

L^AT_EX 2_ε latviski: problēmas un risinājumi, vispārīgas piezīmes

Adrians Heidens

<adrians.heidens@gmail.com>

2008. gada 3. jūnijā.

Anotācija

Darbā ir apskatīta latviešu valodas lietošana dokumentu sagatavošanas sistēmā L^AT_EX 2_ε, kā arī vispārīgas piezīmes par tās lietošanu. Konkrētajā gadījumā ir aplūkots darbs ar Linux operētājsistēmu un L^AT_EX 2_ε instalācijas pakotni `texlive`, bet daudz ko var attiecināt vispārīgi uz jebkuru L^AT_EX 2_ε darbības vidi. Darbā tiek uzskatīts, ka lasītājam L^AT_EX 2_ε lietošanas pamati ir zināmi.

Saturs

1	Vispārēji norādījumi, piezīmes	2
1.1	Pirmā paragrāfa atkāpe	2
1.2	Klases faili	2
1.3	Opcija “draft”	3
1.4	Matemātiskās formulas	3
1.5	Nestandarta saturs	4
1.6	“eps” failu iekļaušana	4
1.7	Ātrāka “pdf” failu veidošana	5
2	Latviešu valodas lietošana	5
2.1	Problēmas apraksts	5
2.2	Pakotnes “ucs” un “inputenc”	6
2.3	Pakotne “babel”	7
A	Pielikums	10
A.1	Savs kompilācijas skripts	10
A.2	Latviešu burtu skripts	12
A.3	Titula lapa	13

1 Vispārēji norādījumi, piezīmes

Lai saprastu $\text{\LaTeX} 2_{\epsilon}$ pietiekamā līmenī, ir nepieciešams izlasīt, pirmkārt materiālu `lshort.pdf` [1], kur var gūt priekšstatu par to, kas tas ir un spert pirmos soļus. Tur ir pietiekami daudz informācijas, lai sāktu rakstīt pirmos dokumentus, kā arī norādes uz dažādiem noderīgiem materiāliem. Grāmatā [2] var iegūt pamatīgākas zināšanas par \LaTeX lietošanu.

Šajā nodaļā apskatīsim dažādus ieteikumus darbā ar $\text{\LaTeX} 2_{\epsilon}$, kurus nevar atrast grāmatās tiešā veidā, kā arī komandas un trikus, kas ir apskatīti grāmatās, bet sākotnēji šķiet nevajadzīgi vai nesvarīgi.

1.1 Pirmā paragrāfa atkāpe

Daudzās grāmatās sastopama prakse, ka visi paragrāfi jeb rindkopas sākas ar atkāpi no malas, tomēr bieži atkāpi liek tikai tādā gadījumā, ja pirms konkrētā paragrāfa iepriekš ir cits paragrāfs. Citiem vārdiem sakot, pirmais paragrāfs pēc virsraksta netiek sākts ar atkāpi no malas. \LaTeX pēc noklusējuma pirmo paragrāfu pēc virsraksta sāk bez atkāpes, tāds ir arī daudzu tipogrāfijas speciālistu ieteikums. Ideja ir tāda, ka atkāpes funkcija ir norādīt, kur beidzas viens paragrāfs un sākas otrs, tātad pēc virsraksta jēgas atkāpei nav, jo ir skaidrs, ka sākas paragrāfs.

Iespējams atkāpes tradīcijas atšķiras dažādās valstīs vai pat dokumentu tipos, tas jāprasa kādam tipogrāfijas speciālistam. Iespējams, ka tas nav nekur noteikts un ir katra paša gaumes jautājums. Man personīgi labpatīk ideja, ka pirmajam paragrāfam nevajadzētu būt ar atkāpi.

Ja tomēr ir nepieciešamība vai vēlēšanās lietot atkāpes pirmajiem paragrāfiem, tad ir divas iespējas. Viens variants ir lietot speciālu \LaTeX pakotni:

```
\usepackage{indentfirst}
```

vai arī pievienot savam klases failam (`cls`) sekojošas rindiņas:

```
\let\@afterindentfalse\@afterindenttrue  
\@afterindenttrue
```

1.2 Klases faili

Ir pieejami standarta \LaTeX dokumentu klases faili, kuros ir definēts, kā izskatīsies dokuments, tā dažādi elementi, kā arī pieejamās komandas un tamlīdzīgi.

Ir iespējams veidot pašam savas klases, kurā definēt dažādu elementu izskatu pēc savas gaumes un nepieciešamības. Vislabāk ir paņemt gatavu

klasi, piemēram, `report.cls` un pamainīt detaļas. Šādā veidā ir ērti mainīt standarta virkņu nosaukumus. Piemēram, ja ir vajadzība nomainīt nosaukumu “Literatūra” par “Literatūras saraksts” un tamlīdzīgi, rekomendē to darīt klases failā, nevis Babel valodas definīcijas failā, pie tam valodas definīcijas fails būs pieejams rakstīšanai tikai sistēmas īpašniekam. Tikai klases failā var izmantot tikai `ascii`.

Lai izmantotu savu klases failu, L^AT_EX programmai jāzina, kur tas atrodas. Viens veids, kā to izdarīt, ir savu klases failu un izejas koda failu novietot vienā direktorijā. Norādīt uz savu klases failu var ar komandu

```
\documentclass[a4paper,12pt]{myreport}
```

kur fails `myreport.cls` ir savs klases fails.

1.3 Opcija “draft”

Pie visām standarta dokumentu klasēm ir iespējams norādīt opciju `draft`

```
\documentclass[a4paper,12pt,draft]{article}
```

Tas ir īpaši noderīgi tad, ja raksta dokumentu bez automātiskas vārdu pārnesšanas jaunā rindā (*hyphenation*). Ja dokumentu nokompilē ar `draft`, tad problemātiskās rindas, kur vajadzētu kādu vārdu pārnest nākamā rindā, ir atzīmētas ar melnu četrstūri rindas galā. Tādā veidā uzreiz var redzēt, kurš vārds būtu jāpārnes.

Ieteicamais darba modelis ir uzrakstīt sākumā visu dokumentu saturiski un tad, sagatavojot gala versiju, pielabot attiecīgās vietas, pārnest vārdus, sakārtot *float* elementus un tamlīdzīgi.

1.4 Matemātiskās formulas

Ja dokumentā ir nepieciešams lietot daudz matemātiskās izteiksmes, tad ir ieteicams izmantot L^AT_EX pakotni `amsmath` [4]

```
\usepackage{amsmath}
```

Bieži ir nepieciešams lietot tādas vides, kā `gathered`, `aligned`, tās kombinēt dažādos veidos, lai panāktu vēlamo efektu, piemēram

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

```

\end{aligned}\\
...
\end{gathered}
\end{equation}

```

1.5 Nestandarta saturs

Arī tad, kad ir izlasīts viss, lai saprastu \LaTeX lietošanu labā līmenī, tomēr bieži nav skaidrs, kā noformēt lapas saturu, kas nav standarta formā, piemēram, titula lapu, dažādas formas un citu.

To var paveikt ar komandu palīdzību, kas aprakstītas [1] 6. nodaļā un [2] 6. nodaļā. Var būt noderīgas sekojošas komandas un vides:

- `fill`, `hfill`, `vfill`, `stretch`, `dotfill`, `hrulefill`
- `hspace`, `hspace*`, `vspace`, `vspace*`
- `mbox`, `parbox`, `tabular`
- `flushleft`, `flushright`, `centering`

Ir iespējamās dažādas kombinācijas, kuras varētu būt noderīgas, kā piemēram

- `\hspace*{\fill}`, `\hspace{\stretch{1}}`

Titula lapas piemērs ir pielikumā A.3.

1.6 “eps” failu iekļaušana

Ir jāuzmanās, kad darbā iekļauj `eps` failus. Tajos var glabāties informācija, kā lapas izmērs, nosaukums un tamlīdzīgi. Nokompilējot dokumentu gala variantā, kā `pdf`, tas var atspoguļoties failā. Piemēram, gatavojot A4 formāta dokumentu un iekļaujot kādu `eps`, kurā ir norādīts, ka tas ir A5, beigās nokompilētais `pdf` var parādīties, kā A5. Iespējams, ka tas neko nemaina, jo izskatās tieši tāpat, kā A4, bet varbūt tomēr tas atstāj kādu ietekmi.

Programmām `dvips` un `ps2pdf` var norādīt lapas izmēru, kā argumentu:

```

bash$ dvips -t a4 <file>
bash$ ps2pdf -sPAPERSIZE=a4 <file>

```

Pielikumā A.1 ir `bash` skripts, kurā ir lietotas šīs komandas.

1.7 Ātrāka “pdf” failu veidošana

Izejas koda failu kompilācija ir iespējama dažādos veidos, kā, piemēram, `pdflatex` un `latex`. Kompilējot ar programmu `latex`, rezultātā iegūst `dvi` failu, no kura var savukārt iegūt gan `ps`, gan `pdf` failu tipus. Savukārt no `ps` arī var iegūt `pdf`.

Dažādās situācijās iespējams, ka ātrums ir atšķirīgs, bet ir gadījies, ka ātrākā metode ir tāda:

1. Ar programmu `latex` iegūst `dvi` failu.
2. No `dvi` faila ar programmu `dvips` iegūst `ps` failu.
3. `ps` failu pārveido par `pdf` failu ar programmu `ps2pdf`.

Skripts, kas realizē šo metodi ir atrodams pielikumā A.1.

Pārsvarā pietiek tikai ar `dvi` failu, lai apskatītu, kā izskatīsies darbs, un tikai gala versijā iegūt `pdf`.

2 Latviešu valodas lietošana

2.1 Problēmas apraksts

Latviešu valodas atbalsts $\text{\LaTeX} 2_{\epsilon}$ dokumentu sagatavošanas sistēmai ir nepietiekams. Oriģinālie D. Knuta (Donald Ervin Knuth) radītie fonti *Computer Modern* savulaik tapuši \TeX sistēmai satur faktiski tikai angļu valodas burtus. Tādēļ attēlojot kādu latviešu valodas burtu, tas tiek kombinēts no vairākām daļām, piemēram, burts “ā” īstenībā ir “a” ar mākslīgi pieliktu svītru virs tā, tas nav atsevišķs burts. Šī īpatnība rada dažādas problēmas, kā, piemēram, problēmas ar vārdu pārnesanu jaunā rindā (*hyphenation*), meklēšanu `pdf` dokumentos un citas.

Lai arī nav pilnīgs valodas atbalsts, tomēr ir iespējams rakstīt dokumentus latviešu valodā. Viens veids, kā to darīt ir kodā attiecīgos burtus rakstīt ar atbilstošu komandu (tab. 1.). Problēma ir ar burtu “ģ”, kodējumā `\v{g}` vai tā citā pierakstā `\v g` burts kompilējās nepareizi. Latviešu valodā pareizi ir “ģ”, nevis “g”. Iespējams, ka līdzīgāki varianti ir `\'g` vai `\'g`, kas attiecīgi ir `g` un `g`, bet tie arī īsti nav tas, ko vajadzētu.

Principā arī šādā veidā var rakstīt dokumentus, bet tas nebūt nav ērti. Lai būtu ērtāk un korektāk, var izmantot pakotnes `ucs` un `inputenc`.

ā	\={a}	\=a	Ā	\={A}	\=A
č	\v{c}	\v c	Č	\v{C}	\v C
ē	\={e}	\=e	Ē	\={E}	\=E
ģ	\v{g}	\v g	Ģ	\v{G}	\v G
ī	\={i}	\=i	Ī	\={I}	\=I
ķ	\c{k}	\c k	Ķ	\c{K}	\c K
ļ	\c{l}	\c l	Ļ	\c{L}	\c L
ņ	\c{n}	\c n	Ņ	\c{N}	\c N
š	\v{s}	\v s	Š	\v{S}	\v S
ū	\={u}	\=u	Ū	\={U}	\=U
ž	\v{z}	\v z	Ž	\v{Z}	\v Z

Tabula 1: Latviešu burtu komandas

2.2 Pakotnes “ucs” un “inputenc”

Kombinējot šīs pakotnes, ir iespēja dokumenta izejas kodā rakstīt ar latviešu burtiem, proti, nav katrs latviešu burts jākodē ar komandām, kuras ir redzamas tabulā 1. Dokumenta preambulā ir jāiekļauj sekojošas komandas:

```
\usepackage[dirty]{ucs}
\usepackage[utf8x]{inputenc}
```

Šīs komandas nodrošina iespēju dokumentu sastādīt **utf8** kodējumā. Arī burts “ģ” tiek kompilēts vizuāli pareizi. Tomēr burts tiek veidots “netīrā” veidā (arguments **dirty**). Tas tiek kombinēts sarežģītāk nekā citi latviešu burti. Tas rada problēmas, ja burts ir sastopams tādās vietās kā nodaļu vai sadaļu nosaukumos un citās.

Lai izmantotu burtu “ģ” nosaukumos, jālieto komanda **\protect** pirms burta, piemēram:

```
\caption{Rež\protectģis}
```

Tomēr ar to nepietiek gadījumos, kad attiecīgais nosaukums iekļaujas kādā sarakstā, kurš tiek ģenerēts automātiski, kā, piemēram, satura, figūru vai tabulu rādītājos. Tad ieteicamais rīcības variants ir norādīt alternatīvo nosaukumu bez attiecīgā burta, piemēram:

```
\section[Rez'g'is]{Rež\protectģis}
```

Piemērā ir kodēti gan burts “ž”, gan “ģ”. Principā var iztikt tikai ar “ģ” burta kodēšanu. Apostrofs izmantots, lai sagatavojot dokumenta gala versiju, varētu viegli nomainīt attiecīgos burtus. Dažādi saraksti, kā, piemēram, satura rādītājs tiek ģenerēti no failiem. Satura fails ir ar paplašinājumu **toc**.

Pirms dokumenta gala versijas kompilēšanas jāizlabo `toc` fails. Tajā var būt arī latviešu burti. Pielikumā A.2 ir Perl skripta piemērs burtu pārvēršanai.

Tomēr tādā veidā nevar atrisināt problēmas ar `fancyhdr` pakotnes izmantošanu, kura ļauj noformēt lapas augšdaļu un apakšdaļu dažādos veidos. Piemēram, katras lapas augšdaļa ar nodaļas nosaukumu. Ja nodaļas nosaukumā ir “ģ”, tad tik vienkārši to nevarēs atrisināt. Pagaidām nav zināms risinājums, labāk atturēties no `fancyhdr` izmantošanas.

2.3 Pakotne “babel”

Katrai valodai ir savas tipogrāfiskās īpatnības, piemēram, dažādas atstarpes starp pieturzīmēm, citādākas pēdiņas, skaitļu pieraksts un tamlīdzīgi. Izmantojot `babel`, ir iespējams vieglāk strādāt ar dažādām valodām. Ja dokumentā vienlaicīgi ir jālieto vairākas valodas, tad rekomendē Babel. Pakotne nodrošina tādas lietas, kā

- Ģenerētās teksta virknes attiecīgajā valodā.
- Automātisku vārdu pārvešanu jaunā rindā pēc noteiktiem likumiem.
- Specifiskus tipogrāfijas noteikumus.

Babel atbalsta daudzas valodas, diemžēl tajā skaitā nav latviešu. Tomēr to var izmantot vismaz divās lietās, ko tā nodrošina, respektīvi, automātiskās teksta virknes latviešu valodā un tipogrāfijas noteikumi. Automātisku vārdu pārvešanu jaunā rindā ierobežo fontu trūkums. Latviešu valodā, par laimi, nav tik bieži sastopama nepieciešamība pārnest vārdu jaunā rindā, kā tas ir, piemēram, somu valodā, kur mēdz būt ļoti gari vārdi. Lietojot citus fontus, nevis oriģinālos *Computer Modern*, zūd daļa no $\text{\LaTeX 2}_{\epsilon}$ vērtības.

Tā kā Babel pakotnē nav iekļauts latviešu valodas definīcijas fails, to jāveido pašiem (`latvian.ldf`). Piedāvāju manis izveidoto failu, kur par pamatu ir ņemtas tādas valodas, kā vācu un igauņu. Faktiski viss ir reducēts līdz pamatiem. Definīcijas failā lietoju latviešu burtus, kaut arī visus citos valodu definīcijas failos izmantots tikai `ascii`.

```
\ProvidesLanguage{latvian}
[2008/05/26 v1.0 Latvian support from the babel system]
\LdfInit{latvian}\captionlatvian
```

Norāda, ka vārdu pārvešanas definīcijas fails ir `hohyphenation`, tas nozīmē, ka bez automātiskas vārdu pārvešanas jaunā rindā.

```
\let\l@latvian\l@nohyphenation
```

Ja gadījumā neatrod `nohyphenation`, tad rīkojas tā, kā gadījumā ar citām valodām. Ja neatrod attiecīgo vārdu pārvešanas definīciju – paziņo par to un uzstāda noklusēto mehānismu, kas latviešu valodai ir nederīgs.

```
\ifx\l@latvian\@undefined
  \@nopatterns{Latvian}
  \adddialect\l@latvian0
\fi
```

Uzstāda latviešu valodas automātiski ģenerētās teksta virknes. Viss nav iztulkots, iespējams, ka šo to vajadzētu tulkot citādāk. Laba ideja ir pievienot vēl vairāk iespējamus nosaukumus, piemēram, `refnamea`, `refnameb` un tamlīdzīgi, lai varētu tos izmantot klašu failos. Ievērojam, ka klašu failos nevar izmantot latviešu burtus, bet te it kā var.

```
\addto\captionslatvian{%
  \def\prefacename{Priekšvārds}%
  \def\refname{Norādes}%
  \def\bibname{Bibliogrāfija}%
  \def\appendixname{Pielikums}%
  \def\contentsname{Saturs}%
  \def\listfigurename{Attēlu saraksts}%
  \def\listtablename{Tabulu saraksts}%
  \def\indexname{Indekss}%
  \def\figurename{Attēls}%
  \def\tablename{Tabula}%
  \def\partname{Daļa}%
  \def\enclname{encl}% <-- Needs translation
  \def\ccname{cc}% <-- Needs translation
  \def\headtoname{To}% <-- Needs translation
  \def\pagename{Lapa}%
  \def\seename{skat.}%
  \def\alsoname{skat. arī}%
  \def\proofname{Proof}% <-- Needs translation
  \def\glossaryname{Glossary}% <-- Needs translation
  \def\abstractname{Anotācija}%
  \def\chaptername{Nodaļa}%
}
```

Latviešu valodas datuma definējums (`\today`).

```
\def\month@latvian{\ifcase\month\or
  januārī\or februārī\or martā\or aprīlī\or maijā\or jūnijā\or
```



```

jūlijā\or augustā\or septembrī\or oktobrī\or novembrī\or
decembrī\fi}

\def\datelatvian{%
\def\today{\number\year.\space gada\space
\number\day.\space\month@latvian.}}

Pielauju, ka latviešu valodai ir tādi paši tipogrāfijas noteikumi, kā vācu un
igauņu, kā arī daudzām citām valodām. Tas attiecas un atstarpi aiz punkta
teikuma beigās. Angļu valodā ir papildus atstarpe pēc punkta teikuma beigās.

\addto\extraslatvian{\bbl@frenchspacing}
\addto\noextraslatvian{\bbl@nonfrenchspacing}

Komanda \providehyphenmins attiecas un vārdu pārnesšanas likumiem. Šādi
ir lielākai daļai valodu, droši vien arī latviešu. Ja netiek pārnesti vārdi auto-
mātiski, komanda diez vai dod kādu ietekmi.

\providehyphenmins{\CurrentOption}{\tw@\tw@}

\ldf@finish{latvian}

\endinput

```

Lai padarītu latviešu valodu atpazīstamu Babel sistēmai, ir nepieciešams izdarīt trīs lietas:

1. Izveidot failu `latvian.ldf`.
2. Izveidot failu `latvian.sty`.
3. Pievienot failā `babel.sty` sekojošu rindiņu:
`\DeclareOption{latvian}{\input{latvian.ldf}}.`

Manā gadījumā šie faili atrodas šādā direktorijā:

`/usr/share/texmf-texlive/tex/generic/babel/`

Šķiet, ka valodu `sty` faili visām valodām ir gandrīz vienādi. Manis lietotais ir paņemts no igauņu valodas un pielāgots latviešu.

```

\ifx\LdfInit\undefined
\def\LdfInit{%
\chardef\atcatcode=\catcode'\@
\catcode'\@=11\relax
\input babel.def\relax

```

```

\catcode'\@=\atcatcode \let\atcatcode\relax
\LdfInit}
\fi
\ifx\PackageError\undefined
\def\ProvidesLanguage#1[#2 #3 #4]{%
\wlog{Language: #1 #4 #3 <#2>}}%
\else
\def\bblstyerror{%
\PackageError{babel}%
{You have used an old interface to call babel%
\MessageBreak
You may expect compatibility problems wit other%
packages}%
{You could proceed but don't complain if you run%
into errors}%
\let\bblstyerror\@undefined
}
\ifx\ProvidesLanguage\undefined
\bblstyerror
\def\ProvidesLanguage{%
\chardef\atcatcode=\catcode'\@
\catcode'\@=11\relax
\input babel.def\relax
\catcode'\@=\atcatcode \let\atcatcode\relax
\ProvidesLanguage}
\else
\bblstyerror
\fi
\fi
\def\CurrentOption{latvian}
\input latvian.ldf\relax

\endinput

```

A Pielikums

A.1 Savs kompilācijas skripts

Skriptam ir jābūt pieejamai programmai `lv-convert`, kura ir pielikumā A.2. Skripts realizē kompilācijas ideju, kas aprakstīta nodaļā 1.7.

```

#!/bin/bash

usage="usage: my-latex file.tex"

if [ ! 'which lv-convert' ]
then
    echo "lv-convert not found" >&2
    exit 1
fi

if [ $# -lt 1 ]
then
    echo $usage >&2
    exit 1
fi

file='expr "$1" : '\(.*\)\'...*'
if [ -e "${file}.toc" ]
then
    lv-convert < "${file}.toc" > "${file}.toc.tmp"
    mv "${file}.toc.tmp" "${file}.toc"
else
    echo "toc file doesn't exist!" >&2
fi

latex "${file}.tex"
if [ $? -ne 0 ]
then
    exit 1
fi

dvips -t a4 "${file}.dvi"
if [ $? -ne 0 ]
then
    exit 1
fi

ps2pdf -sPAPERSIZE=a4 "${file}.ps"
if [ $? -ne 0 ]
then
    exit 1
fi

```

```
fi
```

```
exit 0
```

A.2 Latviešu burtu skripts

Skripts no standarta ievada padotā teksta, kur latviešu burti ir kodēti ar burtu un aiz tā apostrofu, izvada standarta izvadā pārvērstu tekstu, kas ir ar normāliem latviešu burtiem.

```
#!/usr/bin/perl
```

```
use warnings;
```

```
use strict;
```

```
while(<STDIN>) {
```

```
    s/a' /ā/g;
```

```
    s/c' /č/g;
```

```
    s/e' /ē/g;
```

```
    s/g' /ģ/g;
```

```
    s/i' /ī/g;
```

```
    s/k' /ķ/g;
```

```
    s/l' /ļ/g;
```

```
    s/n' /ņ/g;
```

```
    s/s' /š/g;
```

```
    s/u' /ū/g;
```

```
    s/z' /ž/g;
```

```
    s/A' /Ā/g;
```

```
    s/C' /Č/g;
```

```
    s/E' /Ē/g;
```

```
    s/G' /Ģ/g;
```

```
    s/I' /Ī/g;
```

```
    s/K' /Ķ/g;
```

```
    s/L' /Ļ/g;
```

```
    s/N' /Ņ/g;
```

```
    s/S' /Š/g;
```

```
    s/U' /Ū/g;
```

```
    s/Z' /Ž/g;
```

```
    print;
```

}

exit 0;

A.3 Titula lapa

Bakalaura darba titula lapas piemērs.

```
\begin{titlepage}
\setlength{\parindent}{0em}
\begin{center}
\textsc{
Ventspils Augstskola\\
Informācijas Tehnoloģiju Fakultāte}\\
\vspace{2em}
\textbf{Bakalaura darbs}\\
\vspace{2em}
{\LARGE \textbf{Lorem ipsum dolor}}\\
\vspace{2em}
\begin{tabular}{@{}r@{}l@{}}
\parbox[c]{0.4\textwidth}{Autors:}&
\parbox[t]{0.6\textwidth}{
Ventspils Augstskolas\\
Informāciju tehnoloģiju fakultātes\\
Bakalaura studiju programmas\\
‘‘Datorzinātnes’’\\
3. kursa students\\
Jānis Kalniņš\\
Matrikulas Nr. 2005020083\vspace{0.7em}\\
\mbox{} \hrulefill \vspace{-0.4em}\\
{\scriptsize(paraksts)}\vspace{2em}} \\\
\parbox[c]{0.4\textwidth}{Fakultātes dekāns:}&
\parbox[t]{0.6\textwidth}{
Dr.~\mathit{.}, docents R.~Valdmanis\vspace{.7em}\\
\mbox{} \hrulefill \vspace{-0.4em}\\
{\scriptsize(paraksts)}\vspace{2em}} \\\
\parbox[c]{0.4\textwidth}{Zinātniskais vadītājs:}&
\parbox[t]{0.6\textwidth}{
Dr.~\mathit{.}, docente I.~Liepa\vspace{.7em}\\
\mbox{} \hrulefill \vspace{-0.4em}\\
{\scriptsize(paraksts)}\vspace{2em}} \\\
\end{tabular}
\end{center}
\end{titlepage}
```

```

\parbox[c]{0.4\textwidth}{Recenzents:} & \vspace{.7em}\\
\multicolumn{2}{@{}c@{}}{
\mbox{}\hrulefill
}\vspace{-0.4em}\\
\multicolumn{2}{@{}l@{}}{
{\scriptsize(Ieņemamais amats, zinātn. nosaukums,
vārds, uzvārds)}
}\vspace{.7em}\\
&\mbox{}\hrulefill\vspace{-0.4em}\\
&{\scriptsize(paraksts)}\\
\end{tabular}
\vfill
Ventspils, 2008
\end{center}
\end{titlepage}

```

Literatūra

- [1] T. Oetiker, H. Partl, I. Hyna, E. Shlegl. *The Not So Short Introduction to L^AT_EX 2_ε*. Nāk līdz L^AT_EX 2_ε kā `lshort.pdf`.
- [2] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, 1994.
- [3] Johannes Braams. *Babel, a multilingual package for use with L^AT_EX's standard document classes*. Nāk līdz L^AT_EX 2_ε kā `babel.pdf`.
- [4] *User's Guide for the "amsmath" Package*. American Mathematical Society, 1999. Nāk līdz L^AT_EX 2_ε kā `amsl.doc.pdf`.