch in the land that stood spring was not straight. He also pairers got a chance at it. And
 ing the snow, while the October any chance against it. But the rest remarked that parts of the works he used to wonder what became
 leaves were still turning. It hurried of the day it would keep on slow- needed half-soleing. He made these of all the unsuccessful tinkers,
 up house rent, bills payable, and ing down and fooling along until things all right, and then my time- and gunsmiths, and shoemakers,
 such things, in such a ruinous way all the clocks it had left behind piece performed unexceptionably, and engineers, and blacksmiths;
 that I could not abide it. I took it caught up again. So at last, at the save that now and then, after work- but nobody could ever tell him.

Slika 5.34: Primer nadzora oblike tiskane površine s programom \LaTeX

2007

January							February							March							
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
	1	2	3	4	5	6					1	2	3					1	2	3	
7	8	9	10	11	12	13	4	5	6	7	8	9	10	4	5	6	7	8	9	10	
14	15	16	17	18	19	20	11	12	13	14	15	16	17	11	12	13	14	15	16	17	
21	22	23	24	25	26	27	18	19	20	21	22	23	24	18	19	20	21	22	23	24	
28	29	30	31				25	26	27	28				25	26	27	28	29	30	31	
April							May							June							
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
1	2	3	4	5	6	7			1	2	3	4	5							1	2
8	9	10	11	12	13	14	6	7	8	9	10	11	12	3	4	5	6	7	8	9	
15	16	17	18	19	20	21	13	14	15	16	17	18	19	10	11	12	13	14	15	16	
22	23	24	25	26	27	28	20	21	22	23	24	25	26	17	18	19	20	21	22	23	
29	30						27	28	29	30	31			24	25	26	27	28	29	30	
July							August							September							
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
1	2	3	4	5	6	7				1	2	3	4							1	
8	9	10	11	12	13	14	5	6	7	8	9	10	11	2	3	4	5	6	7	8	
15	16	17	18	19	20	21	12	13	14	15	16	17	18	9	10	11	12	13	14	15	
22	23	24	25	26	27	28	19	20	21	22	23	24	25	16	17	18	19	20	21	22	
29	30	31					26	27	28	29	30	31		23	24	25	26	27	28	29	
														30							
October							November							December							
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	
	1	2	3	4	5	6					1	2	3							1	
7	8	9	10	11	12	13	4	5	6	7	8	9	10	2	3	4	5	6	7	8	
14	15	16	17	18	19	20	11	12	13	14	15	16	17	9	10	11	12	13	14	15	
21	22	23	24	25	26	27	18	19	20	21	22	23	24	16	17	18	19	20	21	22	
28	29	30	31				25	26	27	28	29	30		23	24	25	26	27	28	29	
														30	31						

Slika 5.35: V sistemu L^AT_EX lahko izračunamo koledar za poljubno podano leto in oblikujemo njegov izpis

5.3.11 Primerjava vizualnega in logičnega urejanja besedil

Orodja za vizualno urejanje besedil je lažje uporabljati in se jih uporabniki tudi hitreje naučijo. Z vizualnimi orodji je tudi lažje izvajati zahtevno grafično oblikovanje. Primerna so predvsem za kratka besedila, za dolga besedila pa kmalu postanejo preokorna. Nenazadnje kritiki vizualnega urejanja pravijo, da WYSIWYG pravzaprav pomeni “What you see is ONLY what you get” [9]. Skratka, pri bolj zahtevnih besedilih moramo za enotno in predvsem natančno oblikovanje upoštevati tudi njihovo vsebinsko strukturo, kar pa omogoča le logično oblikovanje.

Logično urejanje prav zaradi ločitve vsebine, to je logične strukture besedila, od oblike omogoča konsistentno oblikovanje celotnega besedila na osnovi njegove logične strukture. Logično strukturirana besedila lahko enostavno prevedemo iz ene strukturirane oblike v drugo strukturirano obliko (npr. \TeX v HTML ali obratno), ali pa jih ustvarimo z drugimi računalniškimi orodji (npr. ustvarjanje enačb v formatu \LaTeX v programu Mathematica). Nenazadnje lahko z logičnim urejanjem besedil dosežemo veliko višjo in konsistentno tipografsko kvaliteto.

Kot primer fleksibilnosti logičnega urejanja besedil si oglejmo naslednji primer ločitve strukture in oblike. V izvornem besedilu v formatu \LaTeX bomo pisali notranji produkt na naslednji način: $\text{\np}\{A\}\{B\}$. Z definicijo makro ukaza \np za notranji produkt

```
 $\text{\newcommand}[2]\{\text{\np}\}\{(\#1,\#2)\}$ 
```

notranji produkt A in B izpišemo kot (A, B) . Le z ustrezno spremembo makro ukaza pa lahko notranji produkt povsod v besedilu izpišemo tudi na druge načine: (A, B) , $(A|B)$ ali $\langle A|B \rangle$.

Prednost logičnega urejanja besedil v primerjavi z vizualnim so tudi lažje prenosljive in veliko manjše datoteke. Izvirne datoteke logično označenih besedil (\LaTeX , HTML, XML) so povsem običajne datoteke ASCII, ki ne zastarijo in jih je možno prebrati na praktično katerekoli vrsti računalnika. Datoteke vizualno urejenih besedil pa za razliko poleg samega besedila uporabljajo celo vrsto dodatnih kontrolnih znakov, ki so razumljivi večinoma le programom, s katerimi smo jih uredili. Prenosljivost na druge računalniške platforme je zato veliko težja, težave pa pogosto nastopijo s starejšimi datotekami, ki jih z novimi različicami ne moremo več prebrati. Prisiljeni smo tudi stalno kupovati nove različice vizualnih urejevalnikov, saj s starejšo različico programa običajno ne moremo brati datotek, ki so rezultat novejših različic. Novejše različice povrh vsega za običajnega uporabnika celo ne nudijo nobenih vidnih oziroma funkcionalnih izboljšav. Datoteke vizualno urejenih besedil so nemalokrat 10–100 krat večje kot datoteke istih besedil z logičnimi oznakami.

5.3.12 Naloge

- Opišite bistvene razlike med vizualnim in logičnim urejanjem besedil! Podajte vsaj dva primera orodij za vsako skupino!
- Opišite splošen postopek dela s sistemom L^AT_EX od vpisa kode do prikaza prevoda!
- Kaj smo povedali o osnovni zgradbi datotek s končnico `tex`?
- Naštejte vsaj pet okolij sistema L^AT_EX! Katero okolje smo označili kot okolje vseh okolij?
- V katerih primerih smo rekli, da moramo za pravilen rezultat L^AT_EX kodo prevesti dvakrat? Bi si upali sklepati zakaj je temu tako?
- Kakšna je podobnost in kakšna je razlika med okoljema `verbatim` in `verb`? Opišite še analogijo z okoljema `$$` in `!`
- V sistemu L^AT_EX ustvarite datoteko, katere prikaz bo sledeč:

REZULTATI	
Ime in priimek	Ocena
Jure Kosec	10
Mateja Cerar	9
Vid Kovač	8

- V sistemu L^AT_EX ustvarite datoteko, katere prikaz bo podajal enačbo za izračun povprečne napake ocene globine l v primerjavi z dejansko razdaljo d nad n izbranimi značilkami na sceni:

$$\text{AVG}_{\%} = \frac{\sum_{i=1}^n |l_i - d_i|/d_i}{n} \cdot 100\%$$

- V sistemu L^AT_EX ustvarite datoteko, katere prikaz bo podajal rotacijsko matriko okoli koordinatne osi x :

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$

- Kakšen bo rezultat prikaza naslednje L^AT_EX kode po dvakratnem prevodu?

```

...
\section{Sklepi}\label{razdelek:sklepi}
\begin{itemize}
\item \LaTeX\ je zakon!
\item Sklicevanje je sila enostavno: smo v razdelku \ref{razdelek:sklepi}.

```

```

\item Tudi za na\v{s}tevanje velja enako:
\begin{enumerate}
\item  $a+b=b+a$ 
\item  $$$$a+b=b+a$$ 
\end{enumerate}
\item ...
\item Od danes naprej zato za resno delo uporabljam le \v{s}e \LaTeX!
\end{itemize}
...

```

11. Nek začetnik se trudi s pripravo dokumenta v sistemu \LaTeX . Zamisli si sledečo tabelo:

\	Mesečni	Letni
Doprinos	15%	26%

in vnese naslednjo kodo:

```

...
\begin{tabular}{|clr|}\hline
\ & Mese\v{c}ni & Letni /hline
Doprinos & 15% & 26% /hline
\end{tabular}
...

```

Kje vse se je avtor zmotil? Kako bi izgledala pravilna koda?

5.4 Koristne spletne povezave

1. Spletna stran OpenOffice.org:
<http://www.openoffice.org/>
2. Spletna stran slovenskega OpenOffice.org:
<http://sl.openoffice.org/>
(S te strani je pod licenco GNU/FDL dostopna tudi knjiga [4].)
3. Zavezništvo OpenOffice.org – koristne informacije glede namestitve in uporabe paketa OpenOffice.org:
<http://ooz.agenda.si/>
4. Arhiv CTAN \TeX :
<http://www.ctan.org/>
5. Slovenska skupina uporabnikov sistema \TeX :
<http://vlado.fmf.uni-lj.si/texceh/texceh.htm>
6. Slovenščina in \TeX :
<http://nl.ijs.si/GNUsl/tex/tslovene/slolang/node98.html>

7. Spletna stran podjetja Adobe Systems Incorporated:
<http://www.adobe.com/>
8. PostScript:
<http://www.adobe.com/products/postscript/main.html>
9. Word Processors: Stupid and Inefficient:
<http://ricardo.ecn.wfu.edu/~cottrell/wp.html>

5.5 Literatura

- [1] V. Batagelj and B. Golli. *T_EX, Povabilo v T_EX, L^AT_EX, BibT_EX, PicT_EX*. Društvo matematikov, fizikov in astronomov Slovenije, Ljubljana, 1990.
- [2] M. Goossens, F. Mittelbach, and A. Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, MA, 1994.
- [3] M. Goossens and S. Rahtz. *The L^AT_EX Web Companion*. Addison-Wesley, Reading, MA, 1999.
- [4] M. Goossens, S. Rahtz, and F. Mittelbach. *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, MA, 1997.
- [5] S. Haugland and F. Jones. *OpenOffice.org 1.0 Resource Kit*. Prentice-Hall PTR, Upper Saddle River, NJ, 2003.
- [6] M. Hladnik. *Praktični spisovnik ali šola strokovnega ubesedovanja*. Filozofska fakulteta, Ljubljana, 1991.
- [7] D. E. Knuth. *The T_EXbook*. Addison-Wesley, Reading, MA, 1994.
- [8] D. E. Knuth. *Digital Typography*. CSLI Publications, Stanford, 1999.
- [9] L. Lamport. *L^AT_EX A Document Preparation System, User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, second edition, 1994.
- [10] G. Leete, E. Finkelstein, and M. Leete. *OpenOffice.org for Dummies*. Wiley Publishing, Inc., Hoboken, NJ, 2003.
- [11] R. Ludvik, I. Zajc, and A. Medic. *Hitri vodnik po OpenOffice.org*. Pasadena, Ljubljana, 2003. (Dostopna pod licenco GNU/FDL na: http://openoffice.lugos.si/knjiga/Hitri_vodnik_po_OpenOffice.org_FDL.pdf).
- [12] R. C. Parker. *Grafično oblikovanje*. Pasadena, Ljubljana, 1997.
- [13] Adobe Systems. *PostScript Language Tutorial and Cookbook*. Addison-Wesley, 1985.

- [14] J. Veen. *The Art & Science of Web Design*. New Riders, Indianapolis, IN, 2000.
- [15] I. Winkler. *Pisanje strokovnih sestavkov*. Biotehniška fakulteta, Ljubljana, 1998.
- [16] K. Možina. *Knjižna tipografija*. Bibliothecaria 13. Univerza v Ljubljani, Filozofska fakulteta, Oddelek za bibliotekarstvo, Ljubljana, 2003.